

**ВАРЛАМОВ  
ОЛЕГ ОЛЕГОВИЧ**

**ЭВОЛЮЦИОННЫЕ БАЗЫ ДАННЫХ И ЗНАНИЙ ДЛЯ АДАПТИВНОГО СИНТЕЗА  
ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ.  
МИВАРНОЕ ИНФОРМАЦИОННОЕ ПРОСТРАНСТВО**

Москва  
"Радио и связь"  
2002

УДК 004.82  
ББК 32.813  
В18

**Варламов О.О.**

Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство. - М.: Радио и связь, 2002. - 282 с.: ил. 44.

**ISBN 5-256-01650-4**

Монография посвящена теоретическим и реализационным основам построения эволюционных баз данных и знаний и систем адаптивного синтеза информационно-вычислительных конфигураций. Содержит как обзорный материал, касающийся систем искусственного интеллекта и их философских, психологических и эволюционных аспектов, так и оригинальные результаты, полученные автором в ходе разработки нового класса моделей данных и правил с изменяемой эволюционной структурой, основ адаптивного синтеза интеллектуальных систем и быстродействующих методов обработки информации. Эволюционные интерактивные динамические многомерные объектно-системные структуры унифицированного представления данных и правил названы "миварными". Суть миварного представления данных в том, что формируется информационное пространство <вещь, свойство, отношение> путем определения основных осей и фиксации основных объектов (вещей) и их отношений. Названия объектов и отношений в некотором порядке фиксируются на осях, а на соответствующих пересечениях точках этого пространства, которые называются миварами, записываются конкретные значения свойств и отношений объектов. Мивар - это некоторая вещь, обладающая конкретным свойством, находящаяся в определенном отношении.

Для специалистов в области теоретической и прикладной информатики, вычислительной математики, студентов и аспирантов вузов соответствующих специальностей, системных и прикладных программистов. Может быть использована в качестве учебного пособия для вузов по современным информационным технологиям.

Табл. 16. Ил. 44. Библиогр. 509 названий.

Рецензенты:

Академик РАН, д-р техн. наук, проф. А.В. Каляев.

Д-р физ-мат. наук, проф. Г.О. Крылов.

Д-р техн. наук И.А. Треско.

Канд. техн. наук В.В. Водолазкий.

Научное издание

*ВАРЛАМОВ Олег Олегович*

**ЭВОЛЮЦИОННЫЕ БАЗЫ ДАННЫХ И ЗНАНИЙ ДЛЯ АДАПТИВНОГО  
СИНТЕЗА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ. МИВАРНОЕ ИНФОРМАЦИОННОЕ  
ПРОСТРАНСТВО**

ИБ № 3069

© О.О. Варламов, 2002

Издательская лицензия № 010164 от 29.01.97 г.

## ОГЛАВЛЕНИЕ

Условные обозначения .....	5
Введение .....	6
1. Проблема создания эволюционных баз данных и знаний для синтеза интеллектуальных систем.....	11
1.1. Интеллектуальные системы и эволюционные базы данных и знаний (правил) .....	11
1.3. Основные подходы к проблеме искусственного интеллекта.....	22
1.4. Биологические аспекты эволюционного моделирования .....	27
1.5. Некоторые психологические аспекты синтеза интеллектуальных систем.....	30
1.6. Проблема адаптивного синтеза информационно-вычислительных конфигураций .....	38
1.7. Пути создания эволюционных баз данных и знаний для синтеза интеллектуальных систем.....	49
2. Гносеологические аспекты представления данных и знаний в интеллектуальных системах .....	55
2.1. Философские категории: вещь, свойство, отношение.....	55
2.2. Вещь как система .....	61
2.3. Вещь в процессе своего возникновения, пребывания и развития .....	66
2.4. Вещь, свойство, отношение и процесс познания мира.....	77
2.5. Методологические функции категорий вещь, свойство, отношение и синтез научного знания о мире.....	87
3. Анализ и обобщение структур представления данных традиционных моделей данных .....	91
3.1. Формализованное описание основных структур представления данных традиционных моделей данных.....	91
3.2. Анализ спд традиционных моделей данных .....	102
3.3. Сравнение основных спд традиционных моделей данных .....	105
3.4. Обобщение основных спд традиционных моделей данных .....	107
3.5. Пятиуровневая одномерная таблица представления данных .....	117
4. Миварная концепция многомерного эволюционного объектно-системного унифицированного представления данных и правил.....	124
4.1. Теоретические основы миварного пространства представления данных и правил .....	124
4.2. Основные возможности миварного представления данных .....	130
4.3. Переход от одномерных многоуровневых таблиц представления данных к многомерному миварному пространству представления данных.....	132
4.4. Сравнение трехуровневых таблиц представления данных и трехмерного миварного пространства представления данных .....	134
4.5. Структуры представления данных о миварах .....	136
4.6. Структуры представления данных об отношениях миваров .....	138
4.7. Формализованное описание миварных СПД.....	139
4.8. Пример описания данных в миварном пространстве представления данных и правил.....	140
4.9. Эволюционность миварных СПД .....	152
5. Обработка данных в миварном информационном пространстве.....	154
5.1. Обработка данных на основе адаптивной логической сети правил.....	154
5.2. Особенности обработки данных на основе миварной логической сети .....	159
5.3. Принципы "графового" поиска маршрута вывода на логической сети правил .....	160
5.4. Адаптивный механизм логического вывода на эволюционной активной сети гиперправил с мультиактивизаторами, управляемой потоком данных.....	164
5.5. Адаптивность механизма логического вывода на эволюционной активной сети гиперправил с мультиактивизаторами.....	171
5.6. Квадратичной сложности метод поиска минимального разреза многополюсных сетей .....	174
5.7. Линейный матричный метод определения маршрута логического вывода на сети правил .....	181

6. Метод максимального распараллеливания доступа к общей базе данных .....	188
6.1. Анализ существующих методов распараллеливания доступа к общей базе данных ..	188
6.2. Подход к максимальному распараллеливанию доступа к общей базе данных.....	189
6.3. Постановка проблемы обеспечения максимальности распараллеливания доступа к общей базе данных .....	190
6.4. Конвейерная реализация алгоритма функционирования сервера базы данных с максимальным распараллеливанием.....	191
6.5. Исследование параметров виртуальных потоковых баз данных.....	193
6.6. Структуры представления данных для виртуальных потоковых баз данных.....	198
6.7. Оценка быстродействия обработки потока данных.....	200
7. Линейный метод единично-инкрементного суммирования чисел.....	205
7.1. Необходимость разработки линейного метода суммирования чисел.....	205
7.2. Основы линейного метода единично-инкрементного суммирования чисел .....	206
7.3. Устройство линейного единично-инкрементного суммирования чисел.....	208
7.4. Алгоритм линейного единично-инкрементного суммирования чисел.....	210
7.5. Исследование параметров единично-инкрементного суммирования чисел .....	212
7.6. "Табличная" модификация единично-инкрементного суммирования чисел .....	213
7.7. Модификация метода суммирования чисел с уменьшением необходимого объема оперативной памяти ЭВМ.....	215
7.8. Локальные корректировки вычислений и обработки данных .....	216
8. Эволюционные неоднородные компьютерные системы и синтез интеллектуальных систем .....	219
8.1. Принципы построения однородных многопроцессорных вычислительных систем...	219
8.2. Анализ путей создания эволюционных неоднородных компьютерных систем .....	222
8.3. Взаимосвязь интеллектуальных и эволюционных неоднородных компьютерных систем.....	225
8.4. Основные пути создания систем адаптивного синтеза информационно- вычислительных конфигураций .....	229
8.5. Принципы построения и применения систем адаптивного синтеза информационно- вычислительных конфигураций .....	231
8.6. Роль эвм в процессах обработки информации .....	234
8.7. Системы искусственного интеллекта и теория  активного отражения.....	237
8.8. Психология и логико-информационный подход к моделированию интеллектуальных систем.....	248
8.9. Перспективы применения эволюционных баз данных и правил для адаптивного синтеза интеллектуальных систем .....	252
Заключение .....	260
Список литературы .....	266

## УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

<b>АБД</b>	— адаптивные базы данных,
<b>АСОИ</b>	— автоматизированные системы обработки информации,
<b>АССОИ</b>	— автоматизированные системы сбора и обработки информации,
<b>БД</b>	— базы данных,
<b>БДП</b>	— базы данных и правил,
<b>БДТ</b>	— база данных трассировки,
<b>БЗ</b>	— базы знаний,
<b>ВВК</b>	— высокопроизводительный вычислительный комплекс,
<b>ВПБД</b>	— виртуальная потоковая база данных,
<b>ИАСОИ</b>	— интеллектуальные автоматизированные системы обработки информации,
<b>ИВК</b>	— информационно-вычислительные конфигурации,
<b>ИИ</b>	— искусственный интеллект,
<b>ИМС</b>	— интеллектуальные многопроцессорные системы,
<b>ККС</b>	— конфигурации компьютерных систем,
<b>МВС</b>	— многопроцессорная вычислительная система,
<b>МДн</b>	— модели данных,
<b>МПБД</b>	— миварное пространство представления данных,
<b>ОВС</b>	— однородная вычислительная система,
<b>ОТБД</b>	— одномерная таблица представления данных,
<b>ПАК</b>	— программно-аппаратный комплекс,
<b>ПВПБД</b>	— персональная виртуальная потоковая база данных,
<b>ПДС</b>	— познающе - диагностическая система,
<b>ПБД</b>	— пространство представления данных,
<b>ППБД</b>	— подпространство представления данных,
<b>САПР</b>	— система автоматизированного проектирования,
<b>САС</b>	— система адаптивного синтеза,
<b>СИИ</b>	— системы искусственного интеллекта,
<b>СН</b>	— специального назначения,
<b>СОД</b>	— система обработки данных,
<b>СПД</b>	— структуры представления данных,
<b>ССЗ</b>	— сложные специальные задачи,
<b>СУБД</b>	— система управления базой данных,
<b>ТАО</b>	— теория активного отражения,
<b>УДЗ</b>	— уникальные диагностические задачи,
<b>ЭБД</b>	— эволюционные базы данных,
<b>ЭБДП</b>	— эволюционные базы данных и правил,
<b>ЭВМ ПА</b>	— ЭВМ с программируемой архитектурой,
<b>ЭГСОД</b>	— эволюционная глобальная система обработки данных,
<b>ЭНКС</b>	— эволюционные неоднородные компьютерные системы,
<b>ЭСОД</b>	— эволюционная система обработки данных.

## ВВЕДЕНИЕ

Одной из основных актуальных проблем, решаемых при создании автоматизированных систем обработки информации (АСОИ) и интеллектуальных систем, является обеспечение в условиях их непрерывного функционирования адаптации программно-аппаратных средств и структур представления данных и знаний для оперативного решения различных сложных задач. В настоящее время решены многие проблемы по созданию различных программных и аппаратных средств, созданы отдельные элементы интеллектуальных систем, функционируют различные сложные информационно-вычислительные системы, взаимодействующие посредством компьютерных сетей. ЭВМ проникли и прочно обосновались практически во всех сферах человеческой деятельности. Стремительное развитие информатики и техники изменило мир, решило многие задачи, но и породило новые проблемы.

В данной работе проведен анализ проблемы построения (синтеза) интеллектуальных систем и показано, что для ее решения необходимо, прежде всего, решить следующие две взаимосвязанные научные проблемы. Первая проблема - это создание эволюционных баз данных и знаний, которые могут являться основой программного обеспечения эволюционных (адаптивных) интеллектуальных компьютерных систем (систем искусственного интеллекта). Подчеркнем, что особую актуальность решению этой проблемы придает то, что в последнее время теория баз данных оказывает определяющее воздействие на многие смежные области. Например, базы данных и знаний используются при создании перспективных ЭВМ, что определяет место и роль другой проблемы, исследуемой в данной работе. Вторая проблема - это создание теоретических основ адаптивного синтеза информационно-вычислительных конфигураций (интеллектуальных систем) на базе известных методов синтеза ЭВМ. Отметим, что основное направление решения данной проблемы это именно создание адаптивных конфигураций на основе применения известных методов синтеза вычислительных систем и средств.

Две эти проблемы решают на разных уровнях: на уровне программного обеспечения (первая) и на уровне аппаратных средств (вторая), фактически одну и ту же проблему - создание адаптивных, эволюционных автоматизированных систем обработки информации, которые, в свою очередь, являются основой для создания интеллектуальных систем и систем искусственного интеллекта. Анализ проблем и задач, которые необходимо решить как для создания эволюционных баз данных, так и для построения систем адаптивного синтеза конфигураций компьютерных систем, показал общность и, в некотором смысле, взаимозависимость этих двух проблем. Именно поэтому, обе эти проблемы обсуждаются параллельно. Кроме того, решение этих двух важных проблем позволит сделать еще один шаг на пути создания систем искусственного интеллекта.

В настоящее время, под руководством академика РАН Каляева А.В., ведутся научные исследования и практические разработки по созданию ЭВМ с программируемой архитектурой, т.е. фактически по созданию адаптивных вычислительных систем и средств. Кроме того, существует достаточно много методов синтеза различных специализированных ЭВМ, а также множество научных моделей решения задач различных классов. Кроме того, проблема создания интеллектуальных систем взаимосвязана со многими научными областями, прежде всего, с психологией, биологией и философией. Более того, многие термины информатики взяты из этих наук. Поэтому в данной работе приведены краткие обзоры различных представляющих интерес научных исследований. Если же читателя интересуют только технические результаты, а не объяснение и обоснование положений, на основе которых они получены, то целесообразно пропустить "нетехнические" разделы книги.

Известно, что все модели данных в теории баз данных принято разделять на два класса: структурированные (сильноструктурированные: реляционные, сетевые, "сущность-связь" и т.п.) и неструктурированные (бесструктурные или слабоструктурированные: гипертекст, символьные и т.п.). В данной монографии предлагается рассмотреть новый класс (миварных) моделей данных с изменяемой (эволюционной) структурой. Для создания теоретических и реализационных основ построения эволюционного (миварного) пространства унифицированного представления данных и правил показано, что все традиционные структуры представления данных могут быть обобщены в виде пятиуровневой одномерной таблицы представления данных. Затем, созданы эволюционные динамические многомерные структуры представления данных, в которых могут изменяться не только значения переменных, но и количество осей пространства, т.е. сама структура представления данных. Эволюционные динамические многомерные объектно-системные структуры унифицированного представления данных и правил для краткости и однозначности обозначения названы "миварными". Основной идеей миварного подхода является то, что реальный мир существует сам по себе, но при его изучении, в процессе познания человек или АСОИ представляет себе описание этого мира в виде трехмерного пространства, осями которого являются понятия (категории) вещи (объекта), свойства и отношения. Понятия вещь, свойство и отношение - это всего лишь абстракции удобные для описания реального мира (любой предметной области). Эти абстракции аналогичны, например, трем осям Декартова геометрического пространства, так как это только три разных взгляда на одно объективно существующее "нечто". Эти три разных взаимосвязанных точки зрения на одно и тоже "нечто" и позволяют выделять из предметной области вещи, свойства и отношения. Эти три абстракции абсолютно равнозначны. Именно в этом и заключается новизна и основное отличие миварного подхода.

Суть миварного представления данных в том, что формируется пространство представления данных <вещь, свойство, отношение> путем определения основных осей и фиксации основных вещей (объектов) и их отношений. Названия объектов и отношений в некотором порядке фиксируются на осях, а на соответствующих пересечениях точек этого пространства, которые называются миварами, записываются конкретные значения свойств и отношений объектов. В дальнейшем, при появлении новых данных, они накапливаются при неизменной структуре или происходит изменение структуры путем добавления, изменения или удаления любого отдельного мивара, любой точки на любой оси пространства или даже самой оси. Важной особенностью миварного подхода является то, что отношения объектов хранятся в том же едином миварном пространстве и могут представлять собой правила или процедуры обработки. Отметим, что любое отношение также может быть добавлено, удалено или изменено в любое время.

"Мивар" - это наименьший элемент (или наименьшая адресуемая точка) трехмерного дискретного информационного пространства <вещь, свойство, отношение>. Выбор этого термина обусловлен соображениями однозначности трактовки и интерпретации. Конечно, можно сформулировать и другие определения, более конкретные или конструктивные, например, при миварном подходе к описанию вещей с учетом времени и местоположения можно сформулировать другое (более конкретное) определение мивара. Мивар - это некоторая конкретная вещь, обладающая некоторым конкретным свойством, находящаяся в некотором конкретном отношении в определенный момент времени и в конкретных географических координатах. При таком подходе, вещь (объект) - это уникальное название этой вещи, совокупность всех свойств (атрибутов) этой вещи и значения во всех отношениях этой вещи со всеми другими вещами предметной области. Очень важно, что степень детализации описания вещи может быть различной: от описания всех свойств и отношений до представления лишь сущности вещи. При этом, существует воз-

возможность адаптации данных (повышение адекватности) и учета: времени, географического расположения, системного уровня, объектных характеристик при объектно-ориентированном подходе и любых других характеристик динамического многомерного описания предметной области.

Таким образом, именно миварное информационное пространство создает фундаментальную возможность реализации самообучения для систем искусственного интеллекта и обеспечивает реальную эволюционность, адаптивность синтеза компьютерных конфигураций интеллектуальных систем.

После решения проблемы создания эволюционных (адаптивных) баз данных и знаний, представляется возможным, на основе известных ранее и полученных нами результатов, совместить все эти наработки для решения общей задачи создания оптимальной конфигурации саморазвивающихся (эволюционных, синергетических) адаптивных ЭВМ для оптимального решения конкретного класса сложных задач в некоторый период времени. Практически на вход такой адаптивной ЭВМ, представляющей собой интеллектуальную автоматизированную систему сбора и обработки информации (ИАССОИ), должны поступать некоторые сложные задачи. ИАССОИ должна определить класс решаемых задач, из своих баз данных и знаний извлечь методы решения этой задачи и требования к оптимальным аппаратным и программным средствам для решения этого класса задач. Далее проводится анализ существующих аппаратных и программных средств и формируется запрос на модернизацию программных и аппаратных средств. Затем ИАССОИ без прекращения функционирования должна будет произвести перестройку (адаптивный синтез) своей программно-аппаратной структуры и найти оптимальное решение поставленной задачи.

Кроме того, если такая ИАССОИ является познающей и одновременно диагностической, то она может "самостоятельно" вести исследование и осуществлять контроль некоторой заданной предметной области, анализируя возникающие задачи и классифицируя их. При этом, такая ИАССОИ сможет постоянно отслеживать появление новых программных и аппаратных средств для формирования предложений по адаптации своей структуры с целью оперативного решения возникающих задач и более адекватного моделирования познаваемой (исследуемой и контролируемой) предметной области.

Монография состоит из восьми разделов, первые три из которых содержат обзорный материал, а последние пять, посвящены теоретическим и реализационным основам создания эволюционных баз данных и правил для адаптивного синтеза интеллектуальных систем.

В первом разделе монографии рассмотрена проблема создания эволюционных баз данных и знаний для синтеза интеллектуальных систем, проведен обзор и анализ взаимосвязей интеллектуальных систем, эволюционных баз данных и знаний (правил) и систем искусственного интеллекта. Кроме того, выполнен анализ современного состояния в области искусственного интеллекта; рассмотрены некоторые биологические и психологические аспекты синтеза интеллектуальных систем; сформулирована проблема адаптивного синтеза информационно-вычислительных конфигураций, а также определены основные задачи исследования и пути создания эволюционных баз данных и знаний для синтеза интеллектуальных систем.

Во втором разделе проведен обзор гносеологических аспектов представления данных и знаний в интеллектуальных системах; рассмотрены философские категории: вещь, свойство и отношение, а также их методологические функции в процессе синтеза научного знания. Кроме того, предложено рассматривать категорию свойство как унарное отношение и доказано, что двух оставшихся взаимозависимых и взаимозаменяемых (но все же различных) категорий: вещь и отношение, достаточно для научного описания любых предметных областей (синтеза научного

знания), так как все остальные традиционные категории и характеристики описания являются их частными, производными случаями.

В третьем разделе проведено формализованное описание основных структур представления данных традиционных моделей данных, их анализ, сравнение и обобщение в виде нового формализма пятиуровневой одномерной таблицы представления данных.

Четвертый раздел монографии является основным, так как именно в нем предложена новая миварная концепция многомерного эволюционного объектно-ориентированного системно-структурного унифицированного представления данных и правил, а также разработаны теоретические основы построения миварного информационного пространства. Кроме того, в этом разделе проанализированы основные возможности миварного представления данных, показан переход от одномерных многоуровневых таблиц представления данных к многомерному миварному пространству представления данных, проведено сравнение трехуровневых таблиц и трехмерного миварного пространства представления данных, подробно описаны структуры представления данных о миварах и об отношениях миваров, дано их формализованное описание, определены основные направления их практической реализации, доказана их эволюционность и приведен подробный пример описания данных в миварном информационном пространстве.

Пятый раздел посвящен обсуждению проблем и особенностей обработки данных в миварном информационном пространстве. Предложены новые универсальные методы обработки данных на основе адаптивной логической сети правил и принципы "графового" поиска маршрута вывода на логической сети правил. Там же рассмотрен адаптивный механизм логического вывода на эволюционной активной сети гиперправил с мультиактивизаторами, управляемой потоком данных. Кроме того, впервые разработаны новые методы повышения быстродействия обработки данных в интеллектуальных системах: квадратичной сложности методы поиска маршрута логического вывода на основе определения минимального разреза многополюсных сетей теории графов, а также линейной сложности матричный метод определения маршрута логического вывода на сети правил.

В шестом разделе подробно рассмотрен новый достаточно универсальный метод корректного максимального распараллеливания доступа к общей базе данных в условиях недопущения взаимного искажения одновременно обрабатываемых данных, приведена конвейерная реализация алгоритма функционирования сервера базы данных с максимальным распараллеливанием, исследованы параметры предлагаемых виртуальных потоковых баз данных, а также проведена оценка быстродействия обработки потока данных различными методами, которая и доказала максимальность распараллеливания доступа предлагаемого метода по сравнению с другими известными методами доступа к общей базе данных.

В седьмом разделе монографии рассмотрен новый метод повышения быстродействия обработки данных на основе запатентованного линейного способа единично-инкрементного суммирования чисел, который при решении некоторых классов задач позволяет достичь существенного повышения оперативности обработки данных. Кроме того, предложены "табличная" модификация единично-инкрементного суммирования чисел и модификация метода суммирования чисел с уменьшением необходимого объема оперативной памяти ЭВМ. Там же проанализированы возможности и перспективы использования локальных корректировок вычислений и обработки данных.

В последнем восьмом разделе рассматривается как взаимосвязаны эволюционные неоднородные компьютерные системы и синтез интеллектуальных систем, исследованы принципы построения однородных вычислительных систем, проведен анализ основных путей создания эволюционных неоднородных компьютерных сис-

тем на основе создания систем адаптивного синтеза информационно-вычислительных конфигураций (САС ИВК). Кроме того, разработаны принципы построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций, проанализирована роль ЭВМ в процессах обработки информации, предложено новое теоретическое направление развития области искусственного интеллекта в виде теории активного отражения, подведены итоги исследований и показаны перспективы применения эволюционных баз данных и правил для адаптивного синтеза интеллектуальных систем, а, в перспективе, и для создания систем искусственного интеллекта.

Внедрение разработанных в этой монографии: новой миварной концепции динамического многомерного объектно-системного эволюционного унифицированного представления данных и правил, теоретических и реализационных основ создания эволюционных баз данных и правил (ЭБДП), принципов построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций (САС ИВК), а также методов повышения оперативности решения сложных задач, позволит создать более интеллектуальные эволюционные программно-аппаратные комплексы, т.е. позволит сделать еще один шаг к решению проблемы создания систем искусственного интеллекта.

Данная книга написана по результатам многолетних исследований, которые были бы невозможны без поддержки и помощи многих и многих друзей, товарищей, сотрудников. Разделы 1.2, 1.3, 1.4, 1.5, 8.6, 8.7 и 8.8 данной монографии написаны автором вместе с Адамовой Л.Е. Выражаю искреннюю **благодарность** моей семье, родственникам, моим учителям и наставникам: Атрощенко В.Г., Ростовцеву Ю.Г., Шеремету И.А., Крылову Г.О., Каляеву А.В. Кроме того, автор благодарен: Абатниной И.А., Агафоновой В.А., Адамовой Л.Е., Александровским О.В. и Л.Р., Асташевой Н.П., Барсукову А.Г., Безродному Б.Ф., Березину И.В., Буржановой Г.А., Быкову Г.Д., Варламовым: А.Г., А.О., Д.О., Л.О., М.О., О.Г., О.О., С.Е., Водолазкому В.В., Волковой Л.Н., Глушко В.А., Горлановой Н.К., Дорофееву А.В., Дунаеву Е.А., Заяц В.Т., Казеннову В.Н., Каляеву И.А., Качковой М.А., Козыреву В.П., Колтовой Т.И., Корнееву В.В., Куряпиной В.Ф., Лазареву В.М., Левину И.И., Мальцеву П.П., Маркеловой Н.Н., Маркову А.С., Маслюкову В.Ф., Маслюкову К.В., Отвиновскому В.В., Петричко О.С., Пивоварову А.С., Плеханову С.М., Прохорову Н.А., Рыбкину А.Ю., Сапоженко А.А., Семерикову В.В., Сергиенко И.С., Треско И.А., Труевцеву К.М., Турко Н.И., Тяхтяеву В.А., Фомину Я.А., Черникову Н.Ф., Эйсымонту Л.К. и всем тем, кто оказывал помощь в проведении научных исследований, а также при написании и опубликовании данной монографии.

# 1. ПРОБЛЕМА СОЗДАНИЯ ЭВОЛЮЦИОННЫХ БАЗ ДАННЫХ И ЗНАНИЙ ДЛЯ СИНТЕЗА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

## 1.1. ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ЭВОЛЮЦИОННЫЕ БАЗЫ ДАННЫХ И ЗНАНИЙ (ПРАВИЛ)

В этой работе исследованы актуальные научные проблемы, относящиеся к разным научным областям, но направленным на решение одной важной проблемы: исследование возможностей компьютерных систем по автоматизации умственной деятельности человека ("может ли машина мыслить?"). Отметим, что это лишь конечная цель, которая пока не достигнута, однако в настоящее время полученные теоретические научные результаты позволяют решить несколько промежуточных, но достаточно важных и взаимосвязанных научных проблем.

Понятие "интеллектуальные системы" не является строго определенным, однозначным термином, как и близкое ему понятие "искусственный интеллект". Между тем, под интеллектуальными системами принято понимать такие системы, которые способны выполнять "интеллектуальные" действия: обучаться, проводить логический анализ, синтез, осуществлять ассоциативный поиск и т.п. Отметим, что способность к самообучению является необходимой и одной из самых важных в перечне "интеллектуальных" свойств любого субъекта. На данном этапе, не будем затрагивать проблему "достаточности" этого признака, но подчеркнем именно - **"необходимость" способности к обучению для любой "интеллектуальной" системы.**

Для того чтобы система могла обучаться, ее подсистема накопления и обработки данных должна обладать свойством адаптивности, наращиваемости, изменяемости, т.е. в некотором смысле - эволюционностью. Если перейти к терминам автоматизированных систем обработки информации (АСОИ), то **базы данных (БД) и базы знаний интеллектуальных систем** должны быть адаптивными, а в более широком смысле - **эволюционными**. Термин "знания" является антропоморфным и близок по смыслу к термину "интеллект", но эти термины неоднозначно формализованы, т.е. не имеют четко выраженного, однозначного значения и, следовательно, не являются идентификаторами. Более того, с этой точки зрения, "антропоморфный" термин "базы знаний" не является корректным, так как фактически там хранятся только правила, процедуры и тому подобные отношения объектов. Термин "**правило**" хорошо формализован, а его смысл, в общем виде, более всего соответствует тому, что принято называть "знание". Следовательно, далее целесообразно употреблять вместо термина "базы знаний", в этом же смысле, другой термин: "**базы правил**", а в качестве обобщения терминов "базы данных" и "базы правил" - будем использовать обобщенный термин: "**базы данных и правил**" (БДП).

В настоящее время, в "реальных" (больших) базах данных может храниться практически любая информация, но вопросы адаптации структур хранения данных, как правило, решаются "вручную", т.е. путем перепроектирования и перепрограммирования существующих баз данных. В процессе перепрограммирования такая АСОИ не выполняет свои функции, т.е. не обеспечивается непрерывность функционирования АСОИ. Получаем, что с учетом участия программиста-человека, такие человеко-машинные АСОИ являются адаптивными и эволюционными. Однако, процесс перепроектирования и перепрограммирования БД, если не рассматривать участие человека, в общем случае, не является адапта-

цией системы хранения и обработки данных, так как на самом деле происходит создание новой базы данных, а не ее адаптация.

В определенном смысле, некоторые базы данных позволяют проводить частичную адаптацию путем добавления новых объектов, но это осуществляется либо за счет резервирования лишних объектов в процессе проектирования, либо путем добавления однотипных объектов. На самом деле, исходя из смысла термина "адаптация", такие изменения могут быть охарактеризованы лишь как "**имитация адаптации**" и не более того.

Действительно **эволюционной** (адаптивной) базой данных может быть только **база данных с изменяемой структурой** представления данных. Кроме того, существует достаточно много систем, для которых важным критерием является непрерывность ее функционирования в достаточно длительные интервалы времени. Таким образом, в настоящее время **проблема создания эволюционных (адаптивных) баз данных и правил не решена** в полном объеме.

Отметим, что термины адаптация и эволюция в нашем контексте являются синонимами, но все же более точным, более перспективным является термин эволюция, так как мы ставим целью создание развивающихся, изменяющихся непрерывно на длительном интервале времени баз данных и правил. В то же время, под адаптацией принято понимать некоторое, чаще всего, одномоментное действие. Впрочем, при таком подходе, эволюция - это много раз повторенная адаптация (изменение) в различные моменты времени. Обсуждение темы данной работы с другими уважаемыми учеными и анализ ее возможного названия, показал, что все же наиболее адекватным сейчас является термин "эволюционные базы данных и правил", хотя синонимы: адаптивные, изменяющиеся, развивающиеся и т.п. тоже могут иметь место и право на существование.

Существуют различные варианты "динамических" баз данных и знаний [Л. 9, 23-24, 28, 36, 82, 84, 91, 98, 100-104, 120-124, 141-142, 171, 185, 190-211, 216, 232, 263, 289, 314-319, 321-328, 338-340, 349-356, 358-368, 396-403, 413, 460-465, 481-509], но как было сказано выше, даже их нельзя отнести к адаптивным (эволюционным), так как они тоже только "имитируют адаптацию" и не более того. Более подробно эти вопросы освещены ниже, а сейчас можно сделать вывод о необходимости разработки действительно эволюционных (адаптивных) баз данных и правил (знаний).

Таким образом, для создания **интеллектуальных систем** необходимо решить несколько важных научных проблем. Прежде всего, это проблема создания эволюционных (адаптивных) баз данных. Затем - создание эволюционных автоматизированных систем обработки информации (АСОИ). Такие АСОИ представляется возможным создать путем решения проблемы **адаптивного синтеза конфигураций компьютерных систем**, на основе создания миварного эволюционного многомерного динамического объектно-ориентированного информационного пространства унифицированного представления данных и правил (которому и посвящена эта работа) или, для краткости, просто: **миварного информационного пространства**.

Решение этих взаимосвязанных проблем позволит создать действительно эволюционные компьютерные системы, в том числе и интеллектуальные АСОИ. В перспективе, система адаптивного синтеза конфигураций компьютерных систем предназначена для оценивания решаемой задачи, формирования требований и выбора оптимальных вычислительных ресурсов, адаптации программно-аппаратных средств, выделенных для решения требуемой задачи, и для непосредственного управления процессом решения такой задачи. Таким образом, компью-

тер "сам" будет адаптировать свою структуру для оптимального решения требуемых задач.

Исторически так сложилось, что автор участвовал в нескольких, казалось бы, разнородных, научных проектах: исследование возможностей искусственного интеллекта и экспертных систем, концептуальное проектирование и создание баз данных, оптимизация параллельных вычислительных алгоритмов, исследование особенностей новых информационных технологий и информационно-вычислительных сетей, типа Интернет, а также в создании высокопроизводительных вычислительных комплексов и других проектах. Но на каждом этапе, мои учителя и наставники: Атрощенко В.Г., Ростовцев Ю.Г., Шерemet И.А., Крылов Г.О. и Каляев А.В., помогали "увидеть за деревьями лес" и не отклоняться от цели и главного, фундаментального направления исследований: поиск путей создания интеллектуальных АСОИ.

В целом, исследуемая проблемная область создания интеллектуальных систем находится на стыке нескольких направлений, которые, в различной степени, характеризуются следующими терминами: дискретная математика, математическая логика, кибернетика, автоматизированные системы обработки информации, базы данных (БД), модели данных (МДн), базы знаний (БЗ), синтез вычислительных структур, экспертные системы, системы поддержки принятия решений, теория графов, оптимизация на сетях и графах, параллельные вычисления, системы искусственного интеллекта, многопроцессорные вычислительные системы, математическое и информационное моделирование, архитектура ЭВМ, многопроцессорные системы с программируемой архитектурой, системный анализ, философские, гносеологические и психологические аспекты обработки информации и создания искусственного интеллекта, эволюционные системы, адаптивные системы управления, автоматизация проектирования, логические системы обработки данных и некоторые другие.

Так как многие слова - "термины" используются в разных областях и в разное время в них вкладывался различный смысл, то при написании данного текста достаточно часто приводится одновременно несколько подобных, близких по значению, слов - терминов, позволяющих лучше подчеркнуть конкретное значение терминов и смысл написанного. Кроме того, отметим, что любая однопроцессорная система является частным случаем многопроцессорных систем, следовательно, в дальнейшем целесообразно использовать термин **многопроцессорные системы** при исследовании сформулированных научных проблем. Итак, основой любой автоматизированной системы обработки информации являются базы данных и правил. Термин "интеллектуальные системы" подчеркивает, что такой системе должны быть присущи некоторые черты интеллекта. Прежде всего, это способность к обучению, накоплению и обработке новых данных, к информационному моделированию исследуемой предметной области.

Традиционные структурированные модели данных существующих известных баз данных [Л. 12, 15, 19, 50-51, 87, 100-104, 117, 120-124, 139-142, 183, 190, 196-206, 220, 247-250, 263, 284, 293, 296, 305, 314-319, 461-465, 481-501] были предназначены для автоматизации обработки данных в таких предметных областях, структура описания данных которых была фиксированной, неизменной. При изменении концептуальной модели предметной области, например, добавлении новых объектов или их характеристик, проектирование структуры базы данных приходилось фактически делать заново. При разработке концептуальных моделей данных основной проблемой было учесть как можно больше возможных изменений описания предметной области. Такие системы трудно назвать "обучаемыми",

а тем более "интеллектуальными". Если в процессе функционирования базы данных можно будет, без "потери" производительности, изменять структуру концептуальной модели данных, то многие проблемы будут решены и процесс проектирования баз данных претерпит серьезные изменения в сторону упрощения. Возможно само проектирование баз данных будет осуществляться постоянно в ходе реальной непрерывной работы АСОИ.

Конечно, существуют слабоструктурированные и **неструктурированные модели** данных, т.е. "бесструктурные" (инфологические), типа гипертекста, но для них характерны другие недостатки, связанные со сложностью реализации высокоскоростной обработки данных [Л. 82, 91, 216, 316, 326-327, 462-464, 484-501]. Кроме того, наличие структуры представления данных позволяет проводить систематизацию данных, ассоциативную обработку, а многомерность значительно ускоряет обработку любых массивов и баз данных.

Создание эволюционных (адаптивных) баз данных, т.е. баз данных с изменяемой структурой (в том числе и многомерной), позволит совершить качественный скачок к созданию обучаемых систем, которые будут более соответствовать термину "интеллектуальные". Формализованное описание, анализ и обобщение традиционных структур представления данных, подробно изложенные ниже, позволили предложить новые принципы построения многомерного динамического пространства унифицированного представления данных и правил, так называемого - "миварного пространства".

**В структурированных моделях** создается некоторая структура, в рамках которой затем записываются и хранятся все данные. В неструктурированных типах все данные просто записываются и хранятся без учета структуры. В сильноструктурированные ("жесткие") модели можно записать не любые данные, но в них быстрее всего осуществляется поиск требуемых данных. В неструктурированных моделях можно хранить любые данные, но поиск требуемых данных занимает максимальное время. К **сильно структурированным** моделям можно отнести традиционные реляционные, сетевые, иерархические и т.п. модели, вплоть до модели данных "сущность-связь" [Л. 19, 27, 51, 87, 100-104, 107, 117, 120-123, 139-140, 175, 183, 190, 216, 220, 247, 249, 269-270, 284, 293, 296, 305, 314-316, 319, 363-368, 403].

К **неструктурированным** или, точнее, **слабоструктурированным** моделям можно отнести гипертекстовое представление данных [Л. 82, 91, 123-124, 216, 263, 316, 321, 326-327]. В то же время, объектно-ориентированное представление данных в такой классификации находится в средней части такой "шкалы" и обладает возможностью изменения структур представления данных путем задания новых объектов [Л. 36, 50, 136, 200-202, 216, 263, 269-270, 289, 335, 348-352, 362, 374-376, 396, 462-464, 484-501].

Отметим, что если принято различать два типа представления данных: структурированные и неструктурированные, то **миварный подход и эволюционные (адаптивные) базы данных можно отнести к новому типу представления данных с изменяемой структурой**. Такая структура обладает возможностью эволюционного наращивания, а при необходимости и кардинального изменения структуры представления данных, даже в условиях непрерывности функционирования системы. Если говорить очень кратко, то миварное представление данных позволяет явно выделять структуру или системность объектов, например, вводя в качестве оси измерения уровня системности объекта в терминах: метасистема - система - подсистема, что "поглощает" объектно-ориентированные модели дан-

ных и создает реальные предпосылки для осуществления ассоциативного поиска и выявления новых данных и отношений.

Кроме того, на основе использования материалов [Л. 145], предлагается рассматривать процессы передачи и хранения данных в автоматизированных системах в едином формате (виде):

**{<отправитель>, <получатель>, <способ>, <время>, <данные>}**.

При передаче данных время должно быть мало, а отправитель и получатель - различными. При хранении данных, наоборот: отправитель и получатель должны совпадать, а время может быть достаточно большим. Кроме того, отправитель может посылать данные нескольким получателям, время доставки может быть относительно большим, а несколько пользователей могут хранить одни и те же данные. Способ доставки характеризует форму доставки кода и способ его преобразования в требуемое представление, вид.

В этом случае, миварное представление данных объединяет возможности всех этих моделей. При этом, на начальном этапе накопления данных, миварное представление может быть ближе к слабоструктурированным моделям, но потом, при выявлении четких структур предметной области, определении взаимосвязей и взаимозависимостей миварное представление путем ввода новых осей подпространства представления данных (ППД) или изменения существующих, может быть преобразовано в более сильноструктурированное представление данных.

Более того, некоторые части подпространства миварного ППД могут быть представлены, как будет показано ниже, в различных по степени структурированности представлениях данных. Например, в некоторой автоматизированной системе обработки информации есть несколько групп пользователей. У каждого пользователя может быть свое произвольное подпространство представления данных (ПППД), в котором он обрабатывает только свои личные данные. По совместной договоренности, у каждой группы однотипных пользователей может быть создано или выделено единое общее, сильно структурированное представление накопленных данных, которые отражают некоторое единое представление данных этой группы пользователей. У всех пользователей также по некоторой общей договоренности может быть создано некоторое общее структурированное представление данных (фрагмент описания предметной области). В каждом общем подпространстве представления данных определяется некоторый ответственный пользователь - "администратор представления данных", который обеспечивает функционирование этого подпространства.

Особенностью миварного подхода является то, что при необходимости любые конкретные подпространства могут быть корректно совмещены в некое общее подпространство, а любое подпространство, также корректно, может быть разбито на несколько новых подпространств. При этом, однотипные по характеру хранения данные: изображения, фотографии, звуковые файлы, тексты, электронные таблицы и т.п. независимо от подпространств представления данных могут храниться на однотипных технических средствах, что создает предпосылки для ассоциативного поиска по однотипным (по хранению) данным. Отметим, что при миварном подходе уровень представления данных не обязательно должен соответствовать уровню хранения данных.

Более того, на уровне хранения данных могут использоваться традиционные модели данных: реляционные, сетевые и т.п., а все адресные характеристики, присущие уровню представления данных могут храниться в других таблицах, даже на других технических средствах. Конечно, миварная модель представления данных, являясь более универсальной, в то же время является и более сложной (например:

необходимость хранения и обработки адресных многомерных характеристик) по отношению к другим моделям данных и может быть сопоставлена концептуальному уровню представления данных. Тем не менее, познавательные и исследовательские автоматизированные системы обработки данных целесообразнее строить на основе наиболее универсальной миварной модели данных. Построенная на основе миварного пространства активная адаптивная логическая сеть правил, управляемая потоком данных, как будет доказано ниже, еще более расширяет "интеллектуальные" способности перспективных АСОИ. Таким образом, миварные структуры представления данных являются обобщением и научным развитием всех основных существующих структур представления данных и открывают новые возможности по построению эволюционных, адаптивных баз данных и интеллектуальных автоматизированных систем обработки информации.

Прежде всего, проанализируем область создания интеллектуальных систем и систем искусственного интеллекта. Если же читателя интересуют только технические вопросы и БД, то целесообразно сразу перейти к изучению третьего раздела данной монографии.

## **1.2. ВЗАИМОСВЯЗЬ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ И СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

Область исследования интеллектуальных автоматизированных систем взаимосвязана и взаимодействует с областью создания систем искусственного интеллекта, но эти две области все же различны.

Под **интеллектуальными системами** понимают такие автоматизированные системы, которые (с участием человека-оператора) позволяют решать различные сложные, интеллектуальные задачи. Под **системами искусственного интеллекта** принято понимать автоматические (без участия человека), самостоятельные или самообучающиеся системы, которые также должны решать сложные интеллектуальные задачи.

При общем назначении, такие системы различаются по научным подходам, по принципам построения, по методам решения задач и т.д. Сама возможность создания систем искусственного интеллекта (СИИ) до сих пор находится под вопросом, хотя отдельные интеллектуальные системы, относящиеся к классу автоматизированных систем обработки информации, созданы и эксплуатируются достаточно успешно. Для понимания существа проблемы создания эволюционных баз данных и знаний в целях синтеза интеллектуальных систем, прежде всего, необходимо проанализировать достижения и проблемы, которые существуют в этой научной области. Отметим, что проблема "интеллекта", "интеллектуальных систем" взаимосвязана со многими научными областями, но начать наше исследование целесообразно с анализа проблематики искусственного интеллекта.

### **1.2.1. ПРОБЛЕМАТИКА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

Наиболее полным, "ярким", "предельным" воплощением автоматизированных интеллектуальных систем являются системы искусственного интеллекта (ИИ). В настоящее время продолжают споры о том, что такое искусственный интеллект. Прежде всего, отметим, что проблема искусственного интеллекта возникла давно, и во все времена ученые пробовали найти подходы к ее решению. В ИИ отразились и слились проблемы философии, математики, физики и других ес-

тественных наук, а также и гуманитарные вопросы. Познание искусственного интеллекта неотделимо от познания самого человека, его сознания и мышления. Исследователи ИИ следуют великому призыву древних: "Познай самого себя". Но они не ограничиваются только этим, а выходят за рамки, границы "человекоподобия" и затрагивают самые древние, самые сокровенные тайны природы. У большинства исследователей, искусственный интеллект ассоциируется с человеческим, естественным интеллектом, который и является главным средством и главным орудием познания мира, т.е. "универсальным инструментом познания". На таком уровне осмысления проблемы ИИ, приходим к необходимости признания тесной взаимосвязи и взаимозависимости познания реального мира и познания ИИ. Познание мира невозможно без познания самого орудия этого познания. Рассматривая искусственный интеллект в более узком смысле, подчеркнем, что эта проблема не может решаться, изучаться какой-либо одной наукой. Проблема ИИ требует комплексного подхода с точек зрения различных наук, причем, все это должно сочетаться, собираться и обобщаться в виде некоторой цельной теории. Главной целью (задачей) этой теории - является изучение такого явления, предмета, объекта, который включал бы в себя и мышление человека, и процессы переработки, генерации информации различными техническими системами.

В настоящее время сложно однозначно сказать, что конкретно понимается под предметом изучения ИИ, не совсем ясна, точна и строга даже сама постановка, формулировка проблемы ИИ. Прежде всего, необходимо выяснить, понять, что такое искусственный интеллект, каковы цели и задачи его познания. Возможно, человеку еще многое необходимо изучить и понять, прежде чем он сможет осознать всю сложность, необычность и масштабность проблемы ИИ.

На наш взгляд, актуальными для настоящего этапа исследований ИИ являются следующие три основные задачи.

1. Показать всю важность и сложность проблемы ИИ, доказать, что ИИ - это наука (научное направление), но не совсем обычная.
2. Определить предмет теории ИИ, т.е. определить что изучать, исследовать, создавать.
3. Определить метод теории ИИ, т.е. как изучать.

Итак, подчеркнем, что проблема ИИ гораздо старше и информатики, и кибернетики [Л. 30, 52, 159-162, 174, 193, 212, 215, 232, 235, 239, 258, 277, 288, 292, 298, 304, 306, 324, 340, 381, 414, 415, 462-465, 472-475, 481-503], которые лишь дали новую, более научную интерпретацию искусственного интеллекта. Некоторые исследователи считают: проблема ИИ настолько необычна и сложна, что мы до сих пор не можем осознать и понять ее в целом, во всем ее многообразии, многогранности. Например, одной из задач ИИ является изучение познавательной деятельности человека, а это невозможно без познания реальности и без изучения самого процесса познания. Искусственный интеллект является, образно выражаясь, с одной стороны, "зеркалом", в котором человек видит и изучает самого себя, а с другой стороны, ИИ также является и "окном в мир", через которое человек познает окружающий его мир. При таком подходе, проблема ИИ открывается нашему взору во всем великолепии своей многообразности, сложности, важности, необходимости и неизбежности.

## 1.2.2. ОБЗОР ПОНЯТИЙ, КОНЦЕПЦИЙ И ПРЕДСТАВЛЕНИЙ В ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Прежде чем говорить об уже сложившихся подходах к проблеме искусственного интеллекта, необходимо более подробно обсудить истоки этого понятия, разобрать по отдельности, а затем и в совокупности эти термины: "интеллект", "искусственное" и, наконец, "искусственный интеллект". Прежде всего, целесообразно начать обзор понятий с термина "интеллект".

### 1.2.3. ИНТЕЛЛЕКТ

Понятие "интеллект" впервые возникло в психологии. Психологи считают, что интеллект - это "свойство личности, выражающееся в способности глубоко и точно отражать в сознании предметы и явления объективной действительности в их существенных связях и закономерностях, а также в творческом преобразовании опыта ... (и) ... система ориентировки на существенные отношения решаемой задачи ... . Стадия интеллекта (или "ручное мышление") есть высшая ступень развития психики животных, (а) ... ядро ... собственно интеллекта составляет способность выделить в ситуации ее существенные для действия свойства в их связях и отношениях и привести свое поведение в соответствие с ними" [Л. 153]. Мы видим, что интеллект не отделяется от самого человека. Отметим, что основными свойствами (способностями, возможностями) интеллекта являются:

- отражение объективной действительности,
- накопление и преобразование опыта,
- выделение существенного для отражающей системы,
- управление действиями самой системы.

Есть и другие определения понятия "интеллект" (разум), которые характеризуют взгляды "не психологов", а представителей кибернетики, математики и т.п. Например, "Интеллект - (это) а) способность понимать или обучаться на опыте; способность приобретать и сохранять знания ... ; б) способность быстро и правильно реагировать на новую ситуацию ... ; в) в психологии - мера успешности в использовании названных способностей при выполнении конкретной задачи" [Л. 304, с. 222]. В свое время, М. Минский сказал так: "...интеллект" означает едва ли больше, чем комплекс активности, который мы уважаем, но не понимаем" [Л. 193, с. 453]. В толковом словаре по информатике сказано: "интеллект - ум, рассудок, разум; мыслительные способности человека. Отдельные интеллектуальные способности человека могут быть автоматизированы путем создания систем искусственного интеллекта" [Л. 226, с. 126]. На основе анализа существующих определений и результатов научных исследований Донецкого Института проблем искусственного интеллекта было предложено и обосновано расширенное определение этого термина: "интеллект - алгоритм действия биологической системы, находящейся в сознательном состоянии, способной самостоятельно вести общение с окружающим миром, обобщать опыт, осуществлять постановку и решение задач в соответствии с выбранной целью" [Л. 47, с. 316]. Или еще более кратко: "интеллект - алгоритм решения задач, сформированных сознанием" [Л. 47, с. 318]. При этом, термин интеллект определяется через другое, более сложное понятие "сознание", что в общем случае не совсем корректно и информативно (так как один сложный термин определяется через другой, еще более сложный).

Кроме того, есть и научно-популярные толкования этого термина. Например, А. и Б. Стругацкие считают, что:

- разум есть такое свойство человека, которое отличает его деятельность от деятельности животных;
- разум есть способность живого существа совершать нецелесообразные или неестественные поступки (из горестных наблюдений);
- разум есть сложный инстинкт, не успевший еще сформироваться ...;
- разум есть способность использовать силы окружающего мира без разрушения этого мира ... [Л. 31, с. 35].

Таким образом, в данной работе, под понятием **"интеллект"** будем понимать такую систему (комплекс, объект), которая **способна отражать объективную действительность, преобразовывать накопленный опыт, выделять существенное для себя и управлять своими действиями.**

#### 1.2.4. ИСКУССТВЕННОЕ И ЕСТЕСТВЕННОЕ

Прежде всего, обратим внимание на слова Гете: "И неестественное тоже природа. Кто не видит ее во всем, тот не видит ее как должно" [Л. 31, стр. 36]. Термин "искусственное" можно охарактеризовать тремя положениями Г. Саймона [Л. 258, с. 14]:

- 1) "искусственные объекты конструируются (хотя и не всегда преднамеренно) человеком,
- 2) искусственные объекты можно охарактеризовать их функциями, целями и степенью приспособления к требованиям среды,
- 3) искусственные объекты часто, особенно при их проектировании рассматриваются не только в описательных терминах, но и с точки зрения категории должествования ...".

Поясним эти положения. Искусственные объекты создаются для чего-либо и что-то должны выполнять, делать, либо что-то замещать по замыслу их создателя, конструктора. Приведем пример. Когда человек поднимает с земли палку, чтобы сбить, сорвать банан, то эта палка является естественным объектом. Но, как только, человек соединяет вместе естественные объекты: палку и камень, чтобы такая палка с наконечником пронзала животных, так сразу же получается искусственный объект, предназначенный для охоты. Теперь это уже не палка с камнем, т.е. два естественных объекта, а один искусственный объект: копье! Попробуйте сказать, что палка или камень должны что-то делать. Они ничего не должны, а вот копье должно "летать и пронзать цель". Иначе копье и не было бы копьем.

#### 1.2.5. ПОНЯТИЕ "ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ"

В настоящее время известны различные, порой даже противоречивые, толкования понятия: "искусственный интеллект". Приведем некоторые из них. Итак, искусственный интеллект - это "...создание" таких программ для вычислительных машин, поведение которых мы бы назвали "разумным", если бы обнаружили его у людей" [Л. 298, с. 25]. ИИ - это " ... по существу инженерная дисциплина. Мы хотим строить разумные системы (,) ... эвристический поиск ... представляет главную компоненту техники искусственного интеллекта" [Л. 215, с. 10]. "Искусственный интеллект будет реализован лишь тогда, когда неодушевленная машина сможет решить задачи, которые до сих пор не удавалось решить человеку, - не

вследствие большей скорости и точности машины, а в результате ее способности найти новые методы для решения имеющейся задачи" [Л. 306, с. 28]. Мнение У. Р. Эшби таково: "Сегодня создание системы, подобной мозгу, с определенными интеллектуальными способностями в принципе также доступно, как создание паровой машины с заданной мощностью" [Л. 277, с. 155].

Есть и другие мнения: область ИИ - это "область исследований, в которой цифровые вычислительные машины используются для моделирования разумного поведения..." [Л. 92, с. 19]. "Трудности в разработке проблем "искусственного интеллекта" отражают не столько недостаточное развитие нашей технологии, сколько, пожалуй, указывают на принципиальные границы ее возможностей" [Л. 92, с. 192].

Академик А. Н. Колмогоров отмечает, что "моделирование способа организации материальной системы не может заключаться ни в чем ином, как в создании из других материальных элементов новой системы, обладающей в существенных чертах той же организацией, как и система моделируемая. Поэтому достаточно полная модель живого существа по справедливости должна называться живым существом, модель мыслящего существа - мыслящим существом" [Л. 217, с. 52].

Теперь приведем более общие "философские" определения и прогнозы: "... бурное развитие кибернетики в гармоническом сочетании с развитием молекулярной биологии и наук о высшей нервной деятельности в конечном итоге позволит создать искусственные разумные существа, принципиально не отличающиеся от естественных, но значительно более совершенные, чем они, и способные к дальнейшему самоусовершенствованию" [Л. 329, с. 318]. "Выполнение машиной сложных логических операций не есть ... мышление в строгом смысле ... . Живой мозг решает задачи совсем по иному. Он работает по направляющему плану, ведущему мысль к определенной, ясно осознаваемой цели. Машина же реализует не свои, а человеческие цели. Решая задачи по определенным формальным правилам, она, конечно, не понимает сути самой задачи и последствий своих действий. Разумеется, человек тоже "программируется" жизнью, обществом. Но он сознательно пользуется этой программой" [Л. 267, с. 55]. Г. С. Поспелов писал, что "под "искусственным интеллектом" понимается наука о том, как заставить машину делать то, что умеет делать умный человек" [Л. 31, с.3].

Интересный подход к определению ИИ приведен в книге В.И. Васильева и А.И. Шевченко - ученых Донецкого Института проблем искусственного интеллекта, где прежде определяется термин "искусственное сознание" - как "высшая управляющая система машины, владеющая определенными знаниями о себе и своем окружении, способная получать информацию и формировать новые знания в соответствии с поставленными ее создателем целями и задачами" [Л. 47, с. 331]. Затем, на основе этого определения формулируется оригинальное "простое для понимания определение термина "искусственный интеллект" ... - алгоритм решения задач, сформированный искусственным сознанием" [Л. 47, с. 331]. Впрочем, такое определение представляется не совсем корректным и информативным, так как один сложный термин определяется через другой, еще более сложный, да еще к тому взятый из другой области науки.

Кроме того, существуют определения, которые, по сути, также относятся к исследуемой области. Например, "интеллектуальная система представляет собой объединение комплекса средств познавательной деятельности с функциональным коллективом специалистов, которые используют их при постановке и решении познавательных задач" [Л. 160, с. 3-4]. При этом, такой комплекс может включать методы, семиотические системы, ЭВМ и другие технические средства, информацию и организационные структуры. Люди-специалисты обладают также вполне

конкретными возможностями, которые определяются их профессиональной подготовкой, опытом предшествующей работы, особенностями психического склада и т.п. Каждая интеллектуальная система, при условии возможностей варьирования всех выше перечисленных свойств, обладает своими специфическими особенностями, а ее организация и обеспечение функционирования требуют соответствующих знаний и методики.

Многие авторы подчеркивают необходимость решения проблемы искусственного интеллекта, например: "... под тем, что называют искусственным интеллектом, кроется общая проблема резкого упрощения общения человека с вычислительными системами на пути развития "семантических формализмов"... . Решение задач такого масштаба - задач, с которыми человечество сталкивается все более непосредственно и остро, - непосильно для ничем не вооруженного человеческого интеллекта" [Л. 31, с. 65].

Из всего выше сказанного, с учетом мнения изложенного в работе Б. В. Бирюкова и И. Б. Гутчина [Л. 31, с. 38] можно сделать три основных вывода.

1. Термин "искусственный интеллект" употребляется в двух различных смыслах. Во-первых, под этим термином понимают определенное научное направление, а во-вторых, этот термин используется как название для систем и объектов, на разработку которых и нацелены проводимые исследования.
2. Среди ученых существуют разногласия относительно возможностей, как принципиальных, так и реальных - исследовательского направления "искусственный интеллект".
3. Для обсуждения проблемы искусственного интеллекта характерен функциональный подход, т.е. машина признается "интеллектуальной" - в том или ином смысле, если в рассматриваемой сфере интеллектуальной деятельности она будет выдавать результаты аналогичные тем, какие выдают люди, а желательно, даже далеко превосходящие их результаты.

Приведем еще одно, очень важное на наш взгляд, высказывание биофизика А. Сент-Дьерди: "**Мозг есть не орган мышления, а орган выживания**, как клыки или когти. Он устроен таким образом, чтобы заставить нас воспринимать как истину то, что является только преимуществом, и тот, кто логически доводит мысли до конца, совершенно не заботясь о последствиях, должен обладать исключительной, почти патологической конструкцией. Из таких людей выходят мученики, апостолы или ученые, и большинство из них кончат жизнь на костре или же на стуле - электрическом или академическом" [Л. 239, с. 416].

Таковы основные существующие взгляды и подходы к пониманию содержания термина - "искусственный интеллект". Существуют различные подходы к исследованию проблемы искусственного интеллекта, к исследованию которых мы и переходим в следующем разделе.

### 1.3. ОСНОВНЫЕ ПОДХОДЫ К ПРОБЛЕМЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Проведенный нами анализ показал, что в настоящее время выражение "искусственный интеллект" используется в двойном смысле:

- как техническая или кибернетическая модель естественного интеллекта,
- как научное направление, лежащее на стыке разнообразных научных дисциплин.

Таким образом, аналогично, целесообразно выделить два основных **подхода к проблеме искусственного интеллекта**. На основании этих двух подходов принято выделять два направления исследований в этой научной области:

- 1) **прикладное** направление развития области искусственного интеллекта,
- 2) **фундаментальное** направление развития области искусственного интеллекта.

Рассмотрим эти направления развития области искусственного интеллекта более подробно.

#### 1.3.1. ПРИКЛАДНОЕ НАПРАВЛЕНИЕ РАЗВИТИЯ ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Рассмотрим первый подход. В предисловии к работе [Л. 340] Д. А. Поспелов пишет: "С самого начала работ по созданию интеллектуальных систем возникло два вопроса: какова глобальная цель подобных исследований и какова их глобальная стратегия?" [Л. 340, с. 6]. На первый вопрос Д. А. Поспелов отвечает так: "Несмотря на то, что призыв "Познай самого себя" актуален для человечества с тех давних времен, когда только зародилось его самосознание, на современном этапе его нельзя рассматривать как основную - и тем более единственно возможную цель. С практической точки зрения более важной представляется другая цель: создание искусственных систем, способных выполнять не хуже (а, возможно, и лучше) человека ту работу, которую люди традиционно относят к сфере интеллектуального труда" [Л. 340, с. 6]. При ответе на второй вопрос Д. А. Поспелов занимает компромиссную позицию, сочетающую воедино такие крайние точки зрения, как:

- 1) многие важные задачи могут быть решены путем "копирования" опыта природы;
- 2) ряд функций интеллекта, связанных с символьными преобразованиями, эффективнее и легче реализовывать "нечеловеческими" способами.

При таком подходе, основные исследования, которые ведутся в области искусственного интеллекта, можно свести к следующим 4 направлениям [Л. 340, с. 10].

- 1) Представление знаний и работа с ними, т.е. создание специализированных моделей и языков для представления знаний в ЭВМ, а также программных и аппаратных средств для их преобразования; исследования по созданию специальных логик, позволяющих пополнять и обобщать знания, хранимые в машине.
- 2) Планирование целесообразного поведения, т.е. исследования по созданию методов формирования целей и решения задач планирования

действий автоматического устройства, функционирующего в сложной внешней среде.

- 3) Общение человека с ЭВМ, т.е. задачи создания языковых средств, позволяющих эффективно взаимодействовать с ЭВМ непрограммирующему пользователю, а также исследования в области синтаксиса и семантики естественных языков, способов хранения знаний о языке в памяти машины и построения специальных процессоров, осуществляющих перевод текстовой информации во внутреннее машинное представление.
- 4) Распознавание образов и обучение, т.е. исследования по восприятию зрительной, слуховой и других видов информации, методам ее обработки, формированию ответных реакций на воздействия внешней среды и способам адаптации искусственных систем к среде путем обучения.

Считается, что такой подход непосредственно связан с техникой, с существующими и перспективными ЭВМ, а также с программированием. Поэтому, такой подход можно назвать (определить) "прикладным подходом в области ИИ". При таком подходе под системами искусственного интеллекта (или под искусственным интеллектом) понимают системы, выполняющие функции, которые принято считать интеллектуальными.

Таким образом, основоположники прикладного подхода и их последователи сразу же сужают область ИИ, сводя ее лишь к прикладным, чисто практическим целям. Это тоже важно, но путем создания таких искусственных систем вряд ли можно создать нечто "мыслящее", разумное. Ибо, при таком подходе изучаются лишь отдельные функции и способности мозга, следовательно, происходит упрощение задачи. Конечно же, одну из функций моделировать и исследовать проще, но нельзя забывать о том, что мозг - это динамическая саморегулирующаяся система, в которой все функции взаимосвязаны и взаимозависимы.

**Изучая только отдельные функции мозга - как системы мышления, без учета их взаимозависимости и взаимосвязанности, мы уже не исследуем, не изучаем саму систему,** которая нас интересует. Получается, что, не рассматривая, не изучая мышления как целостную, единую систему мы намереваемся создать искусственные системы, которые выполняли бы некоторые отдельные функции мышления. Но, может быть, даже **отдельные функции мышления являются свойством всей системы мышления в целом.** Следовательно, необходимо изучать, исследовать, познавать процесс мышления в целом, как единую динамическую саморегулирующуюся систему. Можно сделать такой вывод: **следуя только по прикладному направлению** в области ИИ, **в принципе, невозможно будет создать** такую искусственную систему, которая выполняла бы не хуже человека всю ту работу, которую люди относят к сфере интеллектуального труда.

Попробуем определить свойством какой системы является интеллект. Предположим, что это свойство любого человеческого мозга. Тогда можно провести следующий мысленный эксперимент. Берем некий человеческий мозг и изолируем его от всех внешних воздействий, кроме жизнеобеспечивающих, питающих мозг. Для такого эксперимента можно взять известную "голову профессора Доуэла" и накрыть ее звуконепроницаемым, черным ящиком. Итак, если в мозг не будут поступать сигналы от органов чувств, то в мозг перестанет поступать информация, и интеллекту нечего будет перерабатывать и обдумывать. В таком случае, интеллект не сможет развиваться, а, скорее всего, он просто "засохнет", испортится, "сойдет с ума". Таким образом, без органов чувств, без поступления новой ин-

формации мышление не может развиваться (даже, не сможет появиться), а вероятнее всего, мышление просто не может существовать без этого.

Следовательно, система "обладающая" интеллектом должна содержать помимо "мозга" еще и подсистему органов чувств, т.е. подсистему приема информации. По нашему глубокому убеждению, интеллект - это свойство, как минимум, всего человека, а не только человеческого мозга.

Рассмотрим другой, "литературный" опыт, например, жизнь человека на необитаемом острове, т.е. случай с Робинзоном Крузо. Интеллектом Робинзон обладал, но почему и откуда появилось у него это свойство? Интеллект - это свойство личности, отдельного человека? Робинзон был на острове единственным разумным существом и у него не было ни какой связи с другими людьми. Но от этого он не утратил свой "интеллект", ведь до этого Робинзон жил среди людей и общался с ними, получил определенные знания и навыки. Общение с людьми как с носителями "интеллекта" послужило причиной интеллектуальности Робинзона. Следовательно, кроме того, что некоторая система должна быть способной к интеллектуальной деятельности, эта система должна еще пройти курс обучения "интеллектуальности", т.е. научиться мыслить и рассуждать, а, кроме того, она еще должна накопить внутри себя, "в себе", некоторый определенный запас знаний.

В подтверждение правильности наших рассуждений, приведем пример, когда человек сразу после своего рождения остается один, без общения с другими людьми. Например, это случай с Маугли. Известно достаточно много случаев таких природных, трагических "опытов" над человеческими детьми. Такие дети, если они пробыли в лесу достаточно долгий срок, уже не могли быть нормальными людьми в полном смысле этого слова. Такие "маугли" уже никогда не способны научиться говорить, писать, мыслить.

Можем сделать такой **вывод, что конкретный человек - это лишь определенный материальный носитель интеллекта, который, в свою очередь, является свойством общественного явления - мышления или, даже, сознания.** Остается процитировать известную поговорку: "Для того, чтобы быть Человеком мало родиться человеком, надо еще и стать Человеком".

Подведем некоторые итоги. Исходя из смысла установившейся терминологии, получается, что если мы собираемся создать искусственный интеллект, то мы будем стремиться создать такую систему, которая имела бы равные возможности с естественным интеллектом. Получается борьба за чистоту терминологии и корректное применение слов. Это направление занимается созданием систем, которые моделируют, заменяют, усиливают лишь какую-то одну или несколько конкретных функций интеллекта и не более того, т.е. не занимается моделированием системы мышления, интеллекта в целом. Как может такое научное направление называться искусственным интеллектом? Ведь у этого направления даже и цели изучения интеллекта, в целом, нет. Поэтому, предлагается это научное направление называть - "прикладное направление в области искусственного интеллекта".

В заключении отметим, что можно сделать вывод о **необходимых условиях возникновения и существования интеллектуальных систем.** Перечислим данные условия.

1. Непрерывное движение, внутреннее изменение системы.
2. Постоянное отражение, связь, взаимодействие с внешним миром.
3. Активность системы, т.е. наличие у нее собственной внутренней модели, (самосознания, самоотношения), существование и действие комплекса целей.
4. Закрепление опыта, накопление и обработка информации.

5. Необходимый уровень сложности самой системы, т.е. система должна быть "физически" способна к интеллектуальной деятельности.
6. Наличие некоторого количества таких систем, образование некоторой популяции систем.
7. Наличие взаимодействия, взаимопомощи с подобными себе системами внутри определенной популяции.
8. "Размножение", порождение новых подобных себе систем, отличающихся от уже существующих и "умирание", исчезновение старых и не приспособленных систем.
9. Благоприятные, удовлетворительные условия окружающей среды, существование необходимых ресурсов для реализации системами своих целей.

Прежде всего, надо создать, разработать такую систему, которая была бы способна к интеллектуальной деятельности, а уж затем из такой "заготовки", путем обучения и самообучения, создавать систему искусственного интеллекта.

Рассмотрим другое направление в области развития ИИ.

### 1.3.2. ФУНДАМЕНТАЛЬНОЕ НАПРАВЛЕНИЕ РАЗВИТИЯ ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Одним из сторонников этого, более общего, "философского", подхода является С. М. Шалютин [Л. 324]. Данный подход можно назвать "фундаментальным", а в дальнейшем этот термин мы будем использовать без кавычек, так как он вполне выражает сущность этого научного направления. В своей работе [Л. 324, с. 6] С. М. Шалютин подчеркивает, что глубокая связь гносеологии и проблемы искусственного интеллекта имеет и такой важный аспект, как включение технических систем в познавательный процесс. Это означает, что человек как субъект познавательного процесса использует создаваемые им специальные орудия познавательной деятельности. Диалектико-материалистическая теория познания не ограничивается констатацией абстрактной возможности познания мира. Она выявляет реальные возможности и границы познания в каждую эпоху. Познание человека ограничивается психофизиологическими особенностями его организма и, здесь, ему на помощь приходит кибернетическая, информационно-вычислительная техника.

Анализ области ИИ показал, что требуется уточнение проблематики данной области и вычленение ее из совокупности вопросов, относящихся к сфере воспроизведения мышления и его функций. В этой области можно выделить три основных направления, тесно связанных между собой, но вместе с тем и существенно различающихся:

- 1) проблема искусственного интеллекта;
- 2) кибернетическое моделирование мыслительных процессов;
- 3) общетеоретические исследования.

Проанализируем содержание этих направлений [Л. 324, с. 6 - 11].

**1. Проблема искусственного интеллекта.** Исследователи, конструирующие ИИ, с самого начала ставят своей **целью создание технических систем, осуществляющих функции, которые обычно выполняет человек**, естественный интеллект. Конструкторы таких систем ориентируются на получение определенного результата: решение машинами некоторого класса задач, которые до сих пор решал человек. При этом не имеет существенного значения, подобны ли процессы, происходящие в машине, процессам, происходящим в нервной системе че-

ловека. Таким образом, конечная цель заключается в том, чтобы получить определенный результат, аналогичный тому, который получает человек в результате размышлений. В действительности, перед системой ИИ ставится цель: решение задач высокого уровня сложности, а не только получение каких-либо результатов.

**Определение 1.3.1.** В этом смысле, понятие ИИ может быть определено, как свойство технических систем решать задачи, которые и для естественного интеллекта сохраняют элементы творчества, т.е. такие задачи, которые не могут быть решены простым применением последовательности стандартных правил.

"Искусственный интеллект не есть нечто, существующее независимо от естественного интеллекта. Он является техническим, инструментальным продолжением последнего, усилителем интеллектуальных способностей человека" [Л. 324, с. 9]. Благодаря ИИ человек становится способным моделировать сложные системы, познавать их, управлять ими и, таким образом, преодолевать психофизиологическую ограниченность своей нервной системы. Особо подчеркнем, что высшие функции целеположения и формирования субъективных образов остаются только за человеком и его интеллектом. Тем более, что от систем ИИ не требуется сходства с естественным интеллектом, с точки зрения процессов его функционирования.

**2. Моделирование процессов мышления.** При моделировании процессов мышления сущность проблемы иная. Важен не результат сам по себе, а процесс, который приводил бы к результату. Моделирование мышления представляет собой применение метода моделей к познанию определенного объекта: мышления. В рассматриваемой области ученые пытаются упростить сложнейшую задачу познания мышления путем применения различных моделей. Развитие ЭВМ создает предпосылки для технического моделирования мыслительных процессов. Таким образом, постановка проблем и задач при моделировании мышления по сравнению с конструированием ИИ существенно различна. Возможно, что те или иные задачи, связанные с созданием искусственного интеллекта, могут быть решены только при условии воспроизведения в нем определенных черт познавательного процесса. Гипотеза С. М. Шалютина в том и заключается, что ряд выявленных гносеологией исторически сложившихся орудий познавательной деятельности (язык, категории и т.п.) обязательны для всякой системы, которая была бы способна выполнить весь объем интеллектуальных функций, осуществляемых человеком. Чем в большей степени эти орудия будут воплощены в системах искусственного интеллекта, тем более "разумными", "интеллектуальными" будут эти системы.

**3. Общетеоретическое направление** области искусственного интеллекта. Для этого направления актуальным является вопрос: "может ли машина мыслить?". Сущность этого направления состоит в следующем. Если бы модель была тождественна оригиналу, то ее исследование не давало бы познавательного выигрыша. Тем не менее, очень важен вопрос о возможности создания модели, столь адекватной оригиналу, что различие между нею и классом объектов, составляющих оригинал, по основным, конституирующим характеристикам, было бы не большим, чем различие внутри этого класса. Ответ на поставленный вопрос требует анализа мышления как целостной системы и сопоставления с ним созданных моделей.

Итак, сделаем вывод о том, что задача создания мыслящей машины не совпадает с задачей моделирования мышления. Другими словами, сущностью этого направления является теоретическое, фундаментальное изучение и познание мышления, с точки зрения кибернетики, математики и информатики. Результатом развития этого направления, по всей видимости, будет создание абстрактной теории мышления, с точки зрения процессов генерации и обработки знаний, информации. Дальнейшее

исследование области развития интеллектуальных систем целесообразно провести после изложения полученных нами новых научных результатов по созданию эволюционных баз данных и знаний, т.е. в последнем разделе этой работы.

При исследовании систем искусственного интеллекта и интеллектуальных систем необходимо провести анализ в некоторых других научных областях, некоторые результаты которых оказывают самое непосредственное влияние на нашу область. Более того, многие термины были взяты из психологии, медицины, гносеологии и ряда других областей современной науки. Мы не можем показать здесь обзоры всех взаимодействующих областей науки, так как это займет слишком много места в книге. Однако, наиболее важные, оказавшие самое сильное влияние на полученные нами научные результаты, так сказать, фундаментальные предпосылки необходимо здесь, хотя бы очень кратко, обсудить и проанализировать. На данном этапе нашего исследования целесообразно перейти к краткому обзору и анализу некоторых биологических (эволюция) и психологических аспектов создания интеллектуальных систем.

#### **1.4. БИОЛОГИЧЕСКИЕ АСПЕКТЫ ЭВОЛЮЦИОННОГО МОДЕЛИРОВАНИЯ**

Метод моделирования проникает сейчас во все науки. Исследуя истоки моделирования, мы приходим к мифам. Мифы - первые модели: человек мысленно населял природу сверхъестественными существами, снабжал их характерами и судьбами, влечениями и возможностями. Достигали люди этим двух целей: явления природы становились более понятными и как-то предсказуемыми. Эти функции - обслуживание понимания, обеспечение понятности и создание предсказаний - остались и у современных моделей [Л. 77]. С точки зрения нашего исследования, определенный интерес представляет книга [Л. 77] о моделях в теории эволюции, об идеях наследования, оптимальности и естественного отбора. Итак, человеческая деятельность в какой-то степени проектируется всегда, но и некоторые естественные процессы и явления бывает полезно рассматривать так, будто они искусственные, спроектированные для достижения некоторой цели. Такой прием широко распространен в биологии. Свое естественнонаучное оправдание он находит в теории эволюции, рассматривающей, в особенности после работ Дарвина, эволюцию как естественный процесс. Эволюции посвящено, в той или иной степени, большое количество работ, например [Л. 29, 32, 52, 58-61, 67, 76-77, 83, 85, 99, 133, 138, 168, 177, 187-188, 212, 217, 221, 229, 239, 256, 260, 281-283, 288, 295, 299, 301, 312, 320, 330, 332, 334, 336, 341]. Данный перечень конечно не претендует на полноту, но позволяет отразить различные интересные подходы к исследуемым нами понятиям. Например, здесь приведены книги по общим вопросам теории эволюции [Л. 29, 61, 83, 133, 239, 246, 288, 334, 341], по общим вопросам биологии [Л. 177, 187-188], по развитию самой идеи эволюции [Л. 58, 61, 99, 234, 301], по генетике [Л. 67, 168, 221, 295, 320], приведены книги классиков, чьи труды оказали огромное влияние на развитие этой области науки [Л. 85, 260, 281-283, 295, 312, 320], указаны некоторые работы, которые в определенном смысле взаимосвязаны с понятием эволюции [Л. 14, 30, 32, 44, 46, 49, 52, 59-60, 68-69, 72, 81, 92, 108, 112, 114, 118, 119, 127, 138, 159-162, 182, 186, 212, 217, 222, 229, 240, 251, 256, 258, 277, 297-299, 306, 324, 330, 332-333, 342, 427-430], а также некоторые научно-популярные источники [Л. 76-77]. Рассмотрим понятие **эволюции** более подробно.

Платон в мифе об Эпиметее изложил важную идею о существовании единой меры для различных биологических свойств. Эта мера - коэффициент размножения, т.е. среднее число выживших потомков на одного предка [Л. 77, 85, 312]. Чтобы в череде поколений не исчез ни один вид, коэффициент размножения должен быть равен единице. Однако, мир Эпиметеев - застывший мир, в котором виды не возникают и не исчезают. Такой мир эволюционного равновесия вроде бы чужд идее эволюции, но, на самом деле, необходим для ее понимания.

В теории эволюции Дарвина рассматривается возникновение одних видов и вымирание других. Очень важно предположение, что виды возникают и исчезают медленно, если измерять время продолжительностью жизни особей этих видов. Тогда получаем, что коэффициенты размножения для ныне живущих видов практически равны единице. Кроме того, появляется еще "фон сравнения" [Л. 77], т.е. это те, кого недавно не стало. Значит, они были "хуже" и их коэффициент размножения стал меньше единицы. Такой фон сравнения - застывшая история, а коэффициент размножения в данных условиях - единая мера оптимальности или совершенства. Конструировать фон сравнения на основе реальной истории сложно, поэтому используются другие подходы. Горбань А.Н. и Хлебопрос Р.Г. в [Л. 77] предлагают выделить все результаты биологических упрощений в "особое царство формальных и математических моделей", а применение таких результатов к классическим биологическим существам рассматривать на правах сравнительного анализа. Обитателей этого нового "царства" они назвали **матемазаврами**. Это безобидные и полезные существа, заменяющие теоретикам кроликов и дрозофил. Считается, что большинство матемазавров бесцельно слоняется в случайных направлениях, пока не наткнется на пищу. Покушав, матемазавры тут же или чуть позже размножаются и слоняются дальше. Матемазавры эволюционируют очень быстро и могут вести себя более осмысленно.

В конце 20-х годов 20-го столетия оформилось новое научное направление, которое может быть охарактеризовано, как **математическая теория эволюции**. Впрочем, официального единого названия это направление не имеет, но своим созданием обязано многим исследователям, которые теперь называют себя не биологами, а эволюционистами [Л. 76-77]. Наиболее известные из них: Дж. Б.С. Холдейн, Р.А. Фишер, С. Райт. Интересно, что первые шаги в этом направлении были сделаны еще в 19 веке Т. Мальтусом и, как отмечал Ч. Дарвин, послужили основой для наводящих соображений при создании теории отбора. В целом математическая теория эволюции так же, как и более известное направление, не использующее математику, опирается на идеи Ч. Дарвина об отборе. Отметим, что, в целом, в теории эволюции очень много разнообразных идей, часто противоречащих друг другу. Теория эволюции с помощью естественного отбора предстает перед нами сейчас одним из воплощений идеи Ньютона о динамических моделях: описываются изменения, связанные с наследственными вариациями, рожденьями, выживанием, размножением и смертностью на сравнительно небольших временах, и утверждается, что биологическая эволюция сводится к большой последовательности таких изменений.

Со временем выживают те организмы, у которых коэффициент размножения больше. Но это не из-за какого-либо стремления к совершенству, а просто потому, что с ходом времени их доля в системе все увеличивается. При этом коэффициент размножения - просто коэффициент в уравнениях, описывающих изменение численности, а не цель совершенствования. Однако в результате появляется возможность рассматривать его как меру оптимальности. Такого рода динамические модели выживания ("модели дифференциального выживания") призваны объяснить

эволюцию. Такой подход (метод) дает возможность конструировать сценарии эволюционного развития и упорядочить данные биологического моделирования.

Идеи эволюционного моделирования применяются не только в биологии, но и при исследовании других научных объектов, например, в астрономии исследуют эволюцию Вселенной. Интересно, что явно свести всю эволюционную биологию к динамическим моделям отбора наследуемых вариаций невозможно и не нужно, но "мыслить все эволюционные события" происходящими "по Дарвину" (по современной синтетической теории) полезно, т. к. это способствует упорядочению и согласованию знаний.

Подчеркнем, что Дарвиновская теория эволюции исходила во многом из аналогии между процессами, происходящими в природе, и селекцией домашних животных и растений. Селекционер был заменен условиями обитания - от них зависят скорости успешного размножения. Живые существа, видоизменяясь, приспосабливаются к новым условиям. Впрочем, термин "приспособление" метафоричен и неточен, так как подразумевает, что изменения происходят целенаправленно и сознательно. В действительности же наиболее важен такой механизм: одним обитателям становится существенно хуже, другим - не настолько или даже лучше. И эти последние вытесняют прежних жителей, даже если вначале, до изменения в среде, были в меньшинстве. Называя такой процесс отбора приспособлением, мы как бы смешиваем его с другими - индивидуальной адаптацией и модификацией индивидуального развития.

При изменении условий все организмы, так или иначе, меняются - адаптируются. Это происходит довольно быстро, но в сравнительно узких пределах. Более широкий диапазон изменений дает модификация индивидуального развития. В новых условиях развитие идет хоть немного, но по другому. Для проявления таких модификаций требуется уже смена нескольких поколений. Эти изменения обычно обратимы и захватывают сразу значительную часть особей. Не вытеснение одних другими за счет изменений в коэффициенте размножения, а просто изменение многих - в этом отличие эволюции путем отбора наследуемых вариаций [Л. 77]. Он действует медленнее всех индивидуальных приспособлений и модификаций, но зато может приводить к "приспособлениям", намного сильнее уклоняющимся от первоначальных свойств.

Однако процесс эволюции не сводится к одному-двум эффектам. Более того, в биологии складывается такая проблемная ситуация, когда умножение знания о деталях не ведет автоматически к углублению понимания целого. В нынешней теории эволюции отсутствует процедура сбора целостной картины из разнообразных фрагментов. Здесь уместно перейти (вернуться) от биологии к нашей предметной области, т.е. к созданию эволюционных баз данных и знаний для синтеза интеллектуальных систем, и даже более того: к эволюции техносферы.

Когда речь идет об эволюции техносферы, то в действительности имеется в виду эволюция человеческой мысли и деятельности [Л. 77]. Человечество подошло вплотную к такому порогу, когда оно может запустить управляемую эволюцию техносферы. Функции человека, при этом, возможно, будут таковы: не придумывание шагов эволюции, а ее запуск, создание базы, рамок, основных правил. При этом, шаги эволюции такая техносфера будет делать уже сама. Отметим, что современное развитие вычислительной техники, кибернетики, математики и других наук позволяет говорить о такой эволюционной техносфере уже без "налета фантастики", а как о ближайшей перспективе. Важно, что, имея такую прикладную проблему в качестве одного из основных мотивов деятельности, теория эво-

люции получит новые стимулы к развитию, а моделирование эволюционной динамики - конкретные цели.

Как справедливо замечено в [Л. 77], время создания ЭВМ принципиально нового поколения уже пришло. Для таких ЭВМ надо ставить формализованные цели и организовывать "контакт с действительностью". Остальное такие ЭВМ, набираясь опыта, сделают сами, используя при этом эволюционные алгоритмы. Впрочем, адаптация имеет и свои минусы: трудно учитывать концептуальные ограничения, а без этого система может плохо работать в ситуациях, далеких от встречавшихся ранее. Для программируемых машин таких проблем не возникает. Если удастся соединить достоинства этих двух классов машин, то можно будет гибко сочетать программируемость и адаптацию. В частности, такой подход пытаются исследовать и реализовать в рамках создания новых методов обработки данных и конструирования ЭВМ [Л. 5-7, 37, 56, 63, 93, 126-131, 137, 148, 165-167, 189, 244, 245, 263, 274, 287, 343-344, 384-386, 438, 460-461], обучаемых ЭВМ и искусственного интеллекта [Л. 9, 11, 23, 28, 31, 33, 47, 82, 92, 112-118, 171, 203, 232-233, 251, 326-327, 427-430, 438, 462-465, 481-503], нейрокомпьютеров [Л. 17, 76-77, 142, 207-211, 333, 438]. Данная работа также посвящена аналогичным вопросам создания новых обучаемых и адаптирующихся комплексов ЭВМ.

Таким образом, смысл терминов, связанных с эволюцией, в различных областях одинаков, т.е. под этим понимают постепенное изменение системы в течение некоторого интервала времени. Однако, в каждой области знаний, термин "эволюция" имеет свое значение, но вместе с ним можно (как синоним) употреблять термин "адаптация".

## **1.5. НЕКОТОРЫЕ ПСИХОЛОГИЧЕСКИЕ АСПЕКТЫ СИНТЕЗА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ**

### **1.5.1. ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ПСИХОЛОГИЧЕСКИЕ АСПЕКТЫ САМООТНОШЕНИЯ**

Одной из основных проблем при разработке систем искусственного интеллекта является проблема наделения таких систем самосознанием, самооценкой, самоанализом, что в общем случае может быть обозначено в гносеологическом смысле как "самоотношение" СИИ к себе самой. До сих пор исследователи уделяли недостаточно внимания анализу влияния психологических аспектов самоотношения при разработке систем искусственного интеллекта. Особый интерес представляет то, как понятие "самоотношение" определяют при исследовании естественного интеллекта, прежде всего в психологии [Л. 106,153,223,259,265, 272, 273, 291, 427-459].

Рассмотрим основные представления о строении обобщенного самоотношения. Существует пять основных подходов к пониманию глобальной самооценки (самоуважения) и ее структуры [Л. 223]:

1. "Я" - как конгломерат частных самооценок, связанных с различными аспектами "Я-концепции".
2. Самоотношение как интегральная самооценка частных аспектов, взвешенных по их субъективной значимости.
3. Самоотношение как иерархическая структура, включающая частные самооценки, интегрирование по сферам личностных прояв-

лений и в комплексе составляющие обобщенное Я, которое находится наверху иерархии.

4. Глобальная шкала самооценки, которая относительно автономна и одномерна, т.к. выявляет некоторое обобщенное самоотношение, одинаково приложимое к различным сферам "Я".
5. Самоотношение как чувство в адрес "Я", включающее переживания различного содержания (самоуверенность, отраженное отношение и т.п.).

Кроме того, в настоящее время в психологии, в связи с термином "самоотношение", можно выделить два положения, которые, по мнению Панталева С.Р. [Л. 223], априорно принимаются большинством исследователей:

- существует некоторое обобщенное самоотношение (самоуважение, самооценка), которое является целостным, одномерным и универсальным образованием, выражающим степень положительности отношения индивида к собственному представлению о себе;
- это обобщенное самоотношение некоторым образом интегрируется из частных самооценок.

Интересно, что именно эти положения наименее эмпирически обоснованы, но это не приводит к сомнению в их априорной истинности. Считается, что самооценка есть эмоциональная реакция на знание о себе, так как ей просто больше неоткуда взяться, а обобщение этих эмоциональных реакций происходит на основе обобщения отдельных аспектов образа "Я" в целостную Я-концепцию.

При таком подходе, попытки понятийной реконструкции структуры эмоционально-оценочной системы не могут приводить ни к чему иному, как к "коллекционированию" (интегрированию) отдельных аспектов или сфер проявления личности, которые потенциально могут составить содержание "Я-концепции" и стать объектами для самооценок. Именно это и имеет место в большинстве зарубежных исследований [Л. 223].

Определенный интерес представляют взгляды на данную проблему, развиваемые в отечественной психологии (Леонтьев А.Н., Рубинштейн С.Л., Шорохова Е.В., Кон И.С., Столин В.В., Чеснокова И.И., Панталева С.Р.), где преодолены ограничения, связанные с позитивистскими и феноменологическими подходами к пониманию самосознания [Л. 106, 153, 223, 259, 265, 272, 273, 291, 427-459]. Однако решение методологических проблем пока не нашло практического воплощения в конкретные эмпирические исследования, непосредственно связанные с изучением строения самоотношения как эмоционально-оценочной системы. Самооценка чаще всего используется в практических исследованиях как независимая переменная.

Самоотношение (самооценку) нельзя рассматривать в отрыве от других понятий психологии. Наиболее разработанной в отечественной психологии является концепция самосознания, предложенная Столиным В.В. В этой концепции, самосознание понимается не только как самописание, самопознание или комплекс самооценок [Л. 272]. Самосознание личности направлено на то основное, что составляет ее психологическую сущность - на ее собственный личностный способ интеграции различных видов деятельности и иерархизации ее мотивов. Необходимость такого понимания самосознания в психологии связана с тем, что любой человек в какой-то период своей жизни реализует некоторую "совокупность жизненных отношений" с помощью системы различных видов деятельности. При этом одни и те же обстоятельства, действия и т.п. в различных жизненных отношениях будут иметь различный личностный смысл.

Одновременно осуществляя разные отношения с помощью разных видов деятельности, человек находится в потенциально конфликтной ситуации, в которой одни и те же действия могут обладать разными смыслами в отношении к разным мотивам. Этот противоречивый, составной смысл известный психолог Столин В.В. называет "конфликтным смыслом" [Л. 272]. Процесс самосознания по Столину происходит в виде переживания конфликтных смыслов, в ходе которого для личности (индивида) становится ясным, что она может преодолеть, что заставляет ее отступить и через какие преграды личность не может переступить даже под жестким давлением обстоятельств. Получается, что **собственное "Я" человека, иногда даже в большей степени, чем объективные обстоятельства, может выступать как условие, цель или мотив его деятельности.**

В этой концепции самоотношение не является ни следствием знания о себе, ни реакцией на определенные аспекты образа Я. Наоборот, и знание о себе, и самоотношение есть следствие одних и тех же общих причин, лежащих вне субъекта, в его деятельности, - и лишь вторично, в феноменологически превращенных формах, отдельные самооценки могут восприниматься самим субъектом как способные породить его истинное отношение к себе. Научный психологический анализ в этом случае не совпадает, - и даже в чем-то противоположен субъективной "психологике" интроспективного наблюдателя.

Как единица самосознания, смысл "Я" порождается как отношение к мотиву или цели релевантных их достижению качеств субъекта и оформляется в самосознании в значениях (когнитивный аспект) и эмоциональных переживаниях (эмоциональный аспект). В этом случае, смысл "Я" содержит когнитивную, эмоциональную и "отношенческую" (от слова - отношение) компоненты. С другой стороны, самоотношение может рассматриваться как результат столкновения собственных свойств с мотивами и целями субъекта.

Получаем, что термин самоотношение (т.е. отношение к себе) употребляется как минимум в трех следующих значениях [Л. 223].

1. Самоотношение как результат процесса соотнесения Я субъекта с мотивами и целями деятельности, т.е. "личностный" смысл "Я".
2. Самоотношение как субъективный, феноменологически представленный аспект смысла "Я", ограниченный связями последнего с другими составляющими сознания и замкнутый внутри него.
3. Самоотношение как эмоциональная составляющая смысла "Я".

Все три понимания: смысл "Я", аспект смысла "Я" и эмоции правомерны, но их терминологическая нераздельность, неразведенность осложняет их анализ.

Выделяют два основания различения смысловых образований от других образований сознания [Л. 223]:

- опосредованность реальной деятельностью первых - феноменологическая непосредственная данность вторых;
- глубинность первых и функционирование на поверхности сознания вторых.

В [Л. 223] введено еще одно основание для различения понятий "смысл Я" и "самоотношение". В отличие от первых двух собственно психологических, Пантеев С.Р. предложил использовать логико-гносеологическое разграничение понятий "смысл Я" и "самоотношение", а именно, рассматривать их как концептуальное и феноменологическое, как сущность и явление. Тогда самоотношение может быть определено (понято) как "лежащее на поверхности сознания, непосредственно-феноменологическое выражение (или представленность) личностного "смысла Я" для самого субъекта". При этом специфика переживания "смысла

"Я" производна от реального бытия субъекта, его объективной позиции в социуме. Кроме того, отсюда следует и то, что "смысл Я" может выражаться в не всегда адекватном ему самоотношении [Л. 223].

Самоотношение есть личностное образование, а потому его строение и содержание может быть раскрыто лишь в контексте реальных жизненных отношений субъекта и его деятельности, за которыми стоят мотивы, связанные с самореализацией субъекта как личности. Каждая конкретная социальная ситуация развития задает иерархию ведущих видов деятельности и соответствующих им основных мотивов и ценностей, по отношению к которым индивид осмысляет собственное "Я", наделяет его личностным смыслом.

Существует глубокое и содержательное различие двух подсистем самоотношения: самооценочной и эмоционально-ценностной, которые могут взаимно превращаться друг в друга. Выделение этих двух подсистем в составе самоотношения связано с конфликтной природой смысла "Я", с несовпадением "смыслообразующих" и "только побуждающих" мотивов. Иерархия этих мотивов и определенная социальная позиция субъекта и обуславливают, в конечном счете, ту конкретную обобщенную форму, в которой смысл "Я" оказывается презентированным (представленным) индивиду. При этом строение каждой из двух систем самоотношения также оказывается иерархическим. Но это разные иерархии. В их основе лежат различные принципы организации компонентов самоотношения в систему.

В большинстве случаев эти иерархии не совпадают (и иногда даже противоречивы), но и тогда их организация в целостную систему подчиняется принципу смысловой интеграции, в соответствии с которым смысловые отношения занимают более высокое место в иерархии и, в конечном счете, определяют обобщенное и устойчивое отношение субъекта к самому себе. Результаты проведенного анализа можно сформулировать в трех тесно связанных между собой положениях.

1. Самоотношение не сводимо к системе самооценок, а обладает более широким содержанием, связанным с выражением субъекту смысла "Я", который образуется через отношение (столкновение) особенностей и качеств индивида с мотивами ведущих видов деятельности и ценностями значимого социального окружения, которые заданы конкретной социальной ситуацией развития личности.
2. В составе самоотношения можно выделить две совместно функционирующие подсистемы: собственно самооценочную подсистему и подсистему эмоционально-ценностного самоотношения. Компоненты этих подсистем находятся в отношениях взаимного превращения. Тем не менее, выражение их содержания для субъекта может осуществляться в различных языках - самооценок и эмоциональных отношений - что делает соответствующие содержания феноменологически различимыми.
3. Основным принципом строения самоотношения как эмоционально-оценочной системы является принцип смысловой интеграции, в соответствии с которым компоненты самоотношения организованы в иерархическую систему. Причем эта иерархия является динамической и обусловлена конкретной социальной позицией субъекта, социальной ситуацией его развития и соответствующей иерархией видов деятельности, за которыми стоят те или иные смыслообразующие и побуждающие мотивы. Смысловые отношения занимают более высокое место в иерархии и определяют содержание обобщенного и устойчивого (для данной социальной ситуации развития) отношения субъекта к самому себе.

Приходим к выводу, что самоотношение является сложной системой и представляет собой личностное образование, которое может быть понято как непосредственно феноменологическое выражение (представленность) личностного "смысла Я" для самого субъекта. Кроме того, самоотношение представлено субъекту (личности) в специфическом языке, основу "алфавита" которого составляют следующие единицы, полученные Пантлеевым С.Р. [Л. 223] путем эмпирического обобщения частных аспектов самоотношения: чувство осознанности "Я"; самоуверенность; саморуководство; отраженное самоотношение; самооценочность; самопринятие; самопривязанность; самообвинение; внутренняя конфликтность. В целом, психологи в составе самоотношения выделяют две следующие подсистемы:

- а) собственно самооценочную и**
- б) эмоционально-ценностного самоотношения.**

Основным принципом строения самоотношения как эмоционально-оценочной системы является принцип смысловой интеграции. Компоненты самоотношения организованы в иерархическую систему. Изменения в структуре, содержании и уровне обобщенного самоотношения могут происходить только под воздействием факторов, затрагивающих его смысловой ("ядерный") слой. Значимое содержание обобщенного самоотношения и соответствующая чувственная модальность его выражения могут быть различны для индивидов с различным содержанием ведущих мотивов и ценностей. Таким образом, в психологии **самоотношение является сложной системой** и представляет собой личностное образование, которое может быть понято как непосредственно феноменологическое выражение (представленность) личностного "смысла Я" для самого субъекта. Кроме анализа самоотношения, несомненный интерес для синтеза интеллектуальных систем представляет понятие мотивации и её влияние на успешность деятельности. Рассмотрим этот вопрос более подробно.

### **1.5.2. ПСИХОЛОГИЧЕСКИЕ АСПЕКТЫ МОТИВАЦИИ И ЕЁ ВЛИЯНИЕ НА УСПЕШНОСТЬ ДЕЯТЕЛЬНОСТИ**

Мотивация представляет для нас интерес как ведущий фактор регуляции активности интеллектуальной системы, в психологии - личности, её поведения и деятельности, что взаимосвязано с рассмотренным выше самоотношением. Прежде всего, попробуем сформулировать само понятие мотивации и мотива.

В определениях мотива у современных психологов имеются некоторые расхождения [Л. 106, 153, 223, 259, 265, 272, 273, 291, 431-437]. Одни учёные под мотивом понимают психическое явление, которое становится побуждением к действию [Л. 437], другие – осознаваемую причину, лежащую в основе выбора действий и поступков личности [Л. 431].

Автор психологической концепции деятельности А. Н. Леонтьев считает, что мотив – это то, что, отражаясь в сознании человека, служит побуждением к деятельности и направляет её на удовлетворение определённой потребности. И при этом в качестве мотива выступает не сама потребность, а предмет потребности. А. Н. Леонтьев полагает, что предмет деятельности, являясь мотивом, может быть как вещественным, так и идеальным, но и в том и в другом случае за ним всегда стоит потребность, и он всегда отвечает той или иной потребности [Л. 431, с. 55]. Вслед за Реаном А. А., под мотивом мы будем понимать побуждение личности (интеллектуальной системы) к тому или иному виду активности (деятельность, общение, поведение), связанное с удовлетворением определённой потреб-

ности. Исходя из этого, будем считать, что в качестве мотива могут выступать идеалы, интересы, убеждения, социальные установки, ценности, за которыми, так или иначе, стоят разнообразные потребности личности. Согласно современным психологическим представлениям о мотивации (В. К. Вилюнас, В. И. Ковалёв, Е. С. Кузьмин, К. К. Платонов), мы рассматриваем мотивационную сферу личности как совокупность стойких мотивов, имеющих определённую иерархию и выражающих направленность личности [Л. 431, с. 5-6].

Для примера, рассмотрим широко распространённую учебную деятельность, которая есть и у людей, и в некотором смысле, должна быть у интеллектуальных систем. Как известно, успешность учебной деятельности человека зависит от многих социально-психологических и социально-педагогических факторов, кроме того, на успешность учебной деятельности очевидное влияние оказывает сила мотивации и её структура. Согласно закону Йеркса-Додсона, чем выше сила мотивации, тем выше результативность деятельности, но такая связь сохраняется лишь до определённого предела: если после достижения определённого оптимального уровня сила мотивации продолжает расти, то эффективность деятельности человека начинает падать.

Кроме того, мотивы бывают не только сильные и слабые, но и также внутренние и внешние. В последнем случае речь идёт об отношении мотива к содержанию деятельности. Если определённая деятельность значима для личности сама по себе, то мы имеем дело с внутренней мотивацией. В этом случае, когда значимы другие потребности (зарплата, подъём по служебной лестнице), то говорят о внешней мотивации. Нас интересуют именно внутренние мотивы. Деление мотивов на внутренние и внешние очень важно, так как, например, на познавательную мотивацию не распространяется закон Йеркса-Додсона, а это означает, что нарастание силы познавательной мотивации после оптимального уровня не приводит к снижению результативности учебной деятельности.

Однако каким бы значимым не было деление мотивов на внешний и внутренний, оно, тем не менее, является недостаточным. Сами внешние мотивы могут быть положительными (мотивы успеха, достижения) и отрицательными (мотивы защиты, избегания). Принято считать, что внешние положительные мотивы более эффективны, чем внешние отрицательные, даже если по силе (количественный показатель) они равны. Во многих случаях, по мнению многих учёных, вообще не имеет смысла дифференцировать мотивы по критерию «внутренние – внешние». Гораздо более продуктивен подход, основанный на выделении позитивных по своей сути и негативных мотивов [Л. 431, с. 58].

Очень часто в психологической литературе, говоря об учебной деятельности и её успешности, имеют в виду, прежде всего, влияние интеллектуального уровня личности. Однако А. А. Реан, протестировав по шкале общего интеллекта группу студентов, пришёл к выводу, что нет значимой связи интеллекта с успеваемостью ни по специальным предметам, ни по гуманитарному блоку дисциплин [Л. 432]. Этот факт подтвердился и в исследовании В. А. Якунина и Н. И. Мешкова [Л. 433], которые выявили, что «сильные» и «слабые» студенты всё-таки отличаются друг от друга, но не по уровню интеллекта, а по мотивации учебной деятельности. У сильных студентов отмечается внутренняя мотивация, так как у них есть потребность в освоении профессии на высоком уровне, они нацелены на получение знаний и умений для будущей профессиональной деятельности. У слабых студентов мотивы чаще всего внешние, а именно: избежать исключения, не лишиться стипендии и так далее.

Результаты, полученные в некоторых исследованиях по педагогической психологии, позволяют предполагать, что высокая позитивная мотивация может играть роль регулирующего фактора при недостаточно высоких специальных способностях или при сравнительно малом запасе знаний, умений и навыков у студентов; но интересен тот факт, что обратной связи у данного явления нет, то есть, никакой высокий уровень способностей не может компенсировать низкую учебную мотивацию или же её отсутствие, и, следовательно, не может привести к высокой успешности учебной деятельности. Например, при выполнении дипломной работы, высокая положительная мотивация к этой деятельности может компенсировать недостаточный уровень профессиональных знаний. Студенты, заинтересованные в конечном результате, изучают большее количество источников и проводят эксперимент на более высоком уровне с использованием современных методик, чем те, кто, имеет большой запас знаний, но низкую мотивацию к данной деятельности.

Таким образом, **от силы и структуры мотивации в значительной мере зависят и учебная активность** учащихся и их успеваемость. При достаточно высоком уровне развития учебной мотивации она может играть роль компенсаторного фактора в случае недостаточно высоких специальных способностей или недостаточного запаса у студентов и учащихся требуемых знаний, умений и навыков.

Кроме того, по мнению психологов, одним из основных факторов достижения успеха является характер мотивации. Деятельность результативнее в случае усвоения образца поведения «достижение успеха». Человек может действовать, смутно осознавая свои цели и желания, но тогда его действия едва ли будут целенаправленными. Действия плодотворны в том случае, если цель осознается: для чего, почему и каким образом действовать? Цель – это образ желаемого результата, итога предпринимаемых усилий. Многие исследователи полагают, что результат, итог деятельности «программируется человеком» еще до начала действий [Л. 434, с.227]. Результат сначала «идеален», существует как представление о цели.

Психический образ цели – это стимул, «пусковой» механизм активности. Предпринимая действия с самоориентацией на достижение положительного результата, человек предрекает успех в итоге больше, чем, если бы его воображение было задето образом неудачного результата. Таким образом, можно сделать следующий вывод: для успеха в какой либо деятельности требуется создать положительный образ результата. Психологи советуют избегать пессимистических мыслей, а также поддерживать в себе уверенность в благополучном завершении дела. Это – так называемая «надситуативная активность» личности. В исследованиях Ф. Хоппе ("успех и неуспех") были выявлены интересные факты: после серии успехов испытуемый ставит перед собой более сложную цель, повышает уровень притязаний, в то время как ряд неуспехов ведет к постановке более легкой цели, к снижению уровня притязаний [Л. 435, 436].

Какова же вероятность достижения успеха, например, для студентов или аспирантов? Какие факторы, детерминирующие успех, можно выделить по отношению к данной группе испытуемых. На наш взгляд, содержательная интерпретация «фактора успеха», который приводит субъекта к максимально положительному результату, весьма различна: основным фактором достижения успеха, бесспорно, являются знания, приобретенные субъектом в процессе обучения, важнейшим компонентом успеха является наличие способностей, опыта и умения применять их в той или иной ситуации, наличие благоприятных условий социальной среды, это может быть простое стечение обстоятельств. Например, студент, выучивший 5 билетов из 50, может вытащить один из 5-ти «счастливых» билетов и блестяще

ответить на экзамене. В этом случае успех достигается при относительно пассивном поведении за счет благоприятного стечения обстоятельств.

Отметим также, что одним из элементов успеха является поиск и включение в процессе достижения цели недостающего элемента, например, идеи или технологии (при написании дипломной или диссертационной работы). Кроме того, наличие цели и убежденности в необходимости ее достижения – один из факторов достижения успеха. Студент, имеющий хорошую подготовку и имеющий отличные способности, но поступивший в какое-либо учебное заведение по настоянию родителей, имеет меньше шансов на успех, чем тот, кто сознательно поставил себе цель овладеть данной профессией.

Таким образом, успех в учебной деятельности зависит от совокупности факторов; наличие или отсутствие каких-либо из них влияет на конечный результат и определяет вероятность успеха. Эти результаты психологов пока напрямую сложно приложить к области создания интеллектуальных систем, но в перспективе, это обязательно понадобится. Для себя сделаем вывод, что при обучении "самообучаемых" интеллектуальных систем очень важную роль играет мотивация такого обучения. Рассмотрим более подробно анализ влияния психологических мотивов на научную область, которая исследует синтез интеллектуальных систем, в том числе и, в перспективе, в целях создания систем искусственного интеллекта.

### 1.5.3. АНАЛИЗ ВЛИЯНИЯ ПСИХОЛОГИЧЕСКИХ АСПЕКТОВ НА СИНТЕЗ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Анализ влияния психологических аспектов самоотношения и мотивации деятельности при разработке систем искусственного интеллекта показал, что **самоотношение является важным основополагающим термином для моделирования действительно интеллектуальных систем**. Кроме того, для обучения интеллектуальных систем очень важную роль играет мотивация деятельности.

Любая система, претендующая на звание "интеллектуальной", должна иметь некоторую подсистему "самоотношения" (самоописания, самоанализа и т.п.) для выявления, фиксации своей мотивации деятельности, для формулирования своих целей и определения задач для их достижения. Если система не выделяет себя как нечто активное самостоятельное, то такая система не имеет своих целей и не может развиваться, а, следовательно, не может считаться интеллектуальной.

Моделирование самоотношения в СИИ представляется целесообразным реализовать на основе, изложенного более подробно ниже, миварного подхода к накоплению и обработке данных. Исходные данные для такого моделирования могут быть в различных форматах, но их надо обрабатывать в едином структурированном (миварном) пространстве унифицированного представления данных и правил.

Кроме того, понятия самоотношения и мотивации играют важную роль при разработке теоретических основ адаптивного синтеза конфигураций интеллектуальных многопроцессорных систем (ИМС). Использование психологических аспектов самоотношения является перспективным направлением в разработке систем искусственного интеллекта [Л. 427-430].

## 1.6. ПРОБЛЕМА АДАПТИВНОГО СИНТЕЗА ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ КОНФИГУРАЦИЙ

Обзор и анализ литературы, различных информационных материалов показал, что основные результаты и проблемы создания информационно-вычислительных конфигураций (ИВК), или другими словами: многопроцессорных вычислительных систем (МВС), к классу которых относятся и высокопроизводительные вычислительные комплексы (ВВК), отражены в работах Ершова А.П., Головкина Б.А., Каляева А.В., Корнеева В.В., Шпаковского Г.И., Амамии М., Танаки Ю., Ковалика Я.К., Коуги П.М., Терберга К. Дж. и др. [Л. 5-12, 16, 18, 20-22, 25-26, 30-40, 48, 54-57, 62-64, 72-75, 86, 88-90, 93-96, 105, 108-116, 118, 125-132, 135-138, 143-152, 154, 156-167, 189, 195-199, 203-212, 214, 224-228, 236-245, 248-253, 261-264, 274-279, 285-287, 294, 298-300, 310-313, 317-318, 328, 331, 333, 337, 343-344, 346-347, 357, 369-371, 384-386, 391, 393-395, 404-406].

История развития вычислительных средств и современная практика показывает, что всегда существовала и существует потребность во все более и более интеллектуальных и производительных вычислительных системах и компьютерах. Таким образом, проблема создания интеллектуальных систем и высокопроизводительных вычислительных комплексов является в настоящее время **актуальной**. Рассмотрим более подробно проблему создания системы адаптивного синтеза информационно-вычислительных конфигураций (САС ИВК), прежде всего в интересах повышения пиковой производительности ВВК

Повышение производительности ВВК в основном достигалось в результате эволюции технологий производства компьютеров [Л. 21, 34-35, 43, 64, 72-74, 146-152, 204-206, 245, 307, 412]. Наряду с этим имели место попытки использовать несколько процессоров в одной вычислительной системе в расчете на то, что будет достигнуто соответствующее увеличение производительности. Первой такой попыткой, осуществленной в начале 70-х годов, является ILLIAC IV. Отметим, что под параллельным компьютером понимают ЭВМ, состоящую из множества связанных определенным образом вычислительных блоков, которые способны функционировать совместно и одновременно выполнять множество арифметико-логических операций, принадлежащих одной задаче [Л. 10, 20, 148, 331].

В настоящее время существует много параллельных компьютеров (МВС и ВВК), моделей и проектов их реализации, которые описывают Бабаян Б.А., Барский А.Б., Белецкий В.Н., Бурцев В.С., Николаи Дж., Корнеев В.В., Дунаев В., Каляев А.В., Каратанов В.В., Левин В.К., Митрофанов В.В., Эйсымонт Л.К. и др. [Л. 21-22, 25, 34-35, 89-90, 93, 115-116, 126-131, 147-149, 209, 214, 369, 371, 404-405, 412]. По самым оптимистическим прогнозам тактовые частоты современных и перспективных СБИС могут быть увеличены в обозримом будущем до 5 ГГц. В то же время, исходя из анализа информационных материалов [Л. 115-116, 126-131, 135, 311, 318, 328, 331, 333, 384-386, 369, 371, 391, 393-394, 404-406, 410, 412], следует, что достигнутая степень интеграции, позволяет строить параллельные системы, в которых число процессоров может достигать десятков тысяч.

Таким образом, в области повышения производительности вычислительных систем резерв технологических решений ограничивается одним порядком, а освоение массового параллелизма и новых архитектурных решений содержит резерв повышения производительности на несколько порядков. Однако увеличение степени параллелизма вызывает увеличение числа логических схем, что сопровождается увеличением физических размеров, в результате чего возрастают задерж-

ки сигналов на межсоединениях. Этот фактор приводит либо к снижению тактовой частоты, либо к созданию дополнительных логических ступеней и, в результате, к потере производительности. Рост числа логических схем также приводит к росту потребляемой энергии и отводимого тепла.

Кроме того, следует подчеркнуть, что более высокочастотные логические схемы при прочих равных условиях потребляют большую мощность на один вентиль [Л. 115-116]. В результате возникает теплофизический барьер, обусловленный двумя факторами: высокой удельной плотностью теплового потока, что требует применения сложных средств отвода тепла, и высокой общей мощностью системы, что вызывает необходимость использования сложной системы энергообеспечения и специальных помещений.

Другим фактором, влияющим на архитектуру высокопроизводительных вычислительных систем, является взаимозависимость архитектуры и алгоритмов задач. Этот фактор часто приводит к необходимости создания проблемно-ориентированных систем, при этом может быть достигнута максимальная реальная (фактическая, а не пиковая) производительность для данного класса задач.

В настоящее время синтез конфигураций новых поколений ЭВМ приобретает особую актуальность, что обусловлено постоянно возрастающими потребностями решения различных уникальных, сложных и специальных задач (ССЗ) (рис. 1.6.1).

Кроме того, актуальность создания именно адаптивных ЭВМ обусловлена необходимостью постоянной модернизации компьютеров в условиях непрерывности их функционирования. Современные ЭВМ могут содержать тысячи вычислительных блоков, одновременная замена которых требует больших финансовых и временных затрат.

Создание адаптивных ЭВМ позволит изменить процесс проектирования многопроцессорных вычислительных систем, а также обеспечит возможность наращивания и проведения поэтапной модернизации модулей многопроцессорных вычислительных систем в условиях эксплуатации.

В настоящее время, в области создания суперЭВМ возникло противоречие, которое состоит в том, что для любой задачи наиболее эффективными являются специализированные устройства, но их производство оказалось в новых условиях экономически нецелесообразным [Л. 43, 93, 112-116, 148, 154, 156, 206, 209, 245, 371, 391, 404-405]. Поэтому, теперь многопроцессорные вычислительные системы, суперЭВМ и высокопроизводительные вычислительные комплексы собирают из множества унифицированных модулей, производимых различными фирмами (рис. 1.6.2). Более того, появилась возможность самостоятельной сборки суперЭВМ, а один из таких высокопроизводительных вычислительных комплексов уже вошел в 2001 году в "ТОР-500".

Отметим, что в данной работе термины "конфигурации компьютерных систем" (ККС) и "информационно-вычислительные конфигурации" (ИВК) являются синонимами, т.е. могут употребляться вместо друг друга.

**Актуальность проблемы адаптивного синтеза информационно-вычислительных конфигураций** обусловлена изменением принципов создания ВВК, что выявило новые научные противоречия и проблемы, определило необходимость нетрадиционных подходов к их решению. Это требует разработки новых теоретических основ создания многопроцессорных вычислительных систем и высокопроизводительных вычислительных комплексов, и адаптивный синтез конфигураций компьютерных систем является одним из наиболее перспективных направлений исследований (рис. 1.6.3).

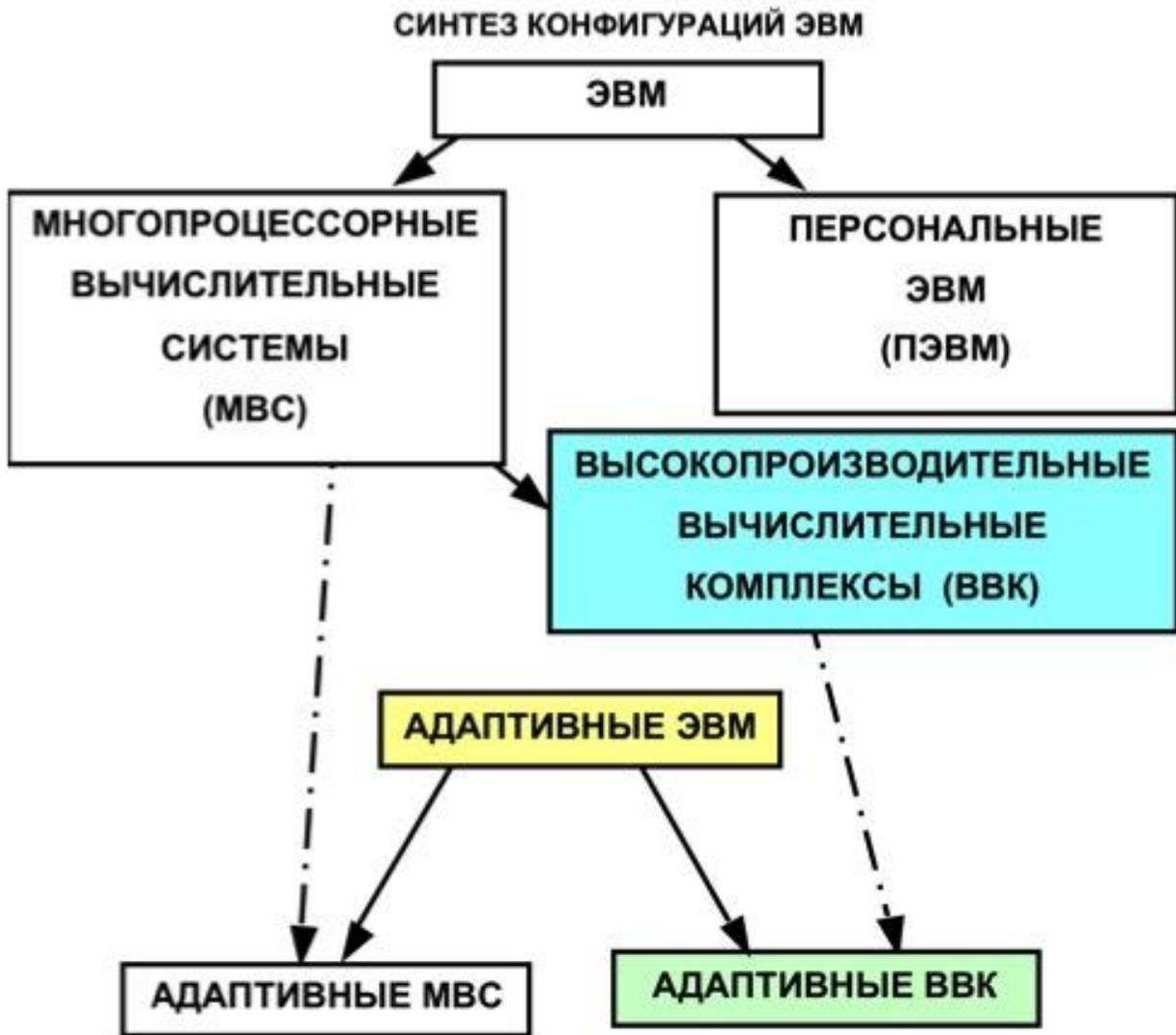


Рис. 1.6.1. Роль и место адаптивного синтеза ЭВМ.

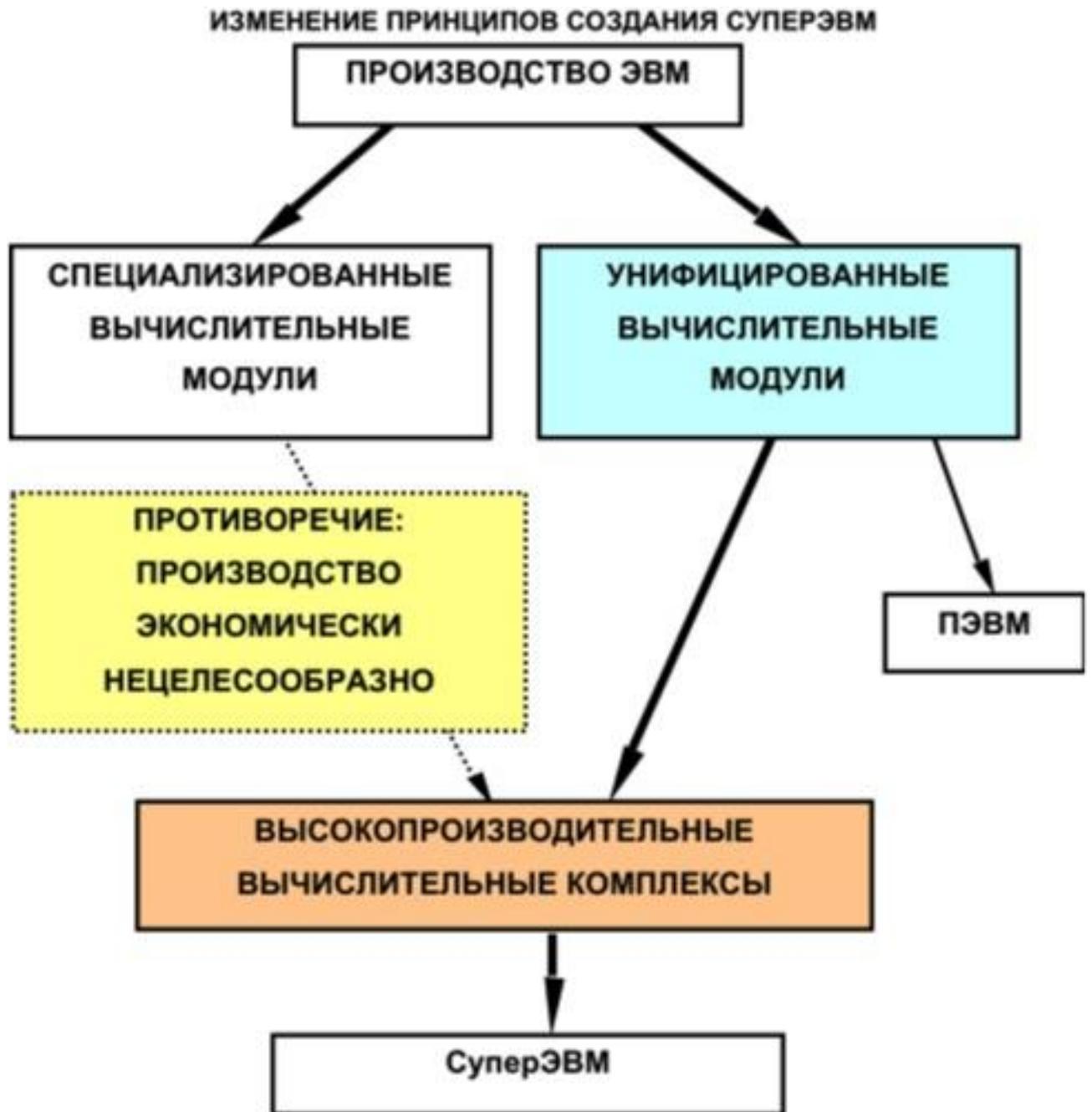


Рис. 1.6.2. Противоречие в области создания ВВК.

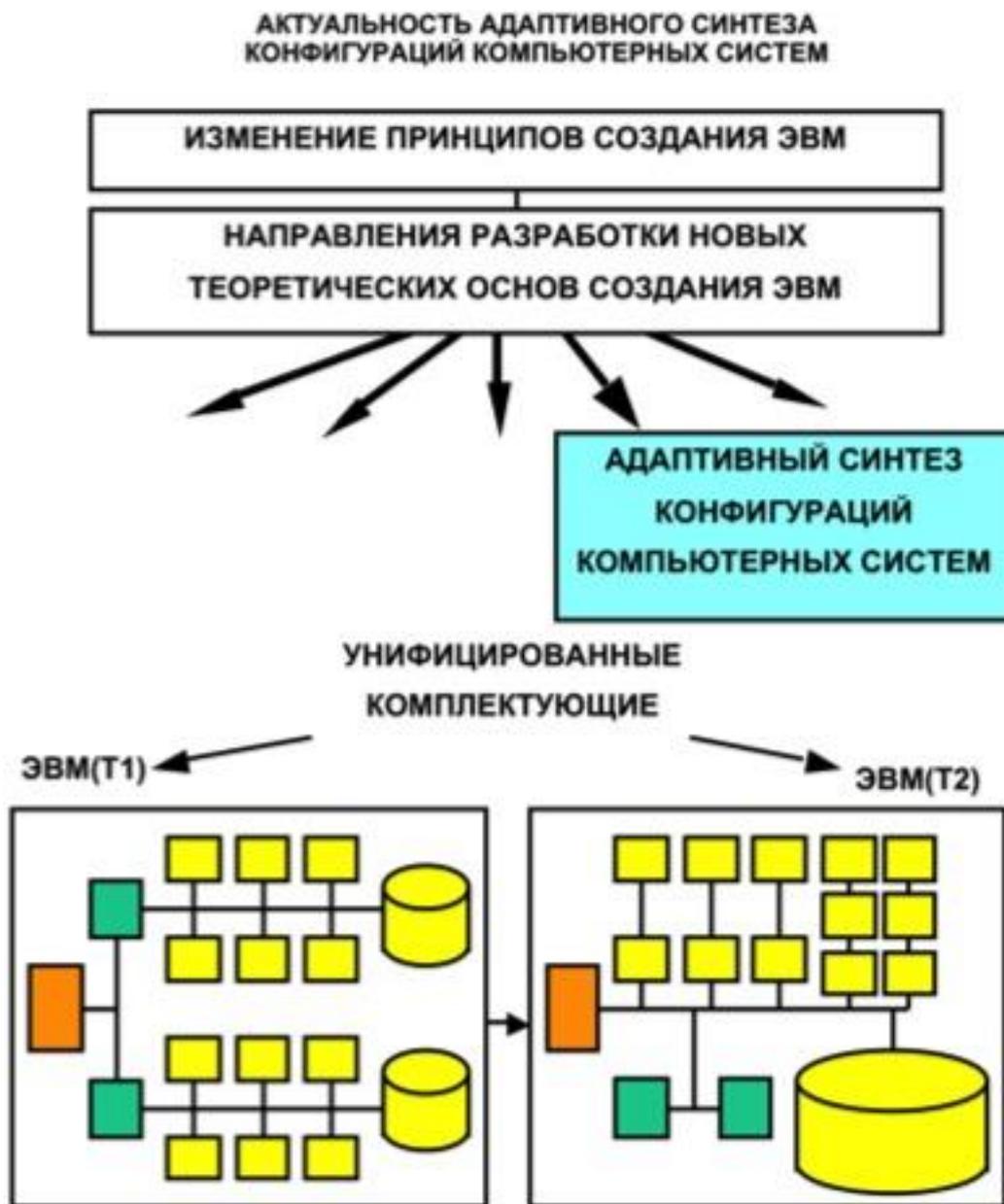


Рис. 1.6.3. Актуальность адаптивного синтеза информационно-вычислительных конфигураций.

Адаптивность синтеза ИВК (ККС) обусловлена:

- уникальностью каждой задачи;
- необходимостью обучения и учета, ранее разработанных вариантов конфигураций;
- быстротой создания, развития и старения программных и аппаратных средств.

Наиболее сложные и специальные задачи могут быть отнесены к классу ресурсоемких научно-практических задач оперативной диагностики. Как правило, это уникальные задачи, решение которых носит эмпирический характер и требует научно-обоснованного синтеза специализированных ВВК.

Современное развитие науки и техники позволяет из унифицированных комплектующих разного типа на необходимый для решения задачи период времени собрать неоднородный ВВК, который после решения этой задачи может быть адаптирован для решения других задач. Следовательно, закупать можно только самые современные комплектующие, добавляя их в новые конфигурации многопроцессорных вычислительных систем.

Методы синтеза отдельных ЭВМ известны [Л. 10, 21-22, 25, 34-35, 39, 43-46, 56, 62-64, 72-75, 94-96, 112-116, 118, 126-131, 135, 146-152, 164-167, 195, 197-198, 203-211, 224, 255, 261, 279, 287, 307, 313, 369-371, 404-406, 412], но проблема адаптивного синтеза конфигураций ВВК не решена.

Для проектирования требуемых адаптивных конфигураций ВВК необходимо создать систему адаптивного синтеза информационно-вычислительных конфигураций (САС ИВК) (рис. 1.6.4).

В САС ИВК должна оперативно накапливаться информация о решаемых задачах и о существующих компонентах ВВК. САС ИВК может использовать различные источники информации, включая Интернет. Исходные данные для синтеза могут быть в различных форматах, но их надо накапливать и обрабатывать в едином структурированном пространстве унифицированного представления данных и правил.

Методы синтеза ЭВМ могут быть представлены в виде логических систем, поэтому, в смысле логической обработки данных, САС ИВК можно отнести к классу автоматизированных систем сбора и обработки информации (АССОИ).

В этой области работали многие специалисты [Л. 1-13, 15-28, 30-48, 54-57, 62-66, 70-80, 86-98, 100-118, 120-132, 135-152, 154-167, 170-176, 178, 183-185, 189-216, 224-228, 231-245, 247-253, 255-258, 261-264, 268-271, 274-280, 284-300, 304-311, 313-319, 321-328, 342-426], но результаты этих исследований не достаточны для создания систем адаптивного синтеза, так как ранее не была учтена совокупность следующих условий, определяющих адаптивность синтеза.

Для обеспечения живучести и надежности система адаптивного синтеза должна быть многопроцессорной. Накопление информации должно быть адаптивным, универсальным и единым для разных предметных областей.

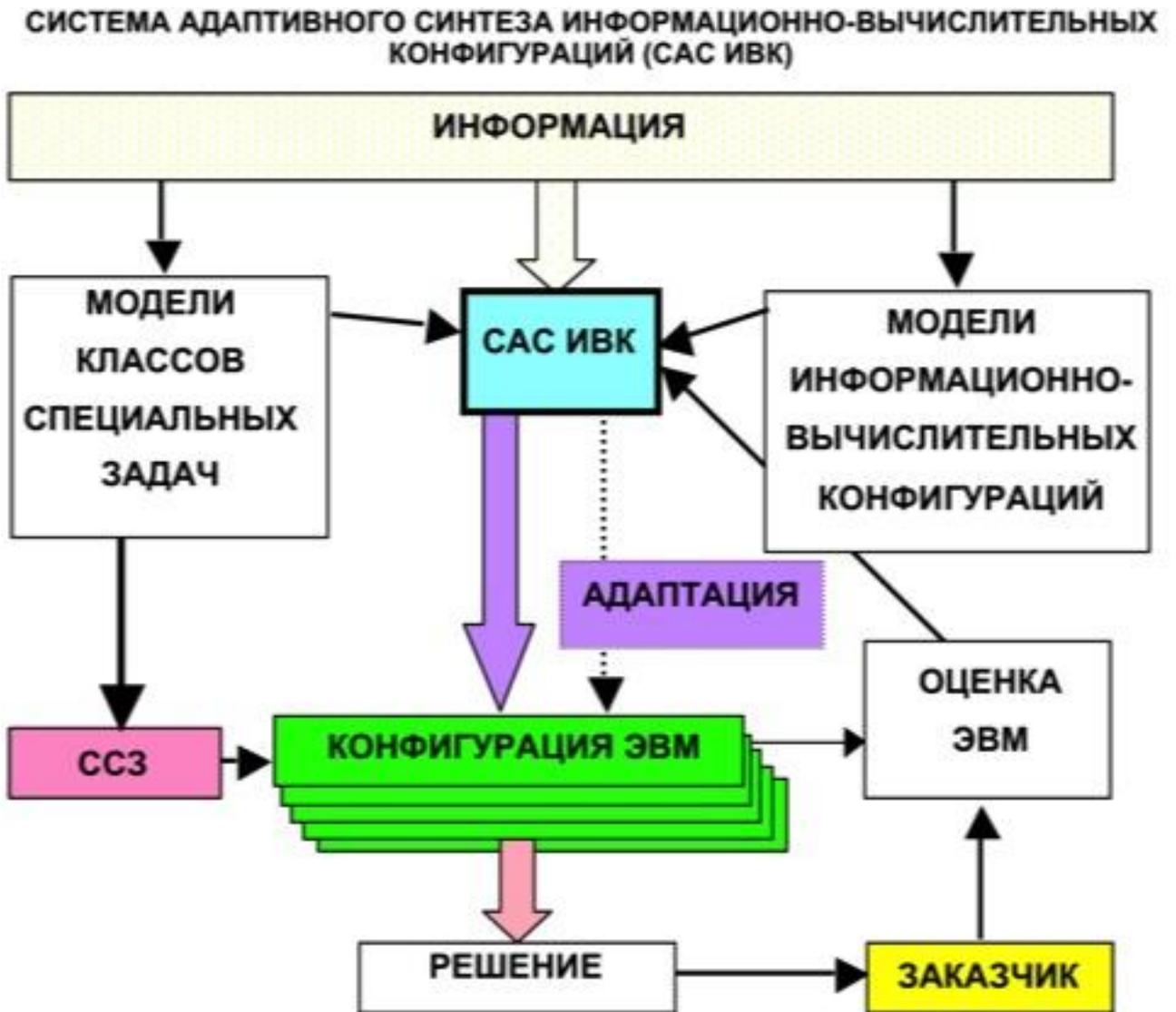


Рис. 1.6.4. Структура системы адаптивного синтеза информационно-вычислительных конфигураций (САС ИВК).

Для своевременного реагирования на изменения предметной области, САС ИВК должна активно управлять процессом сбора необходимой исходной информации на основе логической сети правил, управляемой потоком данных. Функционирование САС ИВК должно быть непрерывным, следовательно, новые данные должны вводиться без перезагрузки и перепланирования баз данных. Таким образом, САС ИВК должна быть многопроцессорной адаптивной неоднородной АССОИ, в которой можно в условиях эксплуатации добавлять, изменять и удалять любые модули, средства и данные.

Для создания САС ИВК необходимо решить следующую сложную наукоемкую проблему. **Научная проблема - создание эволюционных баз данных и знаний (правил), разработка теоретических основ адаптивного синтеза конфигураций компьютерных систем и методов оперативной обработки информации при решении сложных и специальных задач.**

Решение этой научной проблемы, как будет доказано в последнем разделе, позволит достичь следующих целей:

- 1) обеспечить возможность адаптивного синтеза эволюционных интеллектуальных систем, в том числе ЭВМ и ВВК,
- 2) повысить эффективность и оперативность решения сложных и специальных задач,
- 3) снизить финансовые затраты на создание МВС и ВВК при одновременном увеличении сроков его эксплуатации.

Как было показано выше, отдельные методы синтеза интеллектуальных систем известны. Кроме того, уже разработаны методы создания ЭВМ с программируемой, изменяемой под структуру задач архитектурой. Известны и методы решения специальных и сложных задач. Вообще, может показаться, что проблема создания эволюционных баз данных и правил не связана с проблемой создания САС ИВК, но это не совсем так. Конечно, на первый взгляд, это две различные проблемы, но это не так. Уже при современном развитии науки и техники, представляется возможным в автоматизированном режиме синтезировать различные информационно-вычислительные конфигурации.

Если же нам удастся решить проблему создания эволюционной, непрерывно развиваемого во времени программного обеспечения, которое позволит накапливать и обрабатывать неограниченные объемы информации (эволюционные базы данных и знаний), то ничего не будет мешать нам сделать следующий логичный шаг по созданию развиваемого во времени, перенастраиваемого программно-аппаратного комплекса с изменяемой, эволюционной структурой. Ведь на уровне "железа" все проблемы по созданию неоднородных многопроцессорных комплексов с изменяемой структурой практически решены: уже сейчас можно изменять состав и структуру практически любого ВВК, при соблюдении условия совместимости оборудования. Проблема совместимости оборудования решается на уровне создания различных стандартов, интерфейсов. В данной книге проблемы "железа" не обсуждаются, так как мы считаем, что они либо уже решены, либо могут быть решены в ближайшее время.

Главное, чего не хватает на сегодняшний день, это проблема создания эволюционного программного обеспечения (на основе эволюционных баз данных и правил), поэтому именно этой проблеме и посвящена основная часть данной монографии.

Конечно, проблема создания эволюционных баз данных и знаний играет огромную самостоятельную роль и вполне могла бы стать темой отдельной книги, но ... в современных российских условиях (весьма сложных и тяжелых для науч-

ной деятельности), выпуск одной книги - это уже практически чудо, а писать отдельно вторую книгу и дожидаться ее выхода не целесообразно. И зачем ждать, если основные результаты и проблемы создания САС ИВК уже известны и могут быть решены на основе эволюционных баз данных и знаний.

Таким образом, актуальная, сложная научная проблема, решаемая в данной работе, может быть разделена на две взаимосвязанные части:

- 1) разработка теоретических основ адаптивного синтеза конфигураций компьютерных систем (рис. 1.6.5, часть № 1),
- 2) разработка методов оперативной обработки информации при решении сложных и специальных задач (рис. 1.6.5, часть № 2).

Итак, прежде всего, необходимо разработать теоретические основы адаптивного синтеза конфигураций компьютерных систем на основе создания миварного (информационного) пространства данных (рис. 1.6.5, задача № 1). Для этого необходимо создать теоретические основы создания эволюционных (адаптивных) баз данных и правил (т.е. миварного информационного пространства). Затем целесообразно создать принципы построения и применения систем адаптивного синтеза конфигураций компьютерных систем и разработать новый метод обработки данных в САС ИВК. Разработка этого нового метода по существу представляет собой метод обработки данных на основе применения логической сети правил (миварной), управляемой потоком данных (рис. 1.6.5, задача № 2).

Кроме того, представляется необходимым исследовать вопросы повышения оперативности обработки данных и разработать: метод быстрого поиска маршрута логического вывода (рис. 1.6.5, задача № 3), метод максимального распараллеливания доступа к общей базе данных (рис. 1.6.5, задача № 4) и линейный метод единично-инкрементного суммирования чисел (рис. 1.6.5, задача № 5), которые более подробно будут описаны ниже.

Еще раз, но более подробно, обсудим исследуемую нами научную проблему. Начнем с синтеза ВВК. Фактически, при традиционном подходе к синтезу информационно-вычислительных конфигураций и разработке ВВК, под каждую "специальную" задачу необходимо создавать проблемно-ориентированный ВВК, однако всегда имеют место финансовые и другие ограничения, которые не позволяют произвольно наращивать вычислительную мощность. В настоящее время этот "специализированный" подход практически исчерпал свои возможности. В то же время, современное состояние науки и техники позволяет осуществить **новый подход, когда из унифицированных комплектующих разного типа на необходимый для решения задачи период времени собирается неоднородный ВВК, который после решения этой задачи может быть адаптирован, переконфигурирован под решение другой задачи.**

Таким образом, из универсальных комплектующих собирается специализированный ВВК, а за счет его дальнейшей адаптации и переконфигурации представляется возможным оптимально сочетать требования универсальности и уникальности. При этом, адаптацию и переконфигурацию компьютерной системы можно осуществлять по разному: заменой или изменением вычислительных модулей, коммутационной системы ("переконмутацией"), модификацией программных средств и т.п.

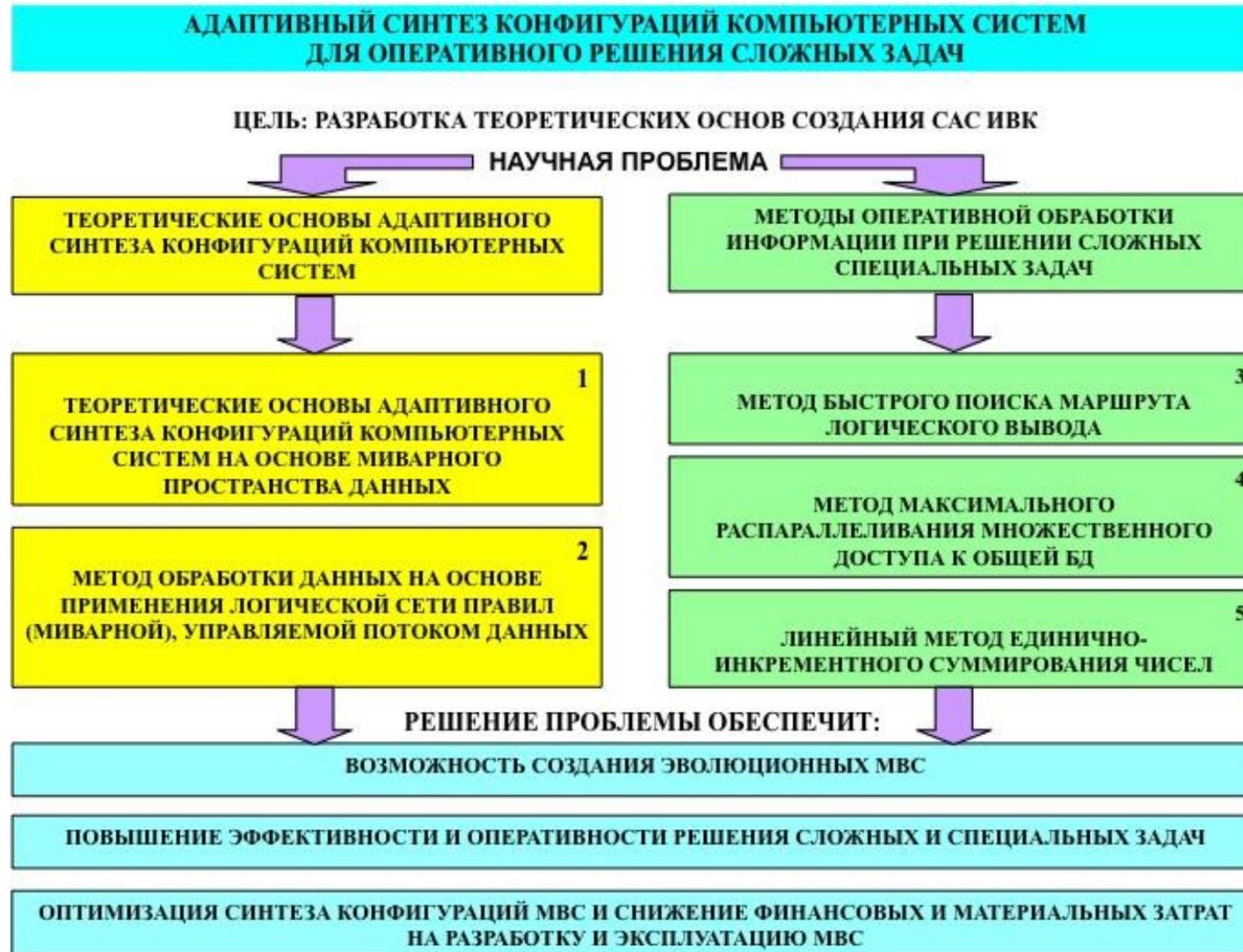


Рис. 1.6.5. Структура решения проблемы создания САС ИВК

Отметим, что особый интерес, с точки зрения адаптации ЭВМ под различные классы задач, представляют разрабатываемые под руководством академика Каляева А.В. вычислительные системы и модули с программируемой под структуру решаемой задачи архитектурой [Л. 126-131, 384-386].

При реализации адаптивного синтеза на вычислительных системах с программируемой архитектурой представляется возможным создать адаптивную многопроцессорную вычислительную систему, которая будет решать в автоматизированном режиме различные классы задач с большей эффективностью, чем существующие сегодня ЭВМ. В такой адаптивной системе (гипотетической) создается база данных известных структур задач и описание (каталог) архитектур вычислительных модулей. При поступлении на вход системы некоторой задачи производится ее анализ, определяется ее структура и из каталога архитектур вычислительных модулей выбирается наиболее адекватный модуль, затем производят перепрограммирование архитектуры вычислительных модулей и непосредственное решение требуемой задачи.

Если для поступившей на вход адаптивной системы задачи в каталоге архитектур существует оптимальный вариант архитектуры вычислительных модулей, то все это может быть реализовано в автоматическом режиме. Если в каталоге нет подходящей архитектуры, то формируется сигнал вызова администратора (человека) и система переходит в автоматизированный режим, когда человек сам принимает решение о выборе "подходящей" архитектуры или формирует новую архитектуру и добавляет ее в общий каталог.

Как известно [Л. 112-116, 126-131], универсальные ЭВМ решают реальные задачи с эффективностью 10-15 % от пиковой производительности, в то время как специализированные ЭВМ позволяют добиться 80-90 % эффективности. Внедрение адаптивного синтеза на вычислительных системах с программируемой архитектурой позволит создать достаточно универсальную адаптивную компьютерную (программно-аппаратную) систему, которая будет решать широкий класс реальных вычислительных задач с эффективностью не менее 60-70 % от пиковой производительности вычислительной системы. В более общем случае, данные адаптивные системы можно рассматривать как эволюционные компьютерные системы, которые будут рассмотрены ниже.

После внедрения и реализации эволюционных систем (МВС и ВВК) закупать можно будет только самые современные комплектующие, добавляя их в новые эволюционные (модифицируемые, адаптируемые) информационно-вычислительные конфигурации.

Таким образом, в настоящее время изменились как сами задачи, так и возможности по созданию ВВК, что выявило новые научные проблемы и определило необходимость нетрадиционных подходов к их решению. Сегодня созданы все предпосылки для решения научной проблемы адаптивного синтеза информационно-вычислительных конфигураций и построения реальных эволюционных компьютерных систем.

Теперь мы можем перейти к описанию основных путей решения нашей научной проблемы и более формальной постановке основных задач по исследованию возможностей создания эволюционных (адаптивных) баз данных и знаний для интеллектуальных систем и их внедрению в системах адаптивного синтеза информационно-вычислительных конфигураций, а также и для других автоматизированных систем обработки информации.

Рассмотрим более подробно основные пути создания эволюционных баз данных и знаний для синтеза интеллектуальных систем.

## 1.7. ПУТИ СОЗДАНИЯ ЭВОЛЮЦИОННЫХ БАЗ ДАННЫХ И ЗНАНИЙ ДЛЯ СИНТЕЗА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Анализ возможных путей построения эволюционных (адаптивных) баз данных и знаний, исследование и оценка известных моделей и баз данных, выявили необходимость обобщения существующих и разработки новых принципов построения систем представления данных.

В этой работе предложены теоретические основы (принципы) построения объектно-структурного динамически многомерного пространства унифицированного представления данных и правил, которые в основном удовлетворяют требованиям к созданию интеллектуальных многопроцессорных систем. Это пространство представления данных и правил (т.е. информационное пространство) получило для краткости наименование - "миварное пространство". Так как новый подход к представлению данных существенно расширяет возможности по универсальности и оперативности обработки информации в автоматизированных системах обработки информации, в том числе и в интеллектуальных МВС, то необходимо разработать новый метод "быстрой" обработки данных на основе применения миварного пространства и построения адаптивной логической сети правил, управляемой потоком данных.

Для повышения быстродействия и оперативности обработки данных в АСОИ путем использования новых возможностей миварного пространства представления данных и правил необходимо разработать новый метод быстрого поиска маршрута вывода на основе использования логической сети правил. Суть этого метода в преобразовании логической сети правил миварного пространства в многополюсную сеть теории графов и поиска ее минимального разреза. При этом вычислительная сложность поиска маршрута логического вывода от полного перебора сводится фактически к вычислительной сложности поиска минимального разреза.

Анализ известных в теории графов алгоритмов поиска минимального разреза многополюсных сетей и дополнительные исследования [Л. 41-46, 225] позволили значительно повысить оперативность обработки данных и разработать квадратичной сложности метод поиска минимального разреза многополюсных сетей.

Оценка возможностей известных методов распараллеливания множественного доступа к общей базе данных выявила необходимость решения проблемы обеспечения реального параллельного доступа к базе данных для обеспечения оперативности потоковой обработки данных в САС ИВК.

На основе результатов проведенных исследований [Л. 45], представляется возможным разработать метод максимального распараллеливания множественного доступа к общей базе данных в условиях недопущения взаимного искажения используемых данных на основе построения виртуальных потоковых баз данных.

Анализ дополнительных возможностей повышения оперативности обработки информации в САС ИВК выявил возможность существенного ускорения обработки при решении некоторых классов переборных задач. На основе результатов проведенных исследований [Л. 225], представляется возможным разработать новый метод (способ) алгоритмической минимизации количества вычислительных процедур и устройств сложения, позволяющий перейти от степенной к линейной зависимости общего количества операций сложения от разрядности чисел.

Таким образом, проведенные исследования и полученные в ходе них новые научные результаты должны позволить разработать теоретические принципы и методы адаптивного синтеза конфигураций компьютерных систем на основе при-

менения миварного пространства представления данных и правил и адаптивной логической сети обработки данных, а реализация метода быстрого поиска маршрута вывода путем разреза многополюсных сетей, метода максимального распараллеливания доступа к БД и способа суммирования чисел позволит обеспечить максимально возможную оперативность функционирования вычислительных средств и анализа сложных, специальных и уникальных диагностических задач.

Рассмотрим **новизну, актуальность и сложность** поставленных научных задач более подробно. Прежде всего, отметим, что создаваемая в миварном информационном пространстве адаптивная активная логическая сеть обработки, управляемая потоком данных должна объединять и обработку известных данных на основе известных правил, и одновременно осуществлять обучение, изменение системы правил со своевременным внесением изменений в соответствующие процессы обработки. Это позволит обеспечить эволюционное, синергетическое развитие автоматизированной интеллектуальной системы обработки информации, т.е. всей компьютерной системы в целом.

Разработка принципов построения и применения систем адаптивного синтеза (САС) информационно-вычислительных конфигураций (ИВК) для оперативного анализа и решения сложных, специальных и уникальных диагностических задач (УДЗ) является сложной наукоемкой проблемой.

Исследуемая в этой работе научная проблема имеет важное **практическое значение** для целого ряда областей, наиболее сложные задачи которых могут быть отнесены к классу задач оперативной диагностики. Как правило, это уникальные задачи, решение которых носит эмпирический характер и требует научно-обоснованного синтеза специализированных конфигураций компьютерных систем.

**Актуальность** проблематики адаптивного синтеза ИВК обусловлена тем, что системы автоматизированного проектирования (САПР) вычислительных конфигураций, созданные для массового обслуживания и решения типовых задач не применимы в данном случае.

**Противоречие** в том, что в условиях дефицита времени требуется оперативно синтезировать уникальный программно-аппаратный комплекс (ПАК) с использованием всех доступных ресурсов, включая Интернет, для обеспечения экспресс - диагностики сложных уникальных специальных задач. Как правило, это NP-полные задачи, которые в зависимости от конкретной ситуации, могут быть сведены к набору полиномиальных задач, решаемых за обозримое допустимое время с учетом конкретных ограничений. Отметим, что при наличии нескольких вариантов и некоторого резерва времени в качестве критерия эффективности выбора конфигурации компьютерной системы, целесообразно использовать отношение достигаемого эффекта к стоимости компьютерной системы.

**Адаптивность синтеза информационно-вычислительных конфигураций обусловлена** уникальностью каждой диагностической задачи, необходимостью учета ранее разработанных вариантов конфигураций, быстротой создания, развития и старения программных и аппаратных средств. Для синтеза требуемых конфигураций фактически необходимо разработать некоторую специализированную САПР, оперативно использующую различные источники информации.

Исходные данные для САС ИВК могут быть в различных форматах представления данных и правил, которые необходимо хранить в едином структурированном универсальном пространстве представления данных и правил (миварном информационном пространстве) с целью научно-обоснованного всеобъемлющего синтеза уникальных конфигураций вычислительных систем, а также для оперативного решения сложных и специальных задач, в том числе и уникальных.

Отметим, что система адаптивного синтеза информационно-вычислительных конфигураций - САС ИВК - сама по себе может являться ядром синтезируемого программно-аппаратного комплекса оперативной диагностики (решения уникальных познающих и диагностических задач).

**Научная проблема создания САС ИВК** относится к управлению в технических системах, в частности, к области самооценки, саморазвития и адаптации компьютерных систем.

В научном плане эта проблема формулируется следующим образом:

**создание адаптивного синергетического (саморазвивающегося) программно-аппаратного комплекса активной обработки потоков входных данных на основе пространства унифицированного представления данных и правил, управляемых потоком данных, в условиях дефицита времени одновременного анализа внезапно возникающих сложных задач обучения и распознавания.**

Таким образом, как уже было отмечено выше, проблема заключается в разработке теоретических основ адаптивного синтеза конфигураций компьютерных систем для оперативного анализа сложных и уникальных диагностических задач на основе технологий самообучения, самооценки, автоматизированного проектирования, распределенных баз данных и знаний.

**Практический эффект** (польза) решения данной проблемы состоит в том, что внедрение результатов нашей работы позволит повысить эволюционные способности компьютерных систем путем активного формирования запросов и ресурсов для выживания и развития в изменяющейся агрессивной внешней среде.

Итак, для достижения цели создания эволюционных (адаптивных) баз данных и знаний для синтеза интеллектуальных систем путем создания систем адаптивного синтеза информационно-вычислительных конфигураций - САС ИВК - необходимо решить следующие задачи:

- 1) разработать принципы построения и применения САС ИВК на основе создания эволюционных баз данных, т.е. объектно-структурного динамически многомерного информационного пространства представления данных и правил ("миварного");
- 2) разработать метод обработки данных на основе применения "активной" логической сети правил, управляемой потоком данных;
- 3) разработать метод быстрого поиска маршрута логического вывода на основе построения многополюсной сети и поиска ее минимального разреза;
- 4) разработать метод максимального распараллеливания потоковой обработки взаимозависимых данных путем построения виртуальных потоковых баз данных;
- 5) разработать способ алгоритмической минимизации необходимого количества вычислительных процедур и устройств сложения для решения отдельных сложных диагностических задач.

Сформулируем и рассмотрим **основные пути** решения этих научных задач, в интересах **создания эволюционных баз данных и знаний для синтеза интеллектуальных систем**, немного подробнее.

**1. Разработка теоретических основ адаптивного синтеза конфигураций компьютерных систем.** Прежде всего, необходимо разработать принципы построения и применения САС ИВК, а также создать теоретические основы эволюционных, адаптивных баз данных путем построения эволюционного пространства унифицированного представления данных и правил. К основным принципам построения и применения САС ИВК могут быть отнесены: адаптивность, модуль-

ность, непрерывность, возможность глобального накопления и обработки в едином унифицированном формате любых данных и правил, максимальная оперативность обработки, активное формирование требований по изменению своих ресурсов и конфигурации, независимость от конкретных технических средств и открытость системы.

Для создания эволюционных, адаптивных баз данных необходимо обобщить все традиционные структуры представления данных в виде пятиуровневой одномерной таблицы представления данных (ОТПД-5). Затем, необходимо создать эволюционные динамические многомерные структуры представления данных, в которых могут изменяться не только значения переменных, но и количество осей пространства, т.е. сама структура представления данных. Таким образом, предлагается рассмотреть новый класс моделей данных с адаптивной, эволюционной (изменяемой, развивающейся, наращиваемой) структурой. Эволюционные динамические многомерные структуры унифицированного представления данных для краткости и однозначности обозначения названы "миварными" структурами.

Суть (основной путь создания) миварного представления данных в следующем. Формируется минимальное пространство представления данных путем определения основных осей и фиксации основных объектов и их отношений. Названия объектов и отношений в некотором порядке фиксируются на осях, а на соответствующих пересечениям точках этого пространства, которые называются миварами, записываются конкретные значения свойств и отношений объектов. В дальнейшем, при появлении новых данных, они накапливаются при неизменной структуре, или происходит изменение структуры путем добавления, изменения или удаления любого отдельного мивара, любой точки на любой оси пространства или даже самой оси. Важной особенностью миварного подхода является то, что отношения объектов хранятся в том же едином миварном пространстве и могут представлять собой правила или процедуры обработки. Отметим, что любое отношение также может быть добавлено, удалено или изменено в любое время. Таким образом, именно миварное информационное пространство должно обеспечить реальную адаптивность и эволюционность синтеза информационно-вычислительных конфигураций (МВС и ВВК).

**2. Разработка метода обработки данных на основе применения логической сети правил, управляемой потоком данных.** Так как миварный подход существенно расширяет возможности обработки информации в АСОИ (интеллектуальных системах), то целесообразно разработать новый метод обработки данных. Для этого, в миварном пространстве из всех правил и объектов формируется логическая сеть вывода (обработки), управляемая потоком данных. В зависимости от имеющихся значений переменных и правил, подграфы обработки-вывода могут динамически изменяться. При этом возможно формирование новых правил, существенно сокращающих процесс вывода.

Отметим, что адаптивная логическая сеть, управляемая потоком данных, должна объединить и обработку известных данных на основе известных правил, и одновременно осуществлять обучение, изменение системы правил со своевременным внесением изменений в соответствующие процессы, что, собственно и должно обеспечить эволюционное развитие всей компьютерной системы в целом. Кроме того, важное требование заключается в том, что такая логическая сеть должна позволит максимально распараллелить обработку любого потока данных.

**3. Разработка метода быстрого поиска маршрута логического вывода на основе построения многополюсной сети и поиска ее минимального разреза.** Для повышения оперативности обработки данных в САС ИВК предполагается

разработать новый метод быстрого поиска маршрута вывода. Суть этого нового метода в преобразовании логической сети правил в многополюсную сеть теории графов и поиска ее минимального разреза. Для этого строится граф обработки, с одной стороны от исходных, известных значений, а с другой стороны - от требуемых, искомым. Затем осуществляется анализ связности этого графа путем поиска его минимального разреза. Если минимальный разрез больше или равен единицы, то решение существует и определяется кратчайший, минимальный путь достижения требуемых результатов. На основе полученного пути непосредственно запускается механизм вывода, что позволит значительно экономить вычислительные ресурсы. Если минимальный разрез равен нулю, то цепочки вывода нет, но можно определить значения каких именно переменных являются критическими для уточняющего запроса пользователю, что и является признаком "активности" обработки данных. Также предполагается разработать квадратичный метод поиска минимального разреза многополюсных сетей. Внедрение этих методов позволит активно формулировать уточняющие запросы и снизить вычислительную сложность поиска маршрута логического вывода с NP-полной до квадратичной (а при возможности - достичь линейной сложности), что, таким образом, значительно повысит оперативность функционирования АССОИ и решения сложных и специальных (уникальных) задач.

**4. Разработка метода максимального распараллеливания множественного доступа к общей базе данных** путем построения виртуальных потоковых баз данных. Оценка возможностей известных методов корректного распараллеливания множественного доступа к общей базе данных (БД) показала их низкую эффективность, поэтому предлагается разработать новый метод. Пусть элементы входного потока поступают на обработку в различные моменты времени, тогда, порождаемые ими процессы обработки, могут быть разделены на старших (поступивших на обработку ранее) и младших. Суть предлагаемого метода в том, что самый старший процесс работает непосредственно с реальной БД, а для всех остальных действующих процессов создается виртуальная потоковая база данных (ВПБД), в которой для каждого процесса создается и ведется соответствующая персональная база. Все изменения, которые процесс должен внести в БД, фиксируются в его персональной базе. Младшие процессы получают доступ по чтению последовательно через персональные базы старших процессов, а все изменения вносят только в "свои" персональные базы.

Виртуальная потоковая база данных фактически является надстройкой над реальной БД. Получаем, что все изменения вносятся в общую БД, но младшие процессы не оказывают никакого влияния на одновременно обрабатываемые с ними старшие процессы. Таким образом, из всех известных способов только предложенная параллельная обработка позволит обеспечить корректное максимальное распараллеливание множественного доступа к общей базе данных.

**5. Разработка линейного метода единично-инкрементного суммирования чисел для решения некоторых классов специальных задач.** Анализ основных факторов, влияющих на процесс обработки информации, выявил возможность повышения быстродействия ВВК при использовании запатентованного способа быстрого суммирования чисел, который позволит разработать новый линейный метод переборного суммирования. Суть этого метода состоит в алгоритмической минимизации количества операций сложения при переборном (единично-инкрементном) суммировании чисел. Реализация этого метода суммирования позволит перейти от степенной к линейной зависимости общего количества операций сложения от разрядности суммируемых чисел.

Для решения этих задач в монографии применяются методы теорий множеств, графов, структур данных, принятия решений, математической логики, информатики, системного анализа, реляционной алгебры математического программирования и методы оптимизации на сетях и графах.

Отметим, что миварный подход должен позволить адаптивно и своевременно наращивать объем хранимой информации и обрабатывать ее с применением активной логической сети правил, управляемой потоком данных. В свою очередь, это должно позволить реализовать предпосылки создания "интеллектуальных" (самообучаемых) компьютерных систем с изменяемой, адаптивной конфигурацией на основе эволюционных неоднородных многопроцессорных вычислительных систем.

Отметим, что эти и другие, представленные в данной работе, научные результаты позволяют принципиально (в основном) решить **проблему создания эволюционных, адаптивных баз данных и знаний для синтеза интеллектуальных систем**. В свою очередь, эволюционные, адаптивные базы данных и правил (знаний) позволят решить важную, актуальную научную **проблему создания систем адаптивного синтеза информационно-вычислительных конфигураций**.

Прежде чем перейти к анализу основных структур представления данных традиционных моделей данных, представляется целесообразным обсудить гносеологические аспекты представления данных и знаний в интеллектуальных системах. Анализ этих философских проблем играет важную роль для последующего понимания основных положений миварной концепции представления данных.

Однако, при первоначальном или кратком ознакомлении с данной работой рекомендуется второй раздел пропустить, так как в нем практически нет естественнонаучных и технических исследований, и сразу перейти к третьему разделу монографии, в котором анализируются традиционные модели данных.

## 2. ГНОСЕОЛОГИЧЕСКИЕ АСПЕКТЫ ПРЕДСТАВЛЕНИЯ ДАННЫХ И ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

### 2.1. ФИЛОСОФСКИЕ КАТЕГОРИИ: ВЕЩЬ, СВОЙСТВО, ОТНОШЕНИЕ

#### 2.1.1. ФОРМИРОВАНИЕ ИСХОДНЫХ ФИЛОСОФСКИХ ПОНЯТИЙ

Для всестороннего понимания и решения проблемы создания эволюционных баз данных и знаний необходимо, прежде всего, исследовать различные аспекты мышления, представления и обработки информации человеком. В самом широком, обобщенном виде и в концептуальном плане эти вопросы исследуются в философии. Рассмотрим обзор гносеологических аспектов представления данных и знаний в интеллектуальных системах, а также проанализируем роль философских категорий: **вещь, свойство и отношение** на основе литературы [Л. 14, 30, 49, 53, 68, 69, 81, 119, 134, 169, 172, 181, 182, 186, 218, 222, 230, 234, 240, 246, 251 - 254, 267, 277, 290, 297, 302, 303, 329].

Прежде всего, отметим, что история науки есть история ее идей, принципов и понятий. Философия начинала формироваться в недрах мифосознания [Л. 246, с. 7], что наложило определенный отпечаток на сам процесс образования некоторых ее понятий и идей. С возникновением поэзии мифологическое **отождествление образа и вещи** превращается в художественное сравнение и аналогию, становится **орудием познания** и средством освоения мира. На основе художественного сравнения и метафоры развивалась научная аналогия и понятийное мышление. В ходе дальнейшего развития образно-аналитический стиль мышления постепенно заменяется на понятийно-аналитический, а вера в приобщенность человека к некоему природному целому трансформируется в идею мирового единства. С сознательной постановки таких вопросов и начинает зарождаться философия, как некоторая система взглядов на мир в целом. Отметим, что особенности стиля мышления древних людей нашли свое выражение в известной антропоморфности некоторых исходных для философии понятий.

С утверждения о единстве мира, с поисков оснований этого единства начинается философия античности. Идея единства в том или ином виде проходит красной нитью через всю древнегреческую философию [Л. 14, 246]. Для периода становления древнегреческой философии характерно обилие различных философских взглядов и идей. Античная философия оказалась настолько идейно богатой, что в своих многообразных формах заложила основу почти всех позднейших типов мировоззрений. Поиски первоначал исходят, по сути дела, из предположения, что всякое сущее может быть расщеплено на то, что есть в нем неизменного, составляющего его основу и суть, и то, что составляет преходящее, изменяющееся в сущем - явление. **Мир есть бесконечное многообразие всего сущего**, но в этой своей дискретности он **един в целом и в каждом своем фрагменте**, части - таков лейтмотив этих поисков, нашедших позднее свое более конкретное выражение в принципе атомизма.

Другой фундаментальной идеей, сыгравшей решающую роль в выработке диалектической концепции вещи, была идея противоречивой природы всего сущего. **Идею противоречивой природы мирового процесса**, согласно которой причиной совершающегося в определенном временном порядке возникновения и гибели вещей, является борьба противоположностей, высказывал Анаксимандр (6 в. до н.э.).

Но наиболее "ярко" и обще идея противоречивой природы мира была выражена греческим философом Гераклитом Эфесским (6-5 в. до н.э.): "в ту же реку вступаем и не вступаем, существуем и не существуем". Гармония мира всякий раз нарушается борьбой противоположных начал, присущих самому существу. По образному выражению Ипполита, комментирующего эту мысль Гераклита, "борьба - отец всего и всему царь, она источник того, что все в мире движется, изменяется и, стало быть, сама гармония также оказывается в известном смысле относительной, а с ней относительной оказывается и определенность самих вещей. **Гармония и борьба - две стороны бытия единого сущего.** Гераклитовская концепция мира как процесса превращений космического огня, управляемого логосом, была, по сути своей, первой попыткой представить мир как саморегулирующуюся систему.

Становится понятным, что **объективный предмет мысли и мысль о предмете не совпадают.** Впрочем, существовала точка зрения, согласно которой, все как оно представляется людям, так и существует. Согласно Демокриту пустота так же реальна, как и сами атомы. Атомы и пустота - это два противоположных начала, при наличии которых только и возможно образование тел. Атомы могут быть не только различной формы, но и величины: от самых мельчайших, до равных всему миру. Райбекас А.Я. подчеркивает, что это созвучно современному представлению ученых, например - Маркову А.А., согласно которому элементарная частица может оказаться целым миром, а мир - элементарной частицей [Л. 246, с. 14]. Отметим, что греки не делали различия между качеством и свойством. Качество или свойство, есть то, что чувственно воспринимается и, исходя уже из этого, рассматривается как нечто присущее, свойственное телу. Качества существуют лишь в отношении к другому (ощущающему их человеку) и зависят от этого другого. Тем не менее, среди свойств можно выделить те, которые присущи всем телам - форма, цвет, величина и т.д. Это постоянные свойства, которым, по Эпикуру, тело обязано своим постоянным существованием.

### 2.1.2. ВЕЩЬ, СВОЙСТВО, ОТНОШЕНИЕ

Три основных философских понятия, категории: **вещь, свойство, отношение** - позволяют отобразить мир как бесконечное многообразие материальных реальностей в их возникновении, пребывании, изменении и гибели. Эти понятия формируются в истории познания именно как **система понятий при центральном, лидирующем положении категории "вещь"**. "Только вещи обладают самостоятельным бытием, ... вне и помимо свойств и отношений, которые образуют и выделяют вещь как определенное существование (сущее), вещи нет, как нет и самих свойств и отношений вне и помимо вещей" [Л. 246, с. 130]. Еще Демокрит подчеркивал, что именно **движение делит мир на совокупность вещей.** Все, что существует в своей различённости от другого сущего, есть вещь. **Вещь есть определенная реальность,** или нечто. В понятии вещи заключено противоречие отражаемого им бытия: нечто - это то, что уже есть (существует), но в такой же мере его еще нет, ибо оно конкретно не определено.

Определенность других нечто также существует для него только благодаря его соотношению с ними и, стало быть, все существует лишь в отношении к чему-нибудь. В отношениях с другими, нечто ограничивает своей определенностью иное и, таким образом, само становится ограниченным; и то, что нечто есть как единица, исключаяющая своим бытием другое, оно есть благодаря этой своей границе. Таким образом, любая (всякая) определенная материальная реальность, су-

ществующая как нераздельность (цельность) многообразного - есть вещь. Тогда, **вещью** будет любой фрагмент материального мира.

### 2.1.3. ВЗАИМОДЕЙСТВИЕ, СВЯЗЬ, ОТНОШЕНИЕ

Понятия связи, взаимодействия и отношения - суть понятия, отображающие один и тот же объект - **процесс материального движения**. В том обстоятельстве, что материальные вещи "находятся во взаимной связи, - подчеркивал Ф. Энгельс, - уже заключено то, что они воздействуют друг на друга, и это их взаимное воздействие друг на друга и есть именно движение" [Л. 246, с. 132]. Более того, **каждая вещь связана с каждой** и отношения каждой вещи не только различны, но и всеобщы, и универсальны. Отметим, что "отношение" - наименее разработанная категория в системе понятий "вещь - свойство - отношение". Между тем, классики философии всегда подчеркивали, что **материальная вещь вне отношений и до отношений не существует**. За понятием "отношение" совершенно четко устанавливается статус категории, отображающей не только отношения между понятиями (логические отношения), но и материальное движение.

Понятия связи, взаимодействия и отношения образуют группу, которую условно можно было бы назвать группой категорий **движения**. Будучи понятиями разного уровня абстракции, эти категории отображают материальное движение с разных сторон. Действительность вещи есть диалектическое единство реальных свойств и отношений, в которых она суть и обнаруживает свое существование. Рассмотрим анализ категории отношение - наиболее абстрактной в ряду категорий, отображающих движение.

Понятие отношение появилось для обозначения логической операции сравнения предметов познания. В этом смысле оно обозначало собой отнесенность одного к другому и выражало активность гносеологической позиции субъекта познания. Всеобщий характер этого понятия определяется тем, что оно служит для отображения воздействующих друг на друга вещей. Понятием отношения охватывается все многообразие известных форм движения. Отношение означает, что в нем есть две стороны, которые относятся друг к другу, поэтому отношение одной вещи к другой есть отношение обеих вещей. В отношении участвуют, по крайней мере, две различные вещи. **Отношение устанавливается между различными вещами по тождественному для них признаку**, причем соотносимые стороны различны и тождественны в одном и том же отношении.

Различенность, отграниченность сторон отношения и их тождественность (одно и то же отношение) есть неотъемлемые моменты отношения вещей и как их противоречивое единство отношение - есть противоречие. В свое время, Гегель писал: "все, что существует, находится в отношении, и это отношение есть истина всякого существования. Благодаря отношению существующее - не абстрактно, не стоит само по себе, а есть лишь в другом, но в этом другом оно есть отношение с собою и отношение есть единство соотношения с собою и соотношения с другими" [Л. 68, 69]. В этом соотношении с собой через другое каждая из сторон отношения противоположна самой себе и, следовательно, всякое **отношение, действительно, суть противоречие**.

**Связь**. Историко-философская традиция состоит в понимании связи вещей **как отношения**. В понятии связи, прежде всего, отображался факт изменения определенности одной вещи в результате воздействия на нее другой. Выяснить зависимость можно лишь соотнеся действие и его результат. "Связать", установить

связь вещей, значит, соединить их, отобразить как различное единство. Связь вещей, с одной стороны, исходит из их единства, с другой стороны - устанавливает его. Следовательно, познавая многообразие связей вещей, мы необходимо приходим к идее глобального единства материального мира. В понятии **связи отображается взаимное изменение вещей**, обусловленное процессом их взаимодействия. В понятии связи находит свое выражение динамический (процессуальный) характер отношений материальных вещей, реализующихся в форме их взаимодействия. Таким образом, **связь** так же **суть** (представляет собой, есть) **отношение**, и в этом смысле - **противоречие**. Взаимозависимость сторон, полюсов взаимодействия есть "внешнее", форма, в которой взаимодействие обнаруживает себя как противоречие.

**Взаимодействие.** Понятие взаимодействия появилось значительно позднее. Ф. Энгельс писал: "взаимодействие есть такой двусторонний процесс, который по своей природе может рассматриваться с двух различных точек зрения. Чтобы его понять как целое, его даже необходимо исследовать в отдельности, сначала с одной, затем с другой точки зрения, прежде чем можно будет подытожить совокупный результат" [Л. 182, 246]. Взаимодействие мысленно расчленяется, разносится на воздействие одного на другое. Идея взаимного воздействия находит свое первоначальное выражение в представлении о взаимной зависимости, взаимной связи материальных вещей. Только с выявлением противоречивой природы движения, с пониманием отношения как противоречия стала возможной конкретизация представления о движении и связи вещей в более глубокое и содержательное понятие взаимодействия. Если, рассматривать материальное отношение вещей как отношение противоположностей, то в нем можно выделить, по крайней мере, два уровня.

1. уровень соотношения с другими - это уровень внешний, уровень существования противоположных сторон отношения в их нераздельности друг от друга.
2. уровень отношения противоположностей - отношение каждой из соотносящихся сторон к самой себе.

Нераздельность бытия взаимодействующих сторон - полюсов материального отношения находит свое выражение во **взаимной зависимости** каждой из них от другой, что и отображается в содержании понятия "связь". Понятие связи, отражая внешнюю сторону взаимодействия вещей, фиксирует скорее уже результат взаимодействия, а именно взаимозависимость соотносимых сторон взаимодействия. Связь не только соединяет, но и объединяет разные стороны отношения. Таким образом, **связь можно рассматривать как внешнее завершенное отношение вещей**.

Материальные отношения вещей всегда носят характер взаимодействия и обнаруживают себя в форме той или иной зависимости соотносящихся друг от друга вещей, т.е. их связи. Дело в том, что если понятие "взаимодействие" есть понятие, обозначающее и отображающее материальное движение на уровне действительного процесса, то понятие "связь" и "отношение" суть понятия, которые по разному, каждое со своей стороны служит для раскрытия содержания этого реального процесса. Предметом их отражения является, таким образом, одно и то же - материальное движение. Вот почему каждое из них (понятие связи, отношения) не обозначает нечто существующее наряду с другим в онтологическом смысле. Они - понятия разного уровня абстракции, с разных сторон и глубиной отображающие один и тот же объективный факт, т.е. объект.

Материальное движение, описанное:

- **на уровне явления**, отображается с помощью понятия взаимозависимости, **связи**;

- как понятие **на уровне сущности** - с помощью понятия **отношения**;
- познанное **на уровне действительного процесса** - с помощью понятия **взаимодействия**.

В этом ряду понятий, отображающих материальное движение, **категория отношения является предельной абстракцией, выражающей самую суть движения - его противоречивую природу**. Именно поэтому и взаимодействие, и связь, фиксирующая внешний аспект взаимодействия вещей, есть отношение. Подчеркивая процессуальный характер материальных отношений вещей, понятие связи дает возможность отобразить не только непосредственное, но и опосредованное другими вещами взаимодействие противоположностей. Таким образом, отображая сущность материального движения и форму, в которой оно обнаруживает себя (изменение вещей), категории отношения и связи охватывают, таким образом, все бесконечное многообразие действительного движения (взаимодействия) вещей и их мыслительных образов (понятий).

#### 2.1.4. КАЧЕСТВО И СВОЙСТВО

Устойчивая определенность вещи, характеризующая индивидуальность ее выделенного бытия и сохраняющаяся в многообразии отношений с другими вещами, есть ее **качество**. **Качество есть такой отличительный признак вещи, который характеризует ее в целом**. Качество есть такая определенность вещи, которая в известной мере дана человеку уже в непосредственности его чувственных восприятий, что и дает основание определить качество как непосредственную определенность вещи. Качество выражает специфику и устойчивость вещи. Будучи определенностью вещи, качество определяет границу ее бытия в качестве данной вещи, и вещь есть то, что она есть только в этой своей границе. Таким образом, вещь в своем качестве не только определена, но и ограничена. Качество есть целостная характеристика ее (вещи) бытия. В этом смысле каждая вещь обладает одним качеством. Однако, это не исключает признания того факта, что объект познания может быть охарактеризован целым набором качеств. Тот факт, что всякая вещь в известном смысле многокачественна, несомненно, обнаруживает противоречивую природу бытия вещи и процесса ее познания.

В своей качественной определенности вещь противостоит иному, исключает своей определенностью определенность других вещей. Поэтому можно сказать, что **способ, каким качество вещи существует, есть отрицание** [Л. 246, с. 143]. Именно отрицая все другое, соотносясь с другим, качество полагает, утверждает себя как определенность. Вещь сохраняет себя как это данное качество через многообразие своих внешних отношений с другими вещами, обнаруживая при этом эту свою качественную определенность как единство своих внешних определений. Именно благодаря своим внешним определениям, которые мы назовем свойствами, вещь присущим ей образом сохраняет себя в соотношении с иным, показывает в ином силу своих собственных определений.

Вещь обнаруживает свое качество через многообразие свойств. Вещь не растворяется в этом многообразии свойств, а во всех отношениях с другими вещами она остается неким единством, исключаяющим своей определенностью все иное. В этих отношениях с другими вещами, данная вещь изменяется, и эти изменения вещи следует отнести, прежде всего, к изменениям ее свойств.

Свойства вещи характеризуют, таким образом, ту сторону ее бытия, в которой она постоянно становится иной. За многоликостью вещи, за разнообразием ее

свойств находится нечто, что составляет устойчивое наличие вещи, одно и то же множество ее определений и что можно бы определить как основание или основу вещи. Внутреннее единство вещи, ее субстанциональная основа обнаруживает себя как единство свойств, как качество вещи. А качество вещи в том и состоит, чтобы посредством внешних соотношений, через другое представить субстанциональную основу (сущность) вещи. **Качество** есть, следовательно, **внешнее выражение внутреннего единства вещи** и в этом смысле может быть определено как внутренняя определенность вещи, тождественная с ее бытием.

### 2.1.5. СУБСТРАТ И СУБСТАНЦИЯ

В философии существуют, по крайней мере, два исторически сложившихся термина для определения, обозначения основания вещи - понятия "субстрат" и "субстанция" (**сущность**), выражающие собой не только разные аспекты, но и различную глубину понимания природы объективного бытия вещи. Если в понятии "субстрат" основание вещи мыслится как нечто опосредствуемое только внешними отношениями с другими вещами, то в понятии "субстанция" основание вещи осмысливается уже как опосредованное самим собой, как то, что самовоспроизводит себя и является в этом смысле собственной причиной. Гносеологическим коррелятом субстанциональной основы вещи выступает понятие сущности [Л. 246, с. 144].

Качество реально существует как единство конкретных свойств конкретной вещи, т.е. как специфическое качество. Качество же вещи, взятой самой по себе, абстрагированной от ее отношений с другими вещами, есть абстрактное качество и оно тождественно ее сущности. В самом деле, вещь абсолютна и относительна в своем бытии, она есть нечто абсолютное как вещь в себе, как абстрактное качество, т.е. сущность, и относительна, поскольку всегда существует в виде единства конкретных свойств, т.е. как специфическое качество. Итак, с одной стороны, свойство вещи есть то, что становится иным, что характеризует изменение вещи (ее определенность), с другой стороны - свойство принадлежит тому, что вещь есть в себе и что обнаруживается как единство свойств в качестве вещи. Категория свойства есть, таким образом, действительно всеобщее понятие, позволяющее отобразить бытие вещей в их пребывании и изменении.

### 2.1.6. СВОЙСТВО И ОТНОШЕНИЕ

Отношение - это противоречивое единство своих сторон, свойств-полюсов, каждое из которых принадлежит отношению. Каждое из свойств, реализуемых в отношении вещей, принадлежит тому, что каждая из соотносящихся вещей есть в себе, что образует суть этой вещи. Таким образом, **свойство принадлежит и вещи, и отношению одновременно** [Л. 246, с. 146]. В этом их обнаруживаемом единстве и заключена **возможность их взаимопереходов** друг в друга. Отделить вещи, свойства, отношения друг от друга можно лишь мысленно, в абстракции. Получаем **диалектический переход "вещь - свойство - отношение"**. Свойство означает способность вещи определенным образом относиться к другим вещам, откуда видно, что свойство есть сторона, момент отношения. Но сторона, момент никогда не бывает равной или тождественной целому. Свойства вещей не существуют вне и помимо их отношений. Более того, свойство есть в известном смысле взаимоотношение, поскольку оно есть определенность вещи не потому, что свойственно ей, а потому, что только в отношениях с другими вещами, т.е. благодаря

своим свойствам вещь сохраняет себя присущим ей образом. И вне этого взаимодействия вещь не существует.

В силу самой структуры материального отношения как отношения противоположностей в каждом отдельном, конкретном отношении данные соотносящиеся вещи обнаруживают себя только с какой-то одной стороны, свойства, определяемого именно этим конкретным отношением. Возможно, именно этим объясняются безуспешные попытки физиков обнаружить в одном и том же эксперименте существование волновых и корпускулярных свойств материальных объектов. Отметим эту важную особенность отношения вещей для того, чтобы подчеркнуть фундаментальность факта существования вещи как единства многообразного.

### **2.1.7. КОЛИЧЕСТВЕННАЯ ОПРЕДЕЛЕННОСТЬ ВЕЩИ**

Итак, качество существует лишь опосредованно, как единство свойств вещи. Опосредование бытия вещи составляет ее количественную определенность (количество). Количественная различенность в бытии вещи есть различие "внутри" (в рамках) данного качества или различие вещей одного и того же качества. Количество есть отношение различных интенсивностей одного и того же качества (свойства) и выражается числом. В своем бытии вещь выступает как "одно" во "многом" или многое в одном, как конкретное тождество. Как нечто единое, - "одно", - вещь непрерывна. Непрерывность есть равенство себе, но равенство себе "многого". Непрерывность, стало быть, содержит в себе момент дискретности, последняя же будучи различным моментом непрерывности, оказывается по словам Гегеля, "сливающейся дискретностью" [Л. 68, 69, 119, 246], и множественность положена в непрерывность бытия вещи как ее момент. Количественная определенность вещи есть единство этих моментов непрерывности и дискретности, а "единица" и "численность" соответственно моменты числа, выражающего ее. Как дискретность вещь есть определенное множество (множество свойств) и имеет численность внутри себя, а как непрерывность (качество) - она есть единица, численно определяемая через соотношение с другими вещами, опять же в форме свойств. Изменение количественной определенности вещи есть прежде всего изменение ее свойств. Количественная определенность вещи есть определенность ее качества, которую она (вещь) получает в своих отношениях с другими вещами. Качество и количество - нераздельные моменты, стороны бытия вещи, и действительность вещи как единства многообразного есть единство ее количественной и качественной определенностей.

## **2.2. ВЕЩЬ КАК СИСТЕМА**

### **2.2.1. ОСНОВНОЕ ПРОТИВОРЕЧИЕ И СИСТЕМНЫЙ ХАРАКТЕР БЫТИЯ ВЕЩИ**

Действительность вещи имеется лишь в отношениях ее к другим вещам, и как количественно определенное качество вещь есть одновременно и единое (качество), и многое (множество свойств). Вещь в своем бытии, существуя в рамках меры как конкретное тождество (и противоречие), может быть представлена как отношение, т.е. вещь есть отношение.

**Вещь есть единство многообразного.** В этом заключается основное противоречие ее бытия. Формой разрешения этого противоречия, в ходе которого оно постоянно воспроизводится, является двойной переход: качества в количество

(как множество свойств, посредством которых оно обнаруживает себя) и количества в качество (как единство свойств). В результате вещь существует в своем бытии как целокупность, цельность, имея в своем основании некоторое внутреннее единство, выявляемое в процессе познания как ее сущность. Противоречивость бытия вещи находит выражение в ее системном характере. Именно **как единство многообразного вещь есть система**. Это созвучно выражению Аристотеля: общее также есть нечто целое, так как охватывает многое наподобие частей [Л. 14, 246]. Представление о природе как системе взаимодействующих друг с другом материальных вещей утверждало динамическую картину природы и отображало ее как процесс. Системный подход оказывается чрезвычайно плодотворным, позволяя выявить динамическую картину генезиса и функционирования вещи в действительности ее материального бытия, и таким образом - объяснить ее.

### 2.2.2. ВЕЩЬ КАК СИСТЕМА И ЭЛЕМЕНТ

Рассмотрим ряд определений понятия "система". Система - "есть комплекс взаимодействующих элементов (Берталанфи) [Л. 30]; ... - это множество объектов вместе с отношениями между объектами и их атрибутами (А. Холл и Р. Фейджин); ... - совокупность взаимосвязанных элементов, выступающих как единое целое (И.В. Блауберг, В.Н. Садовский, Э.Г. Юдин) [Л. 246]; ... - комплекс некоторых объектов, находящихся в определенных отношениях друг с другом (Н.Ф. Овчинников) [Л. 218]; ... - это множество объектов, на которых реализуется отношение с заранее заданным свойством (А.И. Уемов) [Л. 290] и т.д. Отметим, что важнейшей характеристикой системного бытия вещей является их целостность. Термин "Systema" (лат.) буквально означает целое, состоящее из частей [Л. 246, с. 152]. **Быть целым значит быть единым**, и в этом своем основном значении понятие целого отражает общность частей целого ("то же самое" или "одно и то же") различных вещей, единое, которое и **делает их целым**, а, следовательно, **системой**. С точки зрения определения целого (практически) все формы организации материального мира суть системы. И совокупность, как определенным образом организованная система, в этом смысле не составляет исключения. Подчеркнем, что сказанное в полной мере относится и к понятию "множества" вещей. Говоря о вещах, как системах, необходимо иметь в виду, что уровень организации их целостности (иначе, уровень их системной организации) может быть различным. Применительно к познавательным, гносеологическим системам это **означает различный уровень организации (синтеза) знания о вещи**.

Понятия **совокупности** и **множества** выражают, по-видимому, не столько сам факт реального существования таких систем со столь низким уровнем организации целого, сколько степень приближения человека к действительному познанию системного бытия мира: "совокупность", "множество", "система" как понятия (ряд понятий) различного уровня абстракции, отображающей бытие любой материальной вещи как целого.

О понятии "**комплекс**" можно сказать, что оно может быть поставлено в самое начало этого ряда как наиболее неопределенное. Эта неопределенность делает использование термина "комплекс" в определении "системы" наиболее предпочтительным. Исходя из целостности и динамического характера, можно сказать, что как система вещь есть комплекс элементов, взаимодействием которых образовано ее цельное единство [Л. 246, с. 154]. Вместе с тем, определенные преимущества представляет определение системы, данное Н.Ф. Овчинниковым:

**система есть комплекс некоторых объектов, находящихся в определенных отношениях друг с другом** [Л. 218]. Использование терминов "некоторые" и "определенные" позволяет подчеркнуть, что не любой произвольный комплекс вещей и отношений между ними образует систему. Кроме того, что очень важно, охватывая системы как объективно реального, так и познавательного плана, это определение выражает **активный характер гносеологической позиции исследователя** в системном отображении материального мира.

Будучи компонентом целого, образующие систему вещи тем самым становятся ее **элементами**. Становясь элементом разных систем, одна и та же по своей сути вещь приобретает свои, специфические для каждой из этих систем особенности поведения (функционирования), поскольку всякий элемент функционально служит "интересам" именно своей системы. **Элемент** есть простое, в отличие от системы, которая выступает по отношению к своим элементам как нечто сложное. Элемент есть нечто неделимое, целостное по своей функции, по той роли, которую он играет в образовании и функционировании системы, т.е. целого.

**Функция** есть такое отношение элемента к целому, которое делает действие элемента с точки зрения "интересов" данной системы целесообразным и, таким образом, служит обеспечению сохранения целого. Реализуется функция через взаимодействие элементов и системы в целом с другими вещами.

В некотором смысле, можно определить элемент как такой компонент системы, качественная определенность и функционирование которого несут на себе специфику его существования в рамках целого. Отметим, что представление о вещи как системе есть не что иное, как конкретизация и, в этом смысле, развитие процессуального, динамического аспекта классического понимания вещи как целого [Л. 246, с. 156]. Подчеркнем, что понятие "**часть**" не совпадает с понятием "**элемент**". И если любой элемент системы может рассматриваться как ее часть, то далеко не всякая часть системы (целого) является ее элементом. Элемент есть такой компонент системы, который способен к относительно самостоятельному осуществлению определенной функции. В этом заключается онтологическое основание выделения того или иного компонента системы в качестве ее элемента.

### 2.2.3. ОРГАНИЗАЦИЯ СИСТЕМНОГО БЫТИЯ ВЕЩИ. ПОНЯТИЕ СТРУКТУРЫ

Организация вещи как системы находит свое отражение в понятии структуры. **Структура** - это то, что организует целое, т.е. систему. Термин "структура" буквально означает **строение** и обычно интерпретируется как **порядок расположения частей в целом** [Л. 246, с. 157]. Выявление состава целого, т.е. его частей, составляет первоначальную и наиболее простую операцию в познании вещи, а при последующем исследовании строения вещи основной акцент делается уже на исследование отношений частей.

Структура понимается как совокупность отношений, абстрагированных от самих соотносимых частей целого. Иными словами, совокупность отношений самих по себе. Такое отношение можно назвать **формальным**, а выделенные таким путем структуры - **формальными структурами**.

Системный характер организации вещи как целого есть объективный факт. Подчеркнем, что невозможно характеризовать отношение, не характеризуя при этом сторон отношения. Философы считают, что представление о структуре как типе, сети связей, отношений в их абстракции от элементов системы не дает дей-

ствительной картины системной организации вещи, сводя ее лишь к одним отношениям. Структура вещи объективно существует как единство элементов и отношений между ними. Элементы системы выступают, таким образом, в роли ее структурных элементов, а отношения между ними - как структурные отношения. Особенность функционирования структуры вещи в том, что, организуя вещь как данное конкретное целое (систему), структура обеспечивает ее сохранение. Поэтому структура есть то, что остается относительно постоянным, сохраняющимся в изменчивом бытии вещи, она суть ее инвариантный аспект.

Рассмотрим эволюцию взглядов на организованный характер бытия вещей. Инвариантный аспект системы:

- на уровне элементов - это ее состав,
- на уровне отношений между элементами - структурные отношения (формальная структура),
- на уровне элементов и их отношений - структура системы.

Овчинников Н.Ф. вводит еще дополнительно инвариантный аспект системы на уровне целого, как саму систему [Л. 218]. Однако, с этим не согласен Райбекас А.Я., который считает такой аспект сомнительным, так как система инвариантна по определению, а рассмотрение этого дополнительного уровня не несет никакого нового знания и, таким образом, получается удвоение понятия: система оказывается тождественной структуре [Л. 246, с. 161]. Структурные отношения есть отношения, в ходе которых происходит постоянное разрешение и восстановление основного противоречия вещи: целостность системы и ее элементы можно рассматривать как противоположные моменты бытия вещи. Существующее противоречие находит свое разрешение в третьем, а именно, в едином всех образующих систему элементов. Будучи разрешенным, основное противоречие вещи образует ее основание, и пока основное противоречие и способ его разрешения воспроизводятся - вещь существует как данное целое, непрерывно воспроизводящее основание своего системного бытия.

Если **структура** есть в известном смысле **статистический** аспект системной организации бытия вещи, то понятие **организация**, не включая в себя никаких иных признаков, кроме признаков, свойственных понятию структуры, еще более подчеркивает существенную **динамичность** элементов и связей между ними. Можно сказать, что понятие **организации является предельно широким понятием**, характеризующим системный характер бытия всякой вещи, и, стало быть, всего мира материальных вещей. В соответствии с концепцией системной организации вещи как целого, в последней можно выделить, по крайней мере, два уровня ее организации:

- 1) внутренний, или генетический, который условно можно определить как генетическую структуру, и
- 2) внешний, функциональный - структура функционирования.

Организованное бытие вещи оказывается единством ее генетической и функциональной структур. В силу противоречивой природы своего бытия материальная вещь оказывается системно организованной, объективно - она есть система. Впрочем, можно допустить существование известной вольности (произвола) в выделении тех или иных аспектов системной организации объекта, что обусловлено, связано с активной гносеологической позицией субъекта познания.

## 2.2.4. СИСТЕМНОСТЬ ВЕЩИ И ПРОБЛЕМА ЕЕ КАЧЕСТВ

С точки зрения своей системной организации вещь и однокачественна и, в известном смысле, многокачественна. Зачастую многие в этом вопросе смешивают онтологический аспект с гносеологическим. Рассмотрим это более подробно. Под качеством понимается внешнее единство вещи, характеристика ее целостности. Качество можно представить как "множество свойств, органически связанных между собой общим законом взаимодействия", т.е. качество - это система свойств [Л. 246, с. 164]. В то же время, качество можно рассматривать как выражение устойчивой определенности структуры системы. Как отдельное конкретное существование вещь характеризуется одним, присущим ей только в данных конкретных условиях бытия специфическим качеством. Если же мы рассматриваем действительность вещи, т.е. все доступные познанию многообразные формы явлений ее сущности, то в этом случае вещь окажется одновременно однокачественной (как абстрактное качество - сущность) и многокачественной (как многообразие специфических качеств).

Если состояние понимать как систему свойств вещи в данный момент времени ее бытия, то специфическое качество вещи можно определить как то, что выражает конкретное состояние сущности вещи. Те изменения состояний сущности, которые не меняют ее самое, позволяют квалифицировать их, как изменения количественные, и тогда, каждое специфическое качество вещи, характеризующее определенное состояние сущности вещи в конкретных условиях ее осуществления, может рассматриваться как существенное свойство.

При одной и той же генетической структуре вещь может обладать неким многообразием структур своего функционирования, наличие которых и дает онтологическое основание рассмотрения ее как многокачественной. Генетическая структура вещи также не определяется с совершенно жесткой однозначностью, что тоже находит свое известное выражение в специфике качества вещи. Выделение подсистем системной организации вещи, которые определяют те или иные стороны ее бытия как целого, позволяет характеризовать вещь целым же рядом ее существенных свойств (специфических качеств). Причем, именно целым, ибо все это многообразие качественной определенности вещи существует именно как единое целое, как главное качество вещи.

## 2.2.5. ОБЪЕКТИВНАЯ СИСТЕМНОСТЬ ВЕЩИ

Таким образом, **материальная вещь объективно системна** и ее системное представление, если оно стремится быть адекватным, возможно настолько, насколько нам удастся в практическом и теоретическом освоении действительности выявить существенные свойства и отношения вещи. Следовательно, на основе знания этих системообразующих параметров вещи, можно воссоздать картину ее системной организации. Полисистемный характер организации вещи обуславливает возможность ее полисистемного отображения в теории, которое, таким образом, отнюдь не является произвольным. В диалектике системного бытия материальной действительности реализуется диалектика вещи, свойства и отношения, их взаимосвязь, взаимопереходы. При этом, взаимопереходы этих категорий друг в друга в принципе реализуются через внутреннее единство той системы, которую они образуют [Л. 246, с. 166]. Именно этим можно объяснить центральное положение категории "вещь" в системе понятий "вещь - свойство - отношение".

## 2.3. ВЕЩЬ В ПРОЦЕССЕ СВОЕГО ВОЗНИКНОВЕНИЯ, ПРЕБЫВАНИЯ И РАЗВИТИЯ

### 2.3.1. ПОНЯТИЕ ОСНОВАНИЯ ВЕЩИ И УСЛОВИЯ ЕЕ СУЩЕСТВОВАНИЯ

Внутренне единство как бы лежит в фундаменте, в основании вещи. Оно есть нечто сохраняющееся в отношениях вещи с другими вещами и, в этом смысле, представляет собой то, из чего должно быть понято бытие вещи, что обосновывает ее существование. При этом, если основание обосновывает существование вещи, то условия обуславливают его, т.е. существование. Следовательно, вещь в своем реальном бытии есть диалектическое **единство основания и условий**: вещь есть нечто в самой себе (в основании) и в другом (в условиях).

**Условия** выступают как предпосылка и собственный момент основания, так как благодаря условиям (через условия) вещь соотносится сама с собой, образуя тождественное своего бытия - основание.

**Основание вещи** может быть представлено как конкретное тождество, как то, что есть единство с собой в своем отличии от себя, т.е. как внутренняя определенность вещи. Именно в этом отношении к себе как к своему иному в диалектическом саморазличении себя основание приобретает свою собственную определенность как **субстанциональная основа**, суть вещи, выражающая ее природу. Через отношения с другими вещами как внешними для нее условиями субстанция вещи (ее сущность) обнаруживает, являет себя. Форма, в которой вещь обнаруживает свое существование, есть **явление**. Явление не существует наряду с сущностью, с вещами, поскольку оно есть явление сущности, являющаяся сущность, т.е. сама вещь. Явление - это такое объективное существование вещи, которое понято познающим мышлением не только как обоснованное, но и как обусловленное существование. Мир реальных вещей объективно оказывается миром материальных явлений.

Рассмотрим более подробно взаимодействие, т.е. отношение. Отношение вещей есть отношение противоположностей и по своей сути - противоречие. Каждая из противоположных сторон, как нечто внешнее по отношению друг к другу, выступает в роли условия их взаимного обнаружения, существования в рамках данного отношения. Материальное отношение в этом смысле есть отношение взаимной обусловленности [Л. 246, с. 168]. Вместе с тем, каждая из противоположных сторон отношения этой своей определенностью исключает бытие другой, есть, так сказать, основание ее несуществования.

Следовательно, в любом отношении материальных вещей каждая из взаимодействующих вещей есть одновременно (в одном и том же отношении) и условие бытия другой, и основание того, почему другая вещь не существует (не существует именно как эта вещь). Иначе говоря, любое материальное отношение (взаимодействие) есть отношение обусловленности и обоснованности. Хотя любое материальное отношение по самой своей природе есть всегда отношение опосредования, тем не менее сами отношения в процессе познания могут квалифицироваться и рассматриваться и как непосредственные. Отношение вещей будем называть непосредственным, если оно (с точки зрения конкретной познавательной ситуации) не опосредуется отношением к другим вещам. В этом смысле непосредственное отношение есть прямое отношение взаимодействующих друг с другом вещей. Различение непосредственного и опосредованного все-таки весьма

условно, относительно, так как, между ними имеет место переход, и то, что в одном отношении является непосредственным, в другом - опосредовано, и наоборот. Таким образом, сама вещь есть одновременно и нечто непосредственное (как основание, сущность) и опосредованное (как существование, явление).

### 2.3.2. ГЕНЕЗИС НОВОЙ ВЕЩИ

Определенный интерес представляет история развития представлений о генезисе, возникновении вещей. Проблема возникновения вещей является столь же древней, как и история самой человеческой мысли. Выделяют три следующих основных этапа [Л. 246, с. 169 - 179]. Первый этап - это эпоха возникновения и расцвета философии древних и, прежде всего, античной философии. На этом этапе философы связывали ответ на вопрос о происхождении конкретных вещей с поиском глобальных причин, порождающих мир таким, каков он есть. В это время формируются понятия вещи, свойства и отношения, появляется представление о причине как генетическом отношении, а также рождается гениальная догадка о том, что целое вещи не сводится к сумме ее частей.

Затем была тысячелетняя эпоха господства религиозной философии и схоластики. Следующий этап связан с началом эпохи нового времени. Это - эпоха господства метафизического метода познания, рассматривающего вещи как уже имеющиеся налицо, данные, абсолютно сохраняющие эту свою данность в отношениях с другими вещами и, в этом смысле, неподвижные. Потом перешли к растворению вещи в ее свойствах. Вещь мыслилась как нечто обладающее свойствами, которые она обнаруживает в отношениях с другими вещами. При этом, свойство понималось нечто абсолютно внешнее и, таким образом, случайное в бытии вещи. Свойства вещей не создаются, а только проявляются в их отношениях с другими вещами - таков категорический тезис этого этапа познания материальной действительности.

Третий этап. Первую брешь в господстве идей второго этапа пробил Гегель, изобразивший процесс производства вещей [Л. 68, 69, 246]. Подчеркнем, что научное решение проблемы возникновения вещей возможно в рамках диалектико-материалистической концепции вещи. Суть этой концепции, - по Энгельсу, - в том, что "вещи не есть", а вещи "становятся вещами" [Л. 182]. Отмечая необходимость и плодотворность такого подхода, К. Маркс подчеркивал, что подлинно философское рассмотрение вещи не только вскрывает ее противоречия как реально существующие, но и объясняет их, постигает их генезис, их необходимость [Л. 182, 246].

В настоящее время, о генезисе свойств вещи в процессе ее возникновения свидетельствует и тот факт, что целое (вещь как система) не сводимо к сумме элементов, которые его образуют. Следует признать, что сущность вещи создается, возникает в материальных отношениях вещей. В то же время, свойства не есть нечто абсолютно внешнее в бытии вещей, они суть моменты субстанции вещи, ее сущности. Генезис сущности (субстанциональной основы) новой вещи не может не быть вместе с тем и генезисом ее свойств. Следовательно, свойства вещи не только проявляются, но и создаются в отношениях вещей. Материальные вещи имеют условия своего порождения и существования в других вещах, точнее в их отношениях друг к другу. Отношения, порождающие данную вещь, и отношения ее функционирования в качестве данной вещи, - конечно, разные отношения, точнее разные аспекты, уровни рассмотрения материального отношения как взаимодействия. В свое время, К. Маркс высказал мысль о существовании двоя-

кого рода отношений противоположностей: противоположности существования и противоположности сущности [Л. 182].

Разный характер этих противоречий (и, соответственно, различие способа их разрешения), очевидно, и определяет различную роль отношений в генезисе и функционировании вещи. Особенность существования материального отношения как противоречия заключается в том, что противоречие неизбежно требует своего разрешения, т.е. перехода в некий "спокойный" результат, в котором противоречие в старой своей форме исчезает. Противоречие не существует до и независимо от своего разрешения, в процессе которого оно неизбежно воспроизводит себя как противоречие. Разрешение противоречия оказывается, таким образом, способом существования его как отношения противоположностей. Эту мысль особо выделял К. Маркс, отмечая, что "противоречие одновременно и осуществляется, и разрешается" [Л. 182]. Уничтожение противоречия есть уничтожение движения. Райбекас А.Я. отмечает, что справедливо и обратное утверждение, т.е. уничтожение движения приводит к уничтожению противоречия, что разумеется возможно лишь в научных абстрактных моделях. Именно это и объясняет, почему метафизическое рассмотрение вещей вне их отношений друг с другом не обнаруживает в них противоречий [Л. 246, с. 172 - 173]. Противоречие, требуя своего разрешения, воспроизводит себя как противоречие, создавая, таким образом, форму своего бытия как движения. В процессе своего разрешения противоречие порождает самое себя и с этой стороны оно есть постоянно воспроизводимое единство возникновения и прехождения.

Характерной особенностью существования противоречия является то, что в ходе его разрешения старое единство противоположностей рушится и возникает новое. Этот процесс неизбежно связан с соответствующим изменением сторон (полюсов) отношения. Глубина изменений, которые испытывают при этом стороны отношения, зависит от остроты разрешаемого противоречия, что и определяет ту роль, которую играют те или иные материальные отношения вещей в организации новой вещи.

Рассмотрим случай, когда вещи в своем взаимодействии друг с другом порождают новую вещь. Можно выделить два типа генезиса вещей. Первый тип генезиса вещи выделяют тогда, когда процесс порождения сущности новой вещи весьма скоротечен и потому превращение вещей, участвующих в процессе образования новой вещи в элементы новой системы, происходит, в сущности, одновременно (например, превращение элементарных частиц). Второй тип генезиса вещи носит более постепенный характер. Это путь развития сущности данной вещи (развертывание ее противоречия), в ходе которого противоположные моменты сущности разделяются вплоть до превращения их в крайности, выражающие в своей самостоятельности уже, по сути дела, разные сущности. Противоречие достигает своей наибольшей остроты и требует самого радикального своего разрешения. Поскольку "действительного дуализма сущности не бывает [Л. 182], одна из крайних противоположностей, а именно та, которая выражает необходимость развития, берет верх над другой и возникает новая вещь.

Условия генезиса сущности вещи есть одновременно и условия обнаружения, проявления возникшей сущности, условия ее существования. Условия и основание в своем диалектическом единстве образуют стороны того целого, которое мы определили как вещь. Условие и основание, - отмечал Гегель, - это одно и то же существенное единство и как содержимое, и как форма. Они переходят друг в друга благодаря самим себе. Они полагают сами себя как снятые, соотносят себя с этим своим отрицанием и взаимно предполагают себя [Л. 68]. Диалектика отношения "основание - условия" определяет основное противоречие самой сущности: сущность, с одной стороны, есть основанное, а с другой - основание (т.е., то из чего должно быть поня-

то существование вещи). Как основание сущность есть нечто непосредственное, безусловное, как основанное - (т.е. вышедшее из недр условий) - она есть нечто обусловленное, опосредованное. Постоянное воспроизводство этого противоречия составляет движение сущности как субстанции вещи, в процессе которого она не только самопорождает себя, но и обретает существование.

Как определенность основанного, определенность сущности есть качество вещи, а как определенность основы - ее свойства. Поскольку условия генезиса сущности есть вместе с тем существенные условия ее существования, постольку свойства, выражающие определенность основы, есть существенные свойства. Свойства есть то, в чем сущность находится как основание, вместе с тем сущность наличествует в этих свойствах лишь как обусловленное основание.

Свойства - это собственные определения сущности, именно поэтому сущность не есть некая, лишенная определений основа, находящаяся по другую сторону своего внешнего существования. Лишь в своем свойстве, а, следовательно, и в самой своей сущности вещь имеет свое отличие от других вещей. В этом единстве внешности и существенности и состоит действительная диалектика бытия вещи.

Порождение сущности новой вещи всегда есть вместе с тем и процесс порождения существенных свойств этой вещи. Иначе говоря, свойства создаются, порождаются в определенных, генетических для возникающей вещи, отношениях вещей. Вместе с тем, не следует полностью отождествлять свойства и отношения, так как их диалектическое единство совсем не означает их абсолютного тождества. Отношение есть противоречие, и от того, как оно разрешается, зависит решение вопроса о судьбе свойства.

Всякая реальная вещь существует как система материальных отношений вещей, которые в статусе уже ее элементов производят сущность этой вещи (генетическая структура) и обнаруживают ее существование (структура функционирования). Поскольку в одном возможном содержится и его иное, возможное, по словам Гегеля, есть снимающее себя противоречие. В этом снятии противоречия оно и становится действительностью [Л. 68, 69]. Материальные вещи, взаимодействуя друг с другом, всегда имеют возможность выхода за границу своей определенности, к превращению в другое. В каждой вещи в силу ее противоречивой природы содержится возможность собственного отрицания и, стало быть, возможность иной действительности. До тех пор, пока не определен круг конкретных условий, благодаря которым возникает новая вещь или обуславливается существование уже данной вещи, указанные возможности существуют лишь абстрактно, выражая общую тенденцию бытия материального мира.

Можно говорить об абстрактной возможности, предоставляющей при соответствующих условиях различные случаи своего конкретного осуществления: при определенных конкретных условиях она осуществляется неизбежно. В этом случае условия ее существования выражают вместе с тем и необходимость ее осуществления. Применительно к генезису вещи это означает, что, когда налицо **полный круг условий**, новая вещь возникает неизбежно.

Таким образом, приходим к следующим выводам. Обусловленное внешним отнюдь не является только случайным. Понятие необходимости относится не только к тому, что уже произошло, существует, но и к тому, что должно осуществиться. Всякое случайное для данной вещи возможно лишь в плане абстрактной, а не конкретной возможности. Случайным для данной вещи оказывается, таким образом, все то, что не детерминировано возможностями, иными словами, природой самой именно этой вещи. Условия генезиса субстанциональной основы (сущности) вещи есть вместе с тем и необходимые условия ее существования, благодаря

которым сущность только и обретает определенность и обнаруживает свою самостоятельность как бытие вещи.

### 2.3.3. ВЕЩЬ В ПРОЦЕССЕ СВОЕГО ПРЕБЫВАНИЯ И РАЗВИТИЯ. ФУНКЦИОНИРОВАНИЕ ВЕЩИ

Перейдем к рассмотрению функционирования вещи. Вещь в действительности своего бытия представляет собой существенное единство в существенном множестве. Под существенным множеством понимают множество существенных свойств, которые необходимо присущи вещи, поскольку в известном смысле принадлежат самой сути вещи, и таким образом, вытекают, следуют из самой природы вещи. Именно в этом смысле существование вещи обоснованно. Вместе с тем, существование вещи обусловлено, причем обусловлено в прямом смысле этого слова. Отношения уже сформировавшейся сущности вещи с условиями ее существования есть **процесс ее функционирования** в качестве данной вещи. Именно система генетических отношений и отношений функционирования организует вещь как целое. Следовательно, **отношения организующие вещь, включают** в себя не все конкретное многообразие отношений, в которых она существует, а лишь **систему ее существенных отношений**. Представление о вещи как системе функционально-генетических отношений выявляет процессуальный характер действительного бытия вещи. Вещь в процессе своего функционирования непрерывно воспроизводит как свое основание (сущность), так и условия своего бытия. Благодаря этому она и сохраняет себя, пребывая в качестве данной вещи.

**Вещь есть**, таким образом, **процесс**, она **находится в непрерывном движении**, но в этом движении она сохраняет себя, представляя собой, по словам Ф. Энгельса "движение в равновесии" [Л. 182, 246]. В своем функционировании вещь есть не что иное, как пребывание (сохранение) в изменении, как сохранение одного и того же состояния движения. Любая материальная вещь в своем функционировании находится в непрерывном процессе превращения и самообновления. Характер этих изменений может быть различным, и в зависимости от той роли, которую играют изменения в бытии вещи, их принято разделять на изменения количественные и качественные.

**Количественные изменения**, как следует из самого определения, есть изменения количественной определенности вещи. Это изменения несущественные, обусловленные чисто внешними обстоятельствами бытия вещи и происходящие в рамках одной и той же качественной определенности. **Качество** есть единство (система) свойств и изменение количественной определенности вещи есть изменение интенсивности того или иного ее свойства. Эти изменения имеют (в рамках меры вещи) обратимый характер. Напротив, **изменения качественной определенности вещи необратимы** и имеют, таким образом, **направленный характер**. Для отражения такого типа изменения вещей в философии и науке выработано такое понятие, как - развитие [Л. 246, с. 181].

**Развитие есть процесс изменения качественной определенности вещи**. Оно носит необратимый, направленный характер и связано с разрешением противоречий, присущих сущности данной вещи. Отметим, что, как правило, выделяют два таких направления развития: прогрессивное и регрессивное. Между развитием вещей в рамках одного и того же уровня организации и развитием, связанным с переходом от одного уровня организации к другому, имеется явная связь. Механизм развития вещи по сути такой же, как и рассмотренный выше, механизм об-

разования сущности новой вещи. И это не удивительно, так как процесс развития существующей вещи есть в своем конечном итоге процесс образования новой вещи. Процессу развития вещи сопутствуют, с одной стороны, дальнейшая дифференциация и специализация (индивидуализация) структурных элементов, а с другой - рост их функциональной интеграции. Процесс развития вещи приводит к образованию более органичного единства ее структурных элементов, а значит, и к более высокой организации целостности вещи. Становление системы такой целостностью образует момент ее, системы, процесса развития [Л. 182, 246].

Развитие системы в направлении ее целостности не является процессом, в котором, так сказать, "гасятся" специфические краски конкретного многообразия образующих ее вещей. На развитие вещи оказывают влияние изменяющиеся условия ее функционирования. Условия существования вещей, которые существенно изменились в процессе ее функционирования, уже отличаются от условий ее первоначального генезиса. Это, в свою очередь, обуславливает обострение уже существующих противоречий сущности и способствует возникновению новых и, таким образом, изменение условий влияет на развитие вещи, на изменение ее качественной определенности, а значит и ее сущности.

Следовательно, процесс развития вещи можно характеризовать с двух сторон:

- в плане обогащения имеющейся сущности вещи этот процесс развития идет до тех пор, пока сущность не обретет наиболее полную, а значит и всеобщую, форму своего проявления;
- в плане изменения сущности вещи под воздействием изменившихся условий существования.

Темп процесса изменения сущности будет зависеть от того, насколько кардинально и всеобъемлюще изменяются условия ее существования. Противоречие разрешается гибелью стороны противоположности, исчерпавшей возможности своего развертывания, в своем бытии утверждается новое содержание - возникает новая вещь.

**Все существующие материальные системы есть самовоспроизводящиеся системы** (имеет место самодействие, самодвижение). Это не означает, что каждая такая система есть вместе с тем и саморазвивающаяся система. Из полученного выше факта того, что **вещь одновременно однокачественна и многокачественна** следует: вещь в процессе своего бытия может располагать многими линиями развития, наряду с ведущей, главной, которой оказывается всегда та, что наиболее полно реализует возможности заложенные в ее сущности. Понятие развития (как направленное изменение) имеет смысл применительно только к конечным системам и, видимо, неприменимо к материальному миру (движущейся материи) в целом. В этом смысле, развитие выступает как конкретная форма самодвижения материи.

#### 2.3.4. ПРИЧИННОСТЬ И ДЕТЕРМИНИЗМ

Действительность вещи есть система материальных отношений вещей, в которых она возникает и обнаруживает свое существование, т.е. система отношений порождения и функционирования. Диалектически понятый детерминизм заключается, прежде всего, в признании всеобщей взаимосвязи, определенности и обусловленности объективно реальных вещей.

Материальная вещь существует как единство (т.е. система) отношений порождения (генетическая структура) и функционирования (функциональная структура), она суть система функционально-генетических отношений вещей, что следует из

функционально-генетической природы самого взаимодействия. Следовательно, все многообразие материальных отношений, в которых вещь возникает и обнаруживает свое существование, есть отношения детерминации. Отношения порождения (причинные) есть одни из типов детерминации, они составляют главные факторы, детерминирующие бытие данной вещи. Отметим, что, кроме отношений порождения, еще выделяют отношения обоснования и отношения обусловленности.

Приходим к выводу, что причинные отношения имеют место там, где идет процесс формирования сущности новой вещи, а также при воспроизводстве или кардинальном изменении сущности уже существующей вещи. Все же (те) изменения, которые происходят в рамках меры данной вещи, хотя и обусловлены взаимодействием данной вещи с другими вещами и существуют на базе ее собственных, генетических отношений, не являются причинными. Отсюда следует, что далеко не всякое отношение детерминации является причинным, подтверждая справедливость того факта, что причинная связь является лишь частицей мировой универсальной связи.

Строго говоря, наступлению действия предшествует не причина, а носители причинного отношения. С учетом этого, можно говорить лишь о том, что причина предшествует действию. Однако, следует помнить об условности такого разделения: причина есть то, что имеет действие, осуществляет себя в действии и с прекращением деятельности причины прекращает, следовательно, свое существование и действие. В понятиях "причина" и "действие", в сущности, фиксируются различные моменты, фазы развертывания отношения причинения как процесса становления новой вещи, нового существенного свойства. Без условий нет причинного отношения (причины). Строго говоря, причинение есть особая форма разрешения противоречия, связанная с выходом за рамки качественной определенности соотносящихся вещей (носителей причинного отношения), с возникновением новой вещи (или нового свойства). Причинное отношение не перестает быть отношением детерминации. Так как сами носители причинного отношения выступают условиями этого процесса, то можно сказать, что действие (вещь, свойство) обуславливается через носителей процесса причинения.

Причина есть генетическое отношение, и именно благодаря этой своей особенности она отличается от основания и условия, которые обосновывают и обуславливают, но ничего не производят. Следовательно, причинное отношение характеризуется наличием в нем противоречия особой остроты. Подчеркнем, что не может быть разных причин у одного и того же действия, но могут быть разные носители одного и того же причинного отношения, если иметь в роли носителя причинного отношения вещь в целом. Точно так же одна и та же причина (причинное отношение) не может порождать, при одних и тех же строго фиксированных условиях, разные действия.

Функциональные структурные отношения возникают и существуют в рамках бытия вещи лишь на базе ее генетических структурных отношений. Они суть разные аспекты, уровни рассмотрения взаимодействия вещей как отношения детерминации. При этом, причинные отношения лежат в основе всех форм связи, являясь в этом смысле "общим аспектом детерминации явлений". Причина обосновывает функциональные отношения в буквальном смысле этого слова, обнаруживая себя как причина в функциональной зависимости вещей. Всякая функциональная связь (отношение) оказывается всегда (прямо или опосредованно) причинно обоснованной, что не означает, однако, процесса порождения, ибо, как было отмечено выше, основание, как и условия, ничего не производит.

Связь является фундаментальным понятием физики, отображающим вещь в каждый момент ее бытия. Под **состоянием** обычно понимают множество свойств,

присущих данной вещи в данный (конкретный) момент времени ее жизни. Поскольку вещь в своем существовании есть непрерывный ряд последовательно сменяющих друг друга ее состояний, постольку можно говорить о связи состояний одной и той же вещи, а также о связи состояний разных взаимодействующих друг с другом вещей.

Генетическая связь есть функциональная связь, при этом любая связь причинно обоснована. Однако, это не означает, что каждая из взаимосвязанных сторон отношения является причиной изменения другой. Понятием связи отображается внешний аспект взаимодействия и потому "причинная связь" не может быть отношением порождения, хотя она, есть генетическая связь. Между явлениями одной и той же сущности также можно установить генетическую связь, но это не означает, что они находятся в причинном отношении друг к другу. В основе генетической связи лежит определенное причинное отношение, но сама она не есть процесс порождения. В отличие от обычной функциональной связи, которая выражает изменения свойств соотносящихся вещей, не выходящие за рамки их меры, **генетическая связь есть характеристика взаимной зависимости** (связанного изменения) качественной определенности соотносимых вещей. То есть, генетическая связь - это такая связь, в которой причинное отношение обнаруживает себя непосредственным образом.

Вероятность есть количественная характеристика, мера возможностей вещи, зависящая от изменяющихся условий, детерминирующих вещь. Хотя понятие вероятности имеет определенное объективное содержание, оно, тем не менее, есть способ описания возможного движения системы (вещи). Другими словами, действительная линия реализаций той или иной возможности при данных конкретных условиях не носит вероятностного характера. При данном круге условий возможность реализуется однозначно, т.е. единственно возможным образом. Причина и действие неразрывны во времени, т.е. с окончанием процесса причинения заканчивает свое существование и действие. Причинное отношение многозначно не по форме своего действительного осуществления, а по форме своего ожидаемого функционального обнаружения, поскольку условия обнаружения причин отношения действительно достаточно неоднозначны. Эта неоднозначность действия есть неоднозначность, неопределенность его количества, которая не выходит за рамки его меры. Отношение причина - действие не носит вероятностный характер и, стало быть, никакой особой "вероятностной" причины не существует, а есть лишь вероятностный способ описания ожидаемого обнаружения действия при некоторой неоднозначности условий осуществления причинного отношения. Таким образом, принцип причинности, конкретизируя общую познавательную установку на познание вещей в их взаимной связи и обусловленности, требует идти в научном познании дальше - от познания внешней обусловленности вещей, т.е. их функциональных связей, к генезису их сущности, к причинам, порождающим их.

### 2.3.5. ЗАКОНОМЕРНЫЙ ХАРАКТЕР БЫТИЯ ВЕЩЕЙ

Рассмотрим более подробно закономерный характер бытия вещей. В разнообразии и смене явлений **закон** есть нечто постоянное, пребывающее, тождественное себе, в этом смысле закон есть единство в явлении (существенное единство) и составляет его существенное содержание. Отметим, что закон и сущность - понятия однопорядковые, но не тождественные. **Закон есть отношение между сущностями**, или существенное отношение, и потому любой **закон не выражает**

**всей сущности, а лишь какой-то ее момент**, сторону, аспект. Именно поэтому, философы говорят, что всякий закон узок, неполон и приблизителен [Л. 169, 246]. Следовательно, не только явление, но и сущность по своему содержанию значительно богаче закона. Поскольку закон - это существенное отношение, т.е. необходимая, устойчивая связь в явлении, постольку структурные отношения, организующие вещь, есть закономерные отношения.

Так как законы функционирования определяют особенное в явлении, специфическую форму бытия сущности, то законы генезиса определяют генеральную необходимость возникновения и развития самой сущности. Законы генезиса (развития) вещи можно обозначить как причинные законы, а законы функционирования как функциональные. Подчеркнем, что законы генезиса лежат в основе функциональных законов. Выявление законов функционирования вещи позволяет подойти к раскрытию содержания ее сущности, выявить закон ее образования и, таким образом, объяснить бытие вещи. Системно организованное бытие вещи всегда носит закономерный характер, основанный на всеобщем универсальном взаимодействии вещей.

### 2.3.6. ОГРАНИЧЕННОСТЬ ВЕЩИ И КОНЕЧНЫЙ ХАРАКТЕР ЕЕ БЫТИЯ

В основе современных представлений о конечности всякого конкретного существования находится, выявленный в физике, фундаментальный факт конечности скорости, а значит и времени всякого взаимодействия. В силу конечности времени причинного взаимодействия число его опосредований также конечно. Следовательно, время жизни вещи есть время существования системы взаимодействия вещей, которые ее образуют [Л. 246, с. 196]. Вещи возникают в качестве данных вещей и уничтожаются, **всякая вещь имеет свое, присущее ей время жизни**, т.е. всякая вещь конечна. Конечность числа отношений, образующих данную вещь, обуславливает ее отграниченность от других вещей, в силу чего вещи в своем реальном бытии оказываются и конечными, и ограниченными.

В силу конечности числа отношений, опосредствующих образование и существование данной конкретной вещи, **число существенных свойств любой вещи**, как бы оно не было велико, **конечно** и в принципе - исчерпаемо. Следовательно, вопрос об абсолютной неисчерпаемости свойств имеет онтологический смысл лишь применительно к материальному миру в целом. Применительно же к бытию конкретных вещей он имеет чисто гносеологический аспект - неисчерпаемость свойств познаваемой вещи оказывается существенным образом связанной с исторически условными границами самого процесса познания. Так как существование познающего разума конечно, следовательно, с точки зрения времени его (разума) существования, количество познаваемых свойств реальных вещей является неисчерпаемым (но все же конечным, так как мы имеем два конечных множества, в одном из которых элементов гораздо больше, чем в другом). Конечные, в рамках определенных отношений природы, вещи в системе отсчета, связанной с жизнью познающего разума, оказываются для последнего практически бесконечными (в смысле неисчерпаемости). Следует учитывать и то, что многие вещи погибают, так и не успев раскрыть во всей полноте свою сущность.

Существует возможность и такой ситуации, когда в процессе своего развития вещи сущность обретает наиболее полную форму своего проявления, создавая тем самым условия для исчерпывающего познания данного предмета. "Исчерпаемость" свойств таких вещей определяется временем их жизни. О неисчерпаемости

свойств конкретных вещей в онтологическом плане следует, видимо, говорить только в смысле тех потенциальных возможностей, которые заложены в условиях их порождения и существования. Но реализация этих возможностей будет означать появление уже иных вещей с иными границами своего бытия.

Материальный мир как мир вещей, в этом смысле, оказывается действительно неисчерпаем, в силу тех бесконечных возможностей, которые таит в себе вечно движущаяся и развивающаяся материя. [Л. 246]. Получаем, что природа бесконечна, но она бесконечна как многообразие вполне конечных и ограниченных в своем конкретном бытии вещей. Мир бесконечен в целом и конечен в каждом отдельном фрагменте своего бытия. Эта конечность и ограниченность не абсолютны в силу наличия в природе мировой универсальной связи. Познающее мышление оказывается перед противоречием: с одной стороны, задача исчерпывающим образом понять систему мира в ее совокупной связи, а с другой - невозможность полностью решить эту задачу. Впрочем, именно это противоречие является, по словам Ф. Энгельса [Л. 182], главным рычагом всего умственного прогресса и разрешается каждодневно и постоянно в бесконечном прогрессивном развитии человечества.

### 2.3.7. МАТЕРИАЛЬНЫЙ МИР КАК МИР ВЕЩЕЙ И СОВОКУПНЫЙ ПРОЦЕСС

Всякая вещь оказывается системно организованной как данное конкретное целое и в своем существовании необходимо предполагает бытие других вещей. Материальный мир представляет собой, таким образом, бесконечное многообразие различным способом (образом) организованных систем (вещей) и сам, в силу универсального взаимодействия вещей, суть - система. Различным формам движущейся материи соответствуют различные уровни ее организации. Введение понятия формы движения материи позволило развить системное понимание движущейся материи. Отметим, что современное научное знание выделяет такие формы движения материи, как физическая, химическая, биологическая и социальная. При этом, физическая форма является своеобразным фундаментом других форм движения материи. Соответствующие формам движения материи уровни организации материи образуют сложную иерархию структурных образований материального мира. Кроме того, для движущейся материи характерно диалектическое единство двух противоположных тенденций: процессов организации и дезорганизации.

Идея круговорота в природе как формы ее саморазвития, отражая материальное единство мира, всеобщий для всего материального мира характер законов его движения (законы диалектики), необходимо приводит к представлению о "мире в целом". **Движущаяся материя во всем бесконечном многообразии своих структурных форм, в принципе не может быть дана человеку эмпирически, быть объектом эмпирического исследования.** Еще И. Кант говорил, что если мир в целом не дан в опыте, то он дается посредством опыта [Л. 134]. Любая материальная вещь, будучи эмпирически дана познающему субъекту той или иной своей стороной, гранью, как целое является объектом лишь теоретического мышления. Подобно этому движущаяся материя (материальный мир), познаваемая человеком в отдельных ее проявлениях, отображается синтезирующей теоретической мыслью в представлении о мире как закономерном целом. В этом смысле, "мир в целом" и является объектом теоретического познания - философии. Разумно предположить, что материальный мир, системный в каждом своем конкретном фрагменте, не может (в силу универсального характера движения, яв-

ляющегося способом его бытия) не представлять собой в определенном смысле систему и в целом, не может не быть "целым миром". Таким образом, традиционная, называемая "составной", модель организации материального мира в этом случае "не работает". Существует также гипотеза "бутстрапа" (bootstrap) Дж. Чу и Фраутчи, согласно которой все сильновзаимодействующие частицы целиком обязаны своим существованием тем же силам, благодаря которым они взаимодействуют друг с другом [Л. 246, с. 203]. Иными словами, каждая частица участвует в порождении других частиц, которые в свою очередь, порождают ее.

Идея "бутстрапа" дает возможность нарисовать несколько иную картину системной организации мира материальных вещей. При такой форме организации материальных систем отпадает необходимость в существовании систем более обширных, чем рассматриваемые, и содержащих последние в качестве элементов, из которых они состоят. С точки зрения физических представлений не исключено определенное единство этих (микрокосм - элементарные частицы и космологические системы), столь отдаленных друг от друга структурных уровней материального мира. В основе этого единства лежит круговорот материи. Кроме того, существует гипотеза "фридмона" М.А. Маркова о том, что вселенная в целом может оказаться микрочастицей, а микрочастица может содержать в себе целую вселенную. Именно такую гипотетическую систему в ее конечном состоянии М.А. Марков и назвал "фридмоном" [Л. 246, с. 205]. Интересно, что в духе рассмотренной выше гипотезы "бутстрапа" движущуюся материю можно представить как бесконечно многообразную систему взаимодействующих друг с другом "фридмонов". Таким образом, проблемы элементарных частиц и космологии оказываются связанными в единый узел.

Рассмотрим правомерность понятия "мир в целом" и необходимость отличия его от понятия "вселенная". **"Вселенная"** - есть понятие космологии и представляет ту область материального мира, которая охвачена наблюдениями средствами астрономии. Развитие средств наблюдения будет раздвигать рамки такой вселенной. Следовательно, "вселенная" - это та часть (область) материального мира, которая попала в сферу человеческой деятельности. Подчеркнем, что понятия "материальный мир" и **"материальная действительность"**, т.е. **"объективная реальность"**, не являются синонимами, поскольку первое служит для обозначения всей движущейся материи (мир в целом), последнее же - для обозначения той области, того фрагмента движущейся материи, который находится в сфере человеческой практики.

Таким образом, **"объективная реальность" - это фрагмент, часть "материального мира"**. Если материальная действительность является объектом всего человеческого знания, то материальный мир (мир в целом) есть объект философского знания. Если теоретическим отображением Вселенной является физическая картина мира, то теоретической моделью мира в целом можно считать определенную философскую картину мира. В свою очередь, понятие "материальный мир" или "мир в целом" есть конкретизация понятия "движущаяся материя". Внутреннее единство мира - это его материальное единство, обнаруживающее себя в существовании мировой универсальной связи всего со всем, в существовании всеобщих законов. И единство мира, и существование мировой универсальной связи обеспечиваются взаимодействием материальных вещей.

Таким образом, **материальный мир системно организован благодаря самому способу своего бытия - движению**. Как целое мир есть особый тип системной организации, получающий все бесконечное многообразие своих определений в отношении с самим собой. В этом отношении с самим собой (когда движение есть

способ его бытия) он обнаруживает противоречивость своего системного бытия: материальный мир бесконечен как целое и конечен как многообразие в каждом отдельном своем фрагменте. Поэтому, **бытие мира в целом есть его существование в виде бесконечного многообразия качественно различных вещей**. Материальный мир системно организован в каждом отдельном своем фрагменте и в целом. Организация есть форма бытия мира вещей и имеет место на всех уровнях материального движения, присуща всем формам движущейся материи.

Рассмотрение диалектики материальной вещи выявляет процессуальный характер ее бытия как системы, что позволяет в философской картине мира представить последний как **мир вещей** или совокупный процесс. Великая основная мысль, выражающая суть диалектико-материалистического осмысления бытия мира, заключается в том, что **мир состоит не из готовых законченных предметов, а представляет собой совокупность процессов** [Л. 182, 246]. Системное представление любого фрагмента материального мира есть его отображение в системе понятий "вещь - свойство - отношение". Материальный мир есть мир вещей "самих по себе" в их гносеологической независимости от познающего субъекта. Вселенная, материальная действительность - это, так сказать, уже мир вещей "для нас", мир, в известном смысле освоенный человеческой деятельностью.

## 2.4. ВЕЩЬ, СВОЙСТВО, ОТНОШЕНИЕ И ПРОЦЕСС ПОЗНАНИЯ МИРА

### 2.4.1. ВЕЩЬ, СВОЙСТВО, ОТНОШЕНИЕ КАК ФОРМЫ ДЕЯТЕЛЬНОСТИ

Суть научного, материалистического подхода к решению вопроса о том, как вообще объективным образом в нас входит то, что находится вне нас, в понимании человека "как деятельного природного существа" [Л. 49, 53, 119, 169, 182, 186, 246, 254, 267, 290, 303]. Рассмотрим деятельность как основу категориального отражения мира. Прежде всего, человек в качестве живого природного существа с одной стороны, наделен природными силами, жизненными силами, являясь деятельным природным существом. А с другой стороны, в качестве природного, чувственного существа человек является страдающим, обусловленным и ограниченным существом. То есть предметы влечений человека существуют вне его, как не зависящие от него предметы, но эти предметы суть предметы его потребностей, это необходимые, существенные для проявления и утверждения его сущностных сил предметы. Таким образом, человек относится к природе предметным образом, в соответствии со своими потребностями. В этом отношении человек превращает природу в свое "неорганическое тело, поскольку она служит, во-первых, непосредственным жизненным средством для человека, а, во-вторых, материей, предметом и орудием его жизнедеятельности" [Л. 246]. Перерабатывая природу, человек создает предметный мир, в силу чего природа оказывается уже его (человека) производением и его действительностью. Материальные вещи выступают для человека как предметы его потребностей, т.е. приобретают смысл и значение постольку, поскольку включаются в сферу его жизнедеятельности. Так возникает мир человеческой природы, человеческой действительности. Тогда, **предметный мир - это материальный мир вещей, попавших в сферу человеческой деятельности**. Особенность жизнедеятельности человека заключается в том, что она есть сознательная деятельность. Практическое отношение человека к природе, его деятельность возможна лишь через его отношение к другим людям,

т.е. как общественная деятельность, ибо только в этом случае человек выступает как родовое и потому сознательное существо.

**Содержанием сознания является знание и потому познание составляет необходимый момент, неотъемлемую сторону практического отношения человека к миру.** Будем исходить из того, что и практика, и познание есть две неотъемлемые друг от друга стороны (аспекты) единого процесса общественно-исторической деятельности человека. Фундаментальный факт состоит в том, что практическая деятельность есть сознательная, целенаправленная деятельность, посредством которой человек получает возможность предметно относиться к природе. Сторона деятельности человека, характеризующаяся превращением свойств предмета в субъективные возможности человека, реализуемые в его способностях, получила название "распредмечивание".

**Распредмечивание** есть раскрытие объекта субъекту и вместе с другой стороной практического отношения человека к природе, выражающейся в превращении субъективных возможностей человека в свойства предмета (опредмечивание), вместе образуют то, что, мы характеризуем как общественную форму движения материи, или деятельность. Диалектическое единство этих сторон деятельности означает, что опредмечивание, т.е. практически-преобразующая деятельность, есть в то же время и распредмечивание, т.е. идеально-преобразовательная деятельность, и наоборот. В силу предметного характера деятельности она есть определенная форма самого материального бытия. Определения материального бытия, объективно организованного как система вещей (мир вещей), необходимо являются и определениями самой деятельности. Формы, в которых существует знание, пронизывающее всю деятельность, действительно есть вместе с тем и формы самой деятельности [Л. 246]. **Знание - это понятийная форма предметного отношения человека к природе,** которая могла исторически возникнуть в ходе формирования человека как деятельного существа только в случае, если она позволяла последнему осуществлять свою деятельность в соответствии с определениями и логикой бытия самого предметного мира.

#### 2.4.2. СТУПЕНИ ОТРАЖЕНИЯ И УРОВНИ НАУЧНОГО ПОЗНАНИЯ ВЕЩЕЙ

Являясь природным существом, человек в своей деятельности (и в познании мира вещей) исходит, прежде всего, из чувственности: из чувственной потребности, определяющей мотивацию его действий, и из "чувственного сознания". К **формам чувственного сознания** человека обычно относят **ощущение, восприятие и представление**, о которых можно говорить как о разных **уровнях человеческого отражения вещей**. Если ощущения обеспечивают непосредственный, но "точечный" контакт человека (субъекта) с предметом, то восприятия и представления дают уже в известной мере целостное отображение материальных вещей. Человек не мог бы успешно осуществлять свою деятельность, если бы в процессе биологической эволюции у него не была бы сформирована такая способность к чувственному отражению, которая обеспечивает восприятие предмета в единстве и дискретности (дифференцированности) наиболее важных для жизни человека свойств вещей.

Отметим, что чувственному сознанию в целом свойственны и анализ, и синтез, и обобщение - эти необходимые компоненты процесса познания. При этом, в ходе развития человека (истории) на базе форм чувственного отражения предметного мира формируется и развивается способность к рациональным (мыслитель-

ным) формам отражения действительности (мышление в понятиях). Таким образом, **чувственное и рациональное в познании оказывается органично связанным**, образуя собой единый процесс человеческого отражения материального мира [Л. 246, с. 213].

Это единство легко обнаруживается, в частности, в том фундаментальном факте, что смысловое значение слова (понятия) всегда требует воплощения в чувственно-наглядном образе (понятие без такого образа, вообще говоря, не существует), а теоретическое мышление, исходя из практики, которая есть, прежде всего, чувственная деятельность, непременно вновь возвращается к ней. С точки зрения научных представлений, целостное отображение материальных вещей имеет место даже на разных, - сенсорный и перцептивный, - уровнях чувственного отражения. На сенсорном уровне отражения происходит обнаружение вещи и выделение ее из среды, выделение признаков, которые являются сигнальными по отношению к существенным свойствам вещи. Перцептивный уровень характеризуется еще более развитыми формами анализа и синтеза, происходит выделение частей предмета, обнаружение отношения между частями и целым, фиксация структуры и формы, непосредственное соотнесение предмета со средой.

Необходимым моментом этого уровня является наличие памяти. Кроме того, на этом уровне продолжается формирование целостного образа отражаемой вещи и оценки ряда ее интегративных характеристик. Считается, что "тайна" целостности чувственного образа заключается в известной мере в том, что ощущения и восприятия различных "модальностей" (зрительных, осязательных и т.д.) отражают одно и то же, имеют одно и то же гносеологическое содержание. Таким образом, на уровне чувственного отражения мира вещей формирование целостного чувственного образа идет через синтез выделенных в процессе отражения отдельных сторон бытия вещи. Степень абстракции и обобщения, при этом, здесь - минимальная. Более того, не только целостный, но в известной степени и обобщенный образ, т.е. существенно и необходимое в бытии вещей дано здесь еще в нерасчлененном (неявном) виде.

Чувственное познание образует первый виток спирали, характеризующий процесс познания, второй виток начинается с представлений, которые дают не только целостный, но в известной мере - обобщенный образ предмета, представляя его как чувственно конкретное. Второй виток характеризуется более высоким уровнем абстракции и обобщения (синтеза), обусловленными способностью человека к речевому общению. Чувственный образ, будучи связанным с тем или иным словом, становится теперь его смысловым значением, образуя понятие. Подчеркнем, что именно с появлением языка начинается подлинная история познания материального мира.

### 2.4.3. РОЛЬ И ФУНКЦИИ ЯЗЫКА

Язык возникает в процессе деятельности человека, характеризуя развитие последней от ее инстинктивных форм к сознательной. Свою основную социальную функцию быть средством общения и сообщения язык мог успешно выполнять, лишь становясь орудием осуществления различных операций абстрагирования и обобщения. Известно, что трудовая деятельность человека возможна лишь тогда, когда выделен предмет и орудия деятельности, что предполагает наличие мысли о цели (плане) деятельности и результате труда, которые не даны непосредственно. **Без абстракции нет ни обобщения, ни синтеза, а следовательно, и знания вещей.** Диалектический характер этого метко выразил В.И. Ленин:

отойти, чтобы вернее попасть [Л. 169]. При этом, под конкретным понимается бытие вещи как единство многообразного (т.е. действительное бытие), под абстрактным - все вообще выделенное, обособленное, существующее "само по себе", в своей относительной независимости от всего другого, иными словами, сторона, аспект или часть действительного целого.

Подчеркнем, что процесс формирования отображающей функции языка был длительным и сложным. Первоначально язык служил, формировался как средство общения. Процесс превращения человеческого языка в предметный язык заключался во "втягивании" вещей в сферу языкового общения. Слова не обозначали вещей, когда вещи уже были привлечены для обозначения и, таким образом, дифференциации слов. Именно то обстоятельство, что большинство действий предметны и их невозможно смешать в силу объективного различия реальных вещей, предопределило эту **знаковую функцию вещей**. Исполняя роль вспомогательных средств общения, вещи постепенно втягивались в мир слов, подготавливая тем самым переворот в их взаимоотношении: в силу взаимооднозначного соответствия уже слово становится обозначением той или иной вещи. Происходит процесс именованности вещей, т.е. за тем или иным словом (именем) закрепляется чувственный образ вещи и последние, таким образом, становятся теперь уже предметом мышления.

Как мы видим, язык формируется в процессе деятельности как предметный, вещный язык, что и определяет центральное положение вещи в его категориальной структуре.

**Функции языка проявляются, прежде всего, в выделении вещей как предметов деятельности.** Не случайно, первоначально в языке не существовало особой категории слов, которые бы выражали признаки (свойства) предметов. Свойства вещей, которые практически выявляются и раскрываются в предметной деятельности человека, получают свое название по имени того предмета (вещи), который с точки зрения говорящих является преимущественным носителем этого свойства. Например, твердый - камень, красный - кровь и т.д. В процессе предметной деятельности те или иные фрагменты материального мира выделяются формами чувственного отражения как нечто единое, неразложимое на свои отдельные определения и именуется. Так объективный мир предстает перед познающим мышлением как многообразие определенных нечто, т.е. вещей. Последующий процесс отражения связан с абстрагирующей и синтезирующей деятельностью мышления, когда на базе языка создаются понятия о свойствах и отношениях этого поименованного нечто. Понятия формируются как мысль, в которой отражаются отличительные, специфические свойства предметов действительности и отношения между ними.

В отличие от представления, дающего нам относительно целостный чувственный образ предмета, отнесенный, правда, к этому "здесь" и этому "теперь" вещи, в понятии предмет мыслится лишь с точки зрения вполне определенных, выделенных в нем свойств и отношений, при этом мы отвлекаемся от других свойств вещи и от конкретной ситуации, в которой она воспринималась. Мысль, выраженная в слове (понятие), заключает в своем содержании абстрактно-всеобщие определения познаваемой вещи. Это открывает перед мышлением величайшие возможности, позволяя "схватить", отобразить мыслью "неуловимое", чувственно не воспринимаемое в бытии вещей. Диалектика процесса познания заключается, таким образом, в том, чтобы познать вещь в действительности ее бытия, т. е. как единство многообразного - конкретное. Для этого необходимо вначале мысленно расчленив вещь, выделить различные ее абстрактные определения (соответственно: понятия, свойства, отношения), а уже затем синтезировать их в единое целое – в

понятие о вещи. Именно так мысль совершает в процессе познания вещей (их свойств и отношений) восхождение от абстрактного к конкретному.

Некоторые философы считают, что мы никогда не познаем действительное конкретное во всей его полноте. В самом деле, процесс познания, являющийся необходимым моментом, стороной практической деятельности, ограничен в каждый момент своей истории, как исторически ограничена и сама деятельность людей. Соответственно, исторически условны и ограничены основания абстрагирующей и обобщающей деятельности мышления, что проявляется, в частности, в различных уровнях абстракции, закрепляемой в языке.

Обычно выделяют следующие основные виды абстракции [Л. 246]:

- абстракцию отождествления,
- изолирующую (аналитическую) абстракцию,
- процесс конструктивизации (идеализации),
- абстракции актуальной бесконечности и потенциальной осуществимости.

Различные уровни абстракции означают прежде всего различный уровень отождествления и вычленения (отвлечения) отдельных сторон, моментов (т.е. свойств, отношений) бытия вещей, что, в свою очередь, обуславливает различный уровень обобщения и идеализации (конструктивизации). Различный уровень процесса конструктивизации приводит к образованию идеальных конструкторов ("абстрактных объектов"), которые и воплощают (в различной степени и мере) идею конкретного в познании.

"Конструкт" формируется в процессе познания, представляя собой, **знак** или **идеальный образ** (модель) реальной вещи, т.е. выполняет функцию "заместителя". В каждом цикле познания вещи посредством "восхождения от абстрактного к конкретному" неполное конкретное, получающееся в результате этого движения мысли, оказывается в силу своей неполноты, односторонности абстрактным. И уже в качестве абстрактного объекта вновь становится предметом познания, осуществляемого, однако, уже на ином (более высоком) уровне абстракции и обобщения (синтеза) и так далее до тех пор, пока не будет получено конкретное. Где под "конкретным" понимают знание действительного бытия вещи во всей его полноте. Гарантией адекватности (истинности, объективности) получаемого при этом знания о мире является постоянное корректирование процесса познания практикой. Процесс восхождения от абстрактного к конкретному, совершаемый на разных уровнях отвлечения и обобщения, является общей закономерностью процесса познания. Это направление движения мысли в познании вещей от выявления отдельных свойств, отношений к их органическому синтезу определяется объективной диалектикой самой вещи. При этом, вещь является одновременно носителем свойств, отношений и их единством (системой).

**Знание, получаемое при этом, фиксируется в содержании понятий свойств и отношений конкретных вещей, в научных идеях (принципах) и теоретических системах.** Представляется возможным говорить об абстрактно-всеобщих и конкретно-всеобщих определениях бытия, соответственно, о понятиях абстрактных и более конкретных, содержащих, синтезирующих в себе большее богатство особенного, чем первые. В процессе абстрагирования и обобщения создаются понятия о свойствах и отношениях вещей (мира вещей). Из чего следует, что о каком бы уровне абстракции и обобщения познающего мышления мы не говорили, речь всегда идет, в конечном счете, об одном и том же - вещах (мире вещей). Иными словами, **понятие вещи оказывается самым конкретно-всеобщим понятием во всей категориальной структуре знания.**

В своем движении от категории к категории, которые представляют собой "последнюю степень абстракции всякой вещи" [Л. 182], мышление синтезирует содержащееся в них знание в единое представление о вещи как целом. В логическом плане это означает отображение вещи как единства характеризующих ее существенных признаков - определение понятия вещи в форме суждения. Тот факт, что все наше знание о мире существует лишь как знание о вещах, проявляется даже в том, что мы не можем образовать понятия о том или ином конкретном свойстве или отношении, не рассматривая его в качестве (в данном случае - идеальной) вещи. Именно таким образом сформированы, например, понятия тяжести, цвета, энергии и т.д. Объективные основания для такого рода логической операции заключаются в диалектической природе бытия реальных вещей (в смысле: вещь как единство многообразного, система) и самого процесса познания. Процесс восхождения от абстрактного к конкретному в познании есть всегда процесс образования понятия, понятия о вещи.

**Понять, то есть объяснить бытие вещи - это и значит образовать понятие о вещи** [Л. 246, с. 220]. Абстрактное и конкретное представляют собой формы теоретического воспроизводства вещи в процессе познания. Благодаря этой способности человеческого мышления, исторически формирующейся в процессе практической деятельности, последняя, становится осознанной деятельностью. Как необходимая, имманентная деятельности сторона, познание исходит из практики, и всякий раз возвращается к практике, служа ее интересам. Учитывая связь с практикой и роль, которую играют теоретические построения в общем процессе движения познающего мышления от абстрактного к конкретному, обычно выделяют два уровня познания: эмпирический и теоретический. Это и есть уровни мысленного знания, а их отношение друг с другом не следует отождествлять с отношением чувственного и рационального (чувственности и мышления).

На эмпирическом уровне познания знание о вещах выражено в понятийной форме и представлено всякого рода обобщениями, законами (правилами) и гипотезами. Основная особенность эмпирического уровня заключается в том, что познаваемая вещь дана здесь субъекту непосредственно, в отличие от теоретического уровня, где исходный для познания предмет (даже будучи представлен в чувственно-наглядной форме, например: геометрическая фигура, структурная формула и т.п.) все равно есть результат деятельности мышления и представляет собой некий идеальный "конструкт", или абстрактный объект. Кроме того, граница между непосредственным и опосредованным весьма условна.

Различение эмпирического и теоретического уровней познания именно по характеру субъект - объектной связи представляется наиболее целесообразным, поскольку в этом случае более полно учитываются различия, специфика, как в способе возникновения, так и в характере функционирования знания. В.И. Ленин формулировал это так: "чтобы понять - нужно ... от эмпиризма подниматься к общему" [Л. 169]. В движении познающей мысли теоретическое мышление принимает, по словам К. Маркса [Л. 182], в различные времена различные формы и вместе с тем очень различное содержание. Это находит свое выражение в существовании таких форм развивающегося знания о вещах (мире вещей), как определение, идея, теория, наконец, научная картина мира.

Таким образом, можно сделать следующие **выводы**. Рассматривая объективную диалектику бытия вещи, мы выяснили, что вещь есть одновременно и единое, и многое (единство многообразного), и система, и элемент какой-либо другой системы. Диалектика бытия вещи заключается в ее самодвижении (саморазвитии). Логической формой, отображающей этот процесс, является движение

понятий, общий закон которого - **восхождение от абстрактного к конкретному**. Объективная диалектика бытия вещи определяет и тот чрезвычайно важный для понимания процесса познания факт, что материальные вещи ни в объективном, ни в субъективном смысле непосредственно не даны человеку адекватным образом. Вот почему процесс постижения вещи в действительности ее материального бытия есть процесс, на разных этапах и уровнях которого **вещь предстает то как предмет, то как объект познания**.

#### **2.4.4 ВЕЩЬ КАК ОБЪЕКТ - РЕАЛЬНОСТЬ**

Материальный мир, будучи вовлечен в сферу человеческой деятельности, даже не изменяя своей природной формы, оказывается преобразованным в социальном смысле, ибо, включаясь в процесс жизнедеятельности человека, становится общественным, т.е. социально значимым, миром вещей. Природа вещей, попавших в сферу человеческой деятельности, оказывается двоякой: с одной стороны они существуют до и независимо от человека, с другой стороны - несут на себе печать деятельности человека (субъекта). Бытие вещей до и без человека принято называть бытием вещей самих по себе или объективным бытием. Тогда **мир вещей, попавших в сферу человеческой жизнедеятельности, следует рассматривать как объективный мир** (объективная реальность), или мир объект-реальностей.

**Под объект-реальностью принято понимать вещь саму по себе, но ставшую объектом человеческой деятельности** [Л. 246, с. 222]. Вещь как объект-реальность есть исходный пункт познания. Целесообразность введения такого понятия диктуется тем обстоятельством, что объекты практической и познавательной деятельности человека, вообще говоря, не совпадают, выражая противоречивость реального процесса познания. В деятельности объект-реальность выступает всегда с той или иной своей стороны и, выражая таким образом конкретную отнесенность объект-реальности к субъекту, есть уже предмет его деятельности. Будучи вовлеченной в сферу человеческой деятельности, вещь тотчас обнаруживает противоречивость своего системного бытия, выступая одновременно и как объект, и как предмет деятельности.

Как предмет деятельности вещь служит для удовлетворения нашей практической и познавательной потребности: в этом смысле предметное бытие вещи есть ее бытие как "вещи для нас". **Задача познания заключается в том, чтобы синтезировать это многообразие предметных определений вещи, выявляемое в процессе чувственно-предметной деятельности, и, таким образом, воссоздать картину ее действительного бытия**. Как конечная цель познания объект-реальность есть объект. Действительное бытие вещи есть ее системное бытие и как объект познания всякая вещь требует системного, целостного отображения. Таким образом, **вещь как целое, как система - является исходным пунктом и конечной целью познания**.

#### **2.4.5. ВЕЩЬ КАК ПРЕДМЕТ ПОЗНАНИЯ**

Как известно, человек относится к вещам в процессе своей деятельности активно, т.е. целенаправленно, избирательно. В ходе этого процесса природные вещи обретают новые свойства, связанные уже с их существованием в качестве общественных предметов. Производство создает предмет для субъекта. Одна и та же вещь (объект-реальность) может быть представлена в деятельности различным предметным образом. К одной и той же вещи как объекту познания могут быть отнесены различные

предметы исследования.

На основе форм чувственности, посредством категорий рассудка человек формирует предмет своей деятельности. Отметим, что формирование предмета познания невозможно без определенного знания. При этом, предмет есть не что иное, как вещь, функционирующая в сфере человеческой деятельности: вещи, существующие "сами по себе" становятся вещами "для нас". В понятии "явление" отображается процессуальный характер бытия вещей, обнаруживающих ту или иную сторону своей сущности в отношениях с другими вещами. Познание предмета, т.е. определенного аспекта бытия вещи, ее свойств, отношений, есть всегда познание явления или, что то же самое, - познание сущности вещи (объекта) на уровне ее явления. На эмпирическом уровне познания предмет выступает в качестве **эмпирического объекта**, а на теоретическом - в качестве **идеального, теоретического объекта**. Предметом познания может быть как отдельное свойство, отношение вещи, так и подсистема ее свойств и отношений. Это обуславливает необходимость выявления и учета условий познания.

Процедуры эмпирического уровня познания направлены на то, чтобы возможно более полно и точно **описать предмет** посредством выявления характеризующих его признаков. При этом, под описанием понимается фиксация результатов опыта, выраженная в понятиях данной конкретной науки. Процесс описания есть и аналитическая, и синтетическая процедура. Но в полной мере указанные операции осуществляются на теоретическом уровне познания, где предмет как эмпирический объект реконструируется в абстрактный, теоретический объект. Если в качестве предмета выступает отдельное свойство или отношение вещи, то процесс мыслительной реконструкции эмпирического объекта познания в абстрактный, т.е. процесс образования понятия об этом отдельном свойстве или отношении, оказывается возможным лишь при условии, что свойство (отношение) представлено в виде некоторой системы признаков, характеризующих его [Л. 246, с. 225]. То есть, иными словами, процесс образования понятия о свойстве или отношении возможен только при условии, что это свойство или отношение должно быть представлено в виде идеальной вещи. В данном случае, свойство или отношение должны быть представлены как абстрактный теоретический объект. Результатом познания вещи как предмета является познание ее отдельных свойств и отношений: особенное бытие вещи, резюмируемое в образовании абстрактно-общих понятий. На этом этапе познания вещи мы получаем возможность выделить ее посредством описания в абстрактно общих терминах. Последующая задача мышления заключается в том, чтобы синтезировать все многообразие предметных определений объекта и, таким образом, объяснить его, создав теперь уже понятие о самой вещи.

#### **2.4.6. ВЕЩЬ КАК ПРЕДМЕТ И ПРОБЛЕМА ЭМПИРИЧЕСКОГО БАЗИСА НАУЧНОГО ПОЗНАНИЯ**

Материальные вещи даны человеку на основе (и в форме) его деятельности. Это определяет тот факт, что человек в процессе познания создает свой предмет, возможную схему вещи. Человек не обладает способностью к непосредственному "умозрению" объективной реальности. Поэтому результаты чувственно-практической деятельности человека выступают тем эмпирическим базисом (данные опыта), из которого исходит и на котором возводится все здание научного зна-

ния о мире. Эмпирическая реальность есть, следовательно, объективная реальность, как она дана человеку в процессе его чувственно-практической деятельности.

Эмпирическая реальность не есть то же самое, что "объективная реальность" и не тождественна нашему представлению о ней как эмпирическом объекте, поскольку последний есть уже результат рациональной обработки мышлением данных опыта. Объективная реальность не является непосредственным эмпирическим базисом теоретического знания. Таким базисом являются результаты чувственно-практического взаимодействия человека с природой. Для того, чтобы верно интерпретировать результаты эксперимента, необходимо обязательно выявлять и учитывать роль условий на эмпирическом уровне. Недостаточно знать абстрактный закон явления, надо знать еще конкретные условия, в которых действует этот закон, и которые определяют (обуславливают) именно такое, а не иное его проявление.

Под физической реальностью понимается процесс взаимодействия физических объектов и условий познания, а также его результат на уровне эксперимента, т.е. определенная форма данности объективной реальности, которая фиксируется, представляется, модулируется различным образом на различных уровнях познавательного процесса [Л. 246, с. 228]. Кроме того, необходимо учитывать факт гносеологической активности субъекта, находящей свое выражение в представлениях о физической реальности и формирующей ее как эмпирический объект. Физическая реальность есть не что иное, как мир эмпирических физических объектов. Отождествление физической реальности с концептуальной есть отождествление объективного бытия предмета познания со знанием о нем. Выделение материального физического мира связано с выделением физических форм движения материи. Понятия "материя" и "материальный физический мир" отнюдь не тождественные понятия. "Материальный физический мир" - это мир естественных физических объект-реальностей, физический мир как таковой, т.е. "сам по себе". Теоретическим представлением материального физического мира является физическая картина мира. Физическая же реальность - это мир физических явлений (предметов). Это то, в какой форме данности материальный физический мир (мир физических вещей) предстает перед человеком как эмпирический объект познания. Теоретическим представлением физической реальности является физическая теория.

#### 2.4.7. ВЕЩЬ КАК ОБЪЕКТ ПОЗНАНИЯ

Как предметы вещи несут на себе печать человеческой, и в этом смысле - субъективной, деятельности. Предметное выделение вещи опосредованно всей общественно-исторической практикой человека. Предмет как объект познания, т.е. эмпирический, теоретический объект, суть своеобразный "конструкт", представляющий собой единство субъективного и объективного. Предмет объективен, поскольку представляет собой определенный аспект - "объект реальности" (вещи) и субъективен как продукт человеческой деятельности, а также в том смысле, что знание, на основе которого выделяется предмет, неполно, абстрактно, а, следовательно, и относительно истинно.

**Всякий предмет есть единство объективного и субъективного, именно как объект познания** [Л. 246, с. 230].

Единственной материальной вещью, в которой реально осуществлено единство объективного и субъективного, является, по-видимому, только сам человек. Человек, согласно К. Марксу, своей жизнедеятельностью всю природу превращает в свое неорганическое тело [Л. 182]. Человек представляет тотальность всей

природы, мира в целом. Стало быть, процесс самосознания человеком собственной природы, т.е. сам человек как объект познания, есть, в сущности, процесс познания мира в целом. Это является свидетельством правомерности рассмотрения мира в целом как объекта познания. Ведь мир в целом есть теоретический объект философского исследования. Отношение объективного и субъективного в знании о предмете находит свое выражение в процессе познания как противоречие объективной (абсолютной) и относительной истин. Процесс разрешения этого противоречия и есть процесс преодоления субъективного в знании о вещах. Подлинным объектом в прямом и наиболее полном смысле этого слова материальная вещь выступает лишь как объект всего познания в целом. Познавательная реконструкция вещи (объект-реальности) как объекта познания есть процедура целиком теоретическая. Можно сказать, что вещь в процессе познания выступает как эмпирический объект, как абстрактный объект и, наконец, как теоретический объект. Таким образом, познание вещи на теоретическом уровне предполагает два уровня движения синтезирующей мысли: по предмету и по объекту.

Познание вещи на эмпирическом и на теоретическом уровнях позволяет выявить существенные свойства и отношения вещей и создать понятие о них. Предмет рассматривается уже как некоторая система свойств и отношений вещи, знание которой резюмируется в понятии предмета, т.е. предмет как вещь в особенной форме своего бытия.

В развитой форме это знание представляет собой некоторую систему знания о предмете, т.е. его научную теорию. Это линия движения познания "по предмету" (явлению). При этом, под теорией, в данном случае, понимается система понятий и принципов, отражающих существенные свойства и связи вещи. По онтологическому статусу различные предметные представления вещи в деятельности суть различные существования одной и той же вещи, т.е. различные проявления ее сущности. Выявление в процессе познания сущностного единства различных явлений есть не что иное, как представление их как различных предметных определений одной и той же вещи, и дальнейшее познание вещи как объекта заключается в теоретическом синтезе этих определений.

Результатом движения синтезирующего мышления "по объекту" является теперь уже понятие о вещи. Границы познания вещи как объекта совпадают с границей ее отражения в понятии. На различных этапах постижения вещи понятие о ней принимает различные формы. На эмпирическом уровне - это определение ее через род и видовое отличие. Теоретический уровень позволяет создать понятие о вещи как об абстрактном объекте в виде **идеи** вещи, в которой синтезируется знание о сущности вещи. Однако в идее вещь отображена абстрактно, вне ее отношений с другими вещами, и в этом смысле, она есть высший синтез теоретического абстрактного знания о вещи.

Следующей формой является **теория**, возвращающая нас к отображению вещи как единства многообразного. Идея становится конкретным знанием в ходе своего концептуального развертывания, т.е. лишь в рамках определенной теоретической системы. Каждый новый цикл познания вещи характеризуется более высоким уровнем абстракции и обобщения (синтеза) знания, выражающегося в названных формах развивающегося понятия о вещи. Процесс познания действительного бытия вещей, в конечном счете, оказывается процессом познания всего действительного мира. Развитие понятия о вещи не заканчивается теорией, оно продолжается в рамках более общих и глубоких теоретических построений, отражающих бытие уже не отдельных вещей, а мира в целом. Только в рамках научной картины мира можно получить наиболее полный и глубокий ответ на вопрос, почему эта вещь возможна и в чем состоит

действительная всеобщность любых ее форм. В ответах на эти вопросы заключается объяснение бытия вещи, познание ее как объекта. Знание одной сущности недостаточно, чтобы объяснить вещь во всей конкретности ее действительного бытия. Необходимо знать и всеобщие формы ее проявления, т.е. законы функционирования вещи, "как" существует вещь. Именно единство знания "как" и "почему" вещи позволяет объяснить и, таким образом, "овладеть" вещью. Объяснить вещь - значит понять ее как тотальность, как единство многообразного. Идея вещи, хотя и отображает ее сущность, еще не объясняет ее бытие, но содержит в себе (в возможности) это объяснение. **Объяснить вещь, значит, образовать понятие о вещи в форме развернутой системы знаний о ней**, только тогда постигнутый в понятиях мир как таковой предстает как действительный мир.

## **2.5. МЕТОДОЛОГИЧЕСКИЕ ФУНКЦИИ КАТЕГОРИЙ ВЕЩЬ, СВОЙСТВО, ОТНОШЕНИЕ И СИНТЕЗ НАУЧНОГО ЗНАНИЯ О МИРЕ**

Развивающееся знание о мире характеризуется все более и более усиливающейся тенденцией к единству. Давно замечено, что существует определенная связь между единством знания об объекте и его истинностью, адекватностью объекту. Синтез научного знания о мире действительно связан с деятельностью субъекта. Это предполагает рассмотрение объективных оснований именно такого характера процесса познания мира. Ошибочность идеалистического подхода в решении данного вопроса связана с тем, что субъект рассматривается только как субъект познания, а не как субъект практической в своей основе деятельности [Л. 246]. Именно выявление деятельной природы человека позволяет понять его как особого рода материальную вещь, существующую наряду с другими материальными вещами.

Материальные **вещи по самой своей природе суть системы**. Стало быть, системно организованным является как объект, так и субъект деятельности. Как отношение субъекта к объекту системной оказывается и сама деятельность. В самом деле, в силу системной организации человека как природной вещи системным является и любое многообразие его потребностей, а, следовательно, и процесс их удовлетворения, т.е. деятельность. При этом, в ходе трудовой деятельности формируется и социальный аспект системной организации человека. Сказанное позволяет понять, почему в процессе своей деятельности субъект вычленяет в объекте целостный (иначе говоря, системный) фрагмент материальной действительности (предмет) и отражает его системным же образом. Иными словами, адекватное знание о системном объекте не может быть само несистемным. Вот почему "истинной формой, в которой существует истина, может быть, - как подчеркивал Гегель, - лишь научная система ее" [Л. 68, 69, 246].

**Понятия вещи, свойства, отношения представляют**, таким образом, **основные определения движущейся материи**, они суть те самые **узловые точки сети понятий**, накладывая которую на объективную реальность, мы получаем возможность отобразить последнюю во всей противоречивой действительности ее организованного бытия [Л. 246, с. 236]. К методологическим функциям понятий вещи, свойства, отношения в познании следует отнести следующее:

- функцию, которая связана с осуществлением концептуального единства знания, ибо все знание о мире так или иначе всегда есть знание о свойствах и отношениях вещей, о самих вещах;

- регулятивную функцию, определяющую направление процесса познания;
- организующую, синтезирующую функцию.

Все эти функции необходимо взаимосвязаны. Рассматривая объективную диалектику бытия вещи, мы показали, что всякая вещь есть существенное единство в существенном множестве. **Как целое вещь существует и обнаруживает себя только через многообразие своих существенных свойств.** Эта особенность противоречивой природы бытия вещи (вещь как единство многообразного) определяет тот факт, что процесс познания вещи начинается с выявления ее свойств и отношений, затем идет в направлении познания ее существенного единства и, наконец, к познанию ее как тотальности, как единства многообразного. В целом это движение познающей мысли может быть охарактеризовано как процесс восхождения от абстрактного к конкретному. Развитие знания о вещи есть движение (развитие) понятия о ней как объекте. Системный характер реального бытия вещи означает, что знание, адекватно отражающее его, также должно быть системным.

**Понятие о вещи есть всегда система знаний**, и поскольку в понятии о вещи фиксируются существенные моменты ее бытия, постольку степень и характер системной организации знания о ней характеризуют и степень проникновения в суть понимания ее действительного бытия. Формой синтеза знания о вещи как эмпирическом объекте будет определение. **Определение** отражает вещь как некое, пока еще только внешнее, единство существенных свойств, принадлежащих ей как носителю (субстрату). Знание вещи как существенного множества свойств и отношений, характеризующих вещь как предмет и принадлежащих ей как объекту, составляет основное содержание развитого определения. Таким образом, знание, фиксируемое в определении, всегда относительно, так как получено при данных, конкретных условиях познания. Это означает, что знание о вещи, синтезируемое на эмпирическом уровне познания в форме определения, изменяется, имея тенденцию к развитию в направлении своей наиболее развитой и полной (в рамках определения) формы на эмпирическом уровне.

Теоретический уровень познания дает две основные формы синтеза знания о вещи: идею и теорию. Идея есть результат теоретического синтеза существенных определений вещи как эмпирического объекта. Идея выражает сущность вещи и в этом смысле есть знание последней как абстрактного объекта. Особенность гносеологического статуса идеи заключается в том, что она подобно отражаемой ею сущности, заключающей в себе все многообразие своих возможных проявлений, содержит не только знание об уже известных свойствах и отношениях вещи, но и таит в себе новое знание, которого не было в исходных для нее (до синтеза) определениях вещи как эмпирического объекта. Идея есть исходный пункт развития теории и, будучи положена как основание, из которого развивается все богатство содержания теории, выступает уже в роли принципа. Поскольку идея выражает не всю сущность вещи, а только какой-то ее определенный аспект, сторону, то в основание теории объекта может быть положен ряд принципов - система абстрактных объектов. Развертывание идеи в теорию позволяет отобразить вещь в действительности ее конкретного бытия. Вот почему **теория является высшей формой синтеза знания о вещи.**

Функционирование теории заключается, с одной стороны, в объяснении уже наличного эмпирического материала, с другой стороны, в предсказании новых явлений. Теория в процессе своего функционирования осуществляет единство форм экстенсивного и интенсивного знания о вещи. Теория есть система знаний не вещи в целом (вещь как объект-реальность), а предмета. Высший синтез знания

о вещи (и как его результат - понятие о вещи) может быть достигнут лишь в рамках теоретической системы знания, отображающей мир в целом. **По сути дела, объект научного познания всегда один и тот же - мир в целом (движущаяся материя), но исследуется он с разных сторон и на различных уровнях абстракции.** Научная картина мира представляет собой высшую форму синтеза знания о мире. Философская картина мира, опосредствуясь конкретно-научным знанием, развертывается в научную картину мира, представляющую собой уже знание конкретного бытия доступной человеку в его деятельности области мира в целом [Л. 246, с. 239]. Научная картина мира не есть система только методологических принципов, так как она включает в себя, кроме того, наиболее общие и фундаментальные эмпирические определения объекта, т.е. теоретическое представление всех структурных уровней материи.

Научная картина мира функционирует как на эмпирическом, так и теоретическом уровнях познания, представляя собой высший результат и необходимую предпосылку научного постижения его действительного бытия. Имеющиеся естественно - научные картины мира не могут пока претендовать на эту роль, поскольку не включают в себя науки о человеке [Л. 246]. Процесс формирования научной картины мира будет развертываться по мере того, как естествознание включит в себя науку о человеке в такой же мере, в какой науки о человеке включают в себя естествознание. Это будет одна наука [Л. 182, 246]. И ничего удивительного в этом нет, ибо сама история является действительной частью истории природы, становления природы человеком. Непрерывно расширяющейся и углубляющийся синтез знаний о вещах позволяет постепенно, шаг за шагом давать все более объективную картину их действительного бытия. Этот процесс постоянно стимулируется и детерминируется практической деятельностью.

Практика - исходный и конечный пункт этого движения, и в тех границах, в каких мы на практике имеем дело с вещами, представления о вещи и об ее свойствах совпадают с существующей вне нас действительностью, т.е. являются объективно истинными. Мир вещей неисчерпаем. Такова особенность его бытия, вытекающая из того факта, что он (мир) - суть (есть) движущаяся материя, т.е. совершающее процесс противоречие.

**Задача науки** заключается в том, чтобы видимое, выступающее на поверхности явлений движение свести к действительному внутреннему движению. Кроме того, **знание сущности, абстрактного закона еще не дает нам картины действительного бытия вещей.** Поэтому задача науки состоит именно в том, чтобы **объяснить, как проявляется закон.** Объективная неисчерпаемость свойств и отношений мира вещей означает, что практически процесс познания мира бесконечен. И как бы бесконечно близко мы ни приближались на этом пути к истинному познанию мира, оно никогда не будет исчерпывающим. Вот почему перед лицом неизмеримо богатой, постоянно обновляющейся природы, человек, как бы ни был велик прогресс в области научного познания, остается всегда "удивляющимся ребенком и должен быть постоянно готов к новым неожиданностям" [Л. 246, с. 240]. Представляется целесообразным, в заключение, сделать следующие **выводы.** Рассмотрение объективной диалектики материального бытия выявляет тот фундаментальный факт, что:

**1) реальный мир организован как система вещей.**

Всякая реальная вещь по самой сути своей есть совершающее процесс противоречие и, в этом смысле,

**2) вещь есть отношение.**

- 3) **Вещь существует как система свойств и отношений, которые ее образуют, и посредством которых она обнаруживает свое существование.**
- 4) **Вещь есть ничто вне и помимо свойств и отношений, точно так же, как свойства и отношения существуют лишь как свойства и отношения вещей.**
- 5) **Вещь есть система, и это объясняет, почему ее нельзя свести к одним только свойствам или отношениям.**

Это объясняет также, почему вещь обладает большей, чем свойства и отношения, самостоятельностью, обуславливающей ее ведущее положение в системе понятий вещь - свойство - отношение. Выявление категориального статуса понятий вещь, свойства, отношение их методологических функций позволяет ответить на вопрос о месте этих всеобщих понятий в системе категорий диалектики. Действительно, какой бы уровень организации материального мира мы не взяли,

**б) объектом познания всегда является вещь.**

Как система вещей мир есть единство многообразного и, следовательно, сам в известном смысле представляет собой вещь. Это объясняет, почему абстрактно-всеобщие определения всякой вещи, выраженные в форме логических категорий и отображающие диалектику ее объективного бытия, оказываются вместе с тем и определениями материального мира в целом. Диалектика как учение о наиболее общих законах развития природы, общества и мышления оказывается, таким образом, теорией бытия всякой материальной вещи (как природной, так и духовной) [Л. 246]. Это и определяет:

**7) центральное положение подсистемы категорий вещь - свойство - отношение в системе категорий диалектики,**

а вместе с этим и значение (и необходимость) их дальнейшего исследования для построения развернутой системы научного знания о мире. Выявив основные, центральные категории (вещь, свойство, отношение) научного описания любой предметной области, вплоть до описания всей научной картины окружающего нас мира, перейдем к исследованию возможностей структур представления данных традиционных моделей баз данных.

### 3. АНАЛИЗ И ОБОБЩЕНИЕ СТРУКТУР ПРЕДСТАВЛЕНИЯ ДАННЫХ ТРАДИЦИОННЫХ МОДЕЛЕЙ ДАННЫХ

#### 3.1. ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ ОСНОВНЫХ СТРУКТУР ПРЕДСТАВЛЕНИЯ ДАННЫХ ТРАДИЦИОННЫХ МОДЕЛЕЙ ДАННЫХ

Современные АСОИ должны работать с любыми традиционными базами данных (БД), независимо от их моделей данных. Ключевой проблемой при этом является разработка таких структур представления данных (СПД) для баз данных, которые были бы совместимы, как минимум, со всеми основными СПД традиционных моделей данных (МДн). Проблема совместимости СПД различных МДн решается либо на концептуальном уровне, либо путем построения специальных преобразователей из СПД одной МДн в СПД другой [Л. 27, 28, 51, 70, 84, 86-88, 91, 100-104, 107, 111, 120-124, 139-142, 152, 157, 175, 183-185, 190, 196, 200-206, 213, 216, 226-227, 244, 247, 249, 257, 263, 269-271, 289, 293, 314-319, 326-327, 335, 345-356, 372-383, 387-390, 396-403, 461-464, 481-501], например - из реляционных в сетевые.

В результате обзора литературы [Л. 13, 15, 19, 23, 24, 36, 50, 63, 71, 82, 91, 110-118, 136, 144, 155, 170, 176, 178, 192, 195, 199, 220, 231, 250, 284, 287, 293, 296, 300, 305, 321-323, 339, 359-368, 407-413] и проведенных исследований разработаны формализованные описания на едином языке теории множеств всех основных СПД традиционных моделей данных, а также проведен их анализ. Сравнение и обобщение этих основных СПД позволило предложить новый подход к разработке структур унифицированного представления данных. Прежде всего, рассмотрим формализованное описание основных СПД наиболее известных, "традиционных" моделей данных, а затем перейдем к их анализу, сравнению и обобщению.

##### 3.1.1. ОСНОВНЫЕ СПД ТРАДИЦИОННЫХ МОДЕЛЕЙ ДАННЫХ

Итак, в данной работе описаны на едином языке и проанализированы СПД следующих традиционных моделей данных: реляционных; сетевых; иерархических; бинарной; модели данных "сущность-связь"; семантических сетей; объектно-ориентированных; символьных. Кроме перечисленных моделей данных проанализируем и некоторые другие, наиболее известные СПД, либо покажем, что они могут быть сведены, к указанному выше, СПД. Подчеркнем, что под моделью данных понимаются совокупность трех составляющих [Л. 19, 87, 121, 123, 183, 190, 220, 284, 293, 305, 316, 319]: структуры данных, ограничения целостности и операции над данными. Для любой модели данных системообразующую, ключевую роль играют именно СПД, которые мы и рассмотрим. Существует два простейших способа представления данных: таблицы и графы. Необходимо подчеркнуть, что "... описательные возможности табличного и графового представления одинаковы. На основе табличного представления легко получить графовое: достаточно специфицировать путь, соединяющий атрибуты, образующие таблицу" [Л. 316].

Следовательно, исходя из вышеизложенного, представляется целесообразным выделить **графо-табличные** СПД, которые и называются в дальнейшем: **традиционными**. Отметим, что анализ традиционных СПД показал, что наиболее широкое распространение получила форма графа с вершинами-таблицами. Рассмотрим СПД традиционных моделей данных более подробно.

### 3.1.2. ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ РЕЛЯЦИОННЫХ СПД

Реляционная модель данных - это "множество нормализованных отношений (таблиц), к которым применимы операции реляционной алгебры" [Л. 226]. "Единственным средством структуризации данных в реляционной модели является отношение" [Л. 316]. "Одним из основных преимуществ реляционной модели является ее однородность. Все данные рассматриваются как хранимые в таблицах, в которых каждая строка имеет один и тот же формат" [Л. 190]. Интенционал реляционной базы данных задается реляционной схемой, состоящей из одной или нескольких схем отношений. Схема отношения задается именем отношения и именами соответствующих доменов. Выделяют следующие реляционные СПД: **имена отношений, имена атрибутов, значения**. Рассмотрим формализованное описание реляционных СПД. Пусть  $A$  - это множество имен отношений:

$$A = \{a_1, a_2, \dots, a_i, \dots, a_I\}, \quad (3.1.1)$$

где  $i = \overline{1, I}$ . Тогда для

$$\forall a_i \in A \quad \exists V_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ_i}\}, \quad (3.1.2)$$

где  $i = \overline{1, I}$ ;  $j_i = \overline{1, J_i}$  и  $V_i$  - это множество имен атрибутов отношения  $a_i$ . Следовательно, для

$$\forall b_{ij} \in V_i \quad \exists C_{ij} = \{c_{ij1}, c_{ij2}, \dots, c_{ijk_{ij}}, \dots, c_{ijK_{ij}}\}, \quad (3.1.3)$$

где  $i = \overline{1, I}$ ;  $j_i = \overline{1, J_i}$ ;  $k_{ij} = \overline{1, K_{ij}}$  и  $C_{ij}$  - это множество значений атрибута  $b_{ij}$  отношения  $a_i$ .

Подчеркнем, что от традиционного описания реляционных СПД формулы 3.1.1-3.1.3 отличаются большей структурированностью индексов, так как для каждого отношения определяется свое множество атрибутов (формула 3.1.2), а для каждого элемента множества имен атрибутов (для каждого атрибута) определяется свое индивидуальное множество значений (формула 3.1.3). В случае необходимости, все элементы однотипных множеств, т.е. множества имен атрибутов или множества значений атрибута отношения, могут быть соответственно объединены в общее множество имен атрибутов или множество значений. Однако, для дальнейшего анализа и сравнения СПД целесообразно выделять различные множества, как это и сделано в формулах 3.1.2-3.1.3.

### 3.1.3. ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ СЕТЕВЫХ СПД

Сетевая структура данных - это "структура данных, представляющая собой ориентированный граф, в любой узел которого может входить более одной связи" [Л. 226]. "Сетевые модели данных базируются на табличных и графовых представлениях: вершинам графа обычно сопоставляют некоторые типы сущности, которые представляются таблицами, а дугам - типы связей" [Л. 316]. Двумя основными категориями структур данных в сетевой модели являются записи и связи. Типы записей используются для табличного представления типов сущностей. Связи используются для представления типов связей. С помощью связей специфицируются соединения между типами записей. "При реализации модели в различных конкретных СУБД можно применять различные способы представления в памяти системы данных, описывающих связи между сущностями" [Л. 319]. Таким образом, можно выделить следующие основные сетевые СПД: **типы записей, элементы данных типов записей, реализации типов записей, типы наборов-**

**связей, реализации наборов-связей.** Рассмотрим формализованное описание сетевых СПД.

**Замечание.** В связи с ограниченностью количества символов в латинском алфавите и для подчеркивания аналогичных взаимосвязей элементов СПД различных моделей данных, будем использовать в разных параграфах одинаковые символы для описания исследуемых СПД, но с различной их интерпретацией.

С учетом этого замечания, пусть  $A$  - множество имен типов записей:

$$A = \{a_1, a_2, \dots, a_i, \dots, a_I\}, \quad (3.1.4)$$

где  $i = \overline{1, I}$ . Тогда для

$$\forall a_i \in A \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ}\}, \quad (3.1.5)$$

где  $i = \overline{1, I}$ ;  $j = \overline{1, J_i}$  и  $B_i$  - это множество имен элементов данных типа записи  $a_i$ .

Тогда, в свою очередь, для

$$\forall b_{ij} \in B_i \quad \exists C_{ij} = \{c_{ij1}, c_{ij2}, \dots, c_{ijk_{ij}}, \dots, c_{ijK_{ij}}\}, \quad (3.1.6)$$

где  $i = \overline{1, I}$ ;  $j = \overline{1, J_i}$ ;  $k_{ij} = \overline{1, K_{ij}}$  и  $C_{ij}$  - это множество значений элемента данных

$b_{ij}$  типа записи  $a_i$ . Пусть  $Z$  - множество имен типов наборов-связей:

$$Z = \{z_1, z_2, \dots, z_l, \dots, z_L\}, \quad (3.1.7)$$

где  $l = \overline{1, L}$ . Тогда, для любого элемента множества  $Z$  существуют два множества:

$$\forall z_l \in Z \quad \exists Y_l = \{y_{l1}, y_{l2}, \dots, y_{lm_l}, \dots, y_{lM_l}\}, \quad (3.1.8)$$

$$\forall z_l \in Z \quad \exists X_l = \{x_{l1}, x_{l2}, \dots, x_{ln_l}, \dots, x_{lN_l}\}, \quad (3.1.9)$$

где  $l = \overline{1, L}$ ;  $m_l = \overline{1, M_l}$ ;  $n_l = \overline{1, N_l}$  и  $Y_l$  - это множество реализаций записей-владельцев набора-связи типа  $z_l$  и  $X_l$  - это множество реализаций записей-членов набора-связи типа  $z_l$ . При этом,

$$\forall z_l \in Z \quad \exists \{ \langle y_{lm_l}, x_{ln_l} \rangle \mid y_{lm_l} \in Y_l, x_{ln_l} \in X_l \}. \quad (3.1.10)$$

### 3.1.4. ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ ИЕРАРХИЧЕСКИХ СПД

Иерархическая модель данных - это "модель данных, в основе которой лежит граф типа "дерево". Вершине дерева соответствует тип записи, дуге - отношение между двумя типами записей" [Л. 226]. "Структурная диаграмма иерархической базы должна быть упорядоченным деревом" [Л. 316]. В дереве определения, вершина соответствует типу сущности и называется типом записи. Тип записи состоит из одной или более единиц данных, которые могут быть определены на простом домене. Дуга дерева определения, соответствующая функциональному типу связи, называется связью "исходный - порожденный" и не помечается, так как между двумя типами записей может быть не более одной такой связи. Среди вершин выделяется одна, называемая типом корневой записи. Выделим следующие иерархические СПД: **категория типа записи (корневая, порожденная), типы записей, элементы данных типов записей, реализации типов записей, реализации наборов-связей.** Рассмотрим формализованное описание иерархических СПД. Пусть  $A$  - множество имен категорий типов записей (реально существуют только две категории: "корневая" и "порожденная"):

$$A = \{a_1, a_2, \dots, a_i, \dots, a_I\}, \quad (3.1.11)$$

где  $i = \overline{1, I}$ . Далее, для

$$\forall a_i \in A \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ_i}\}, \quad (3.1.12)$$

где  $i = \overline{1, I}$ ;  $j_i = \overline{1, J_i}$  и  $B_i$  - это множество имен типов записей категории типа записи  $a_i$ . Тогда, в свою очередь, для

$$\forall b_{ij} \in B_i \quad \exists C_{ij} = \{c_{ij1}, c_{ij2}, \dots, c_{ijk_{ij}}, \dots, c_{ijK_{ij}}\}, \quad (3.1.13)$$

где  $i = \overline{1, I}$ ;  $j_i = \overline{1, J_i}$ ;  $k_{ij} = \overline{1, K_{ij}}$  и  $C_{ij}$  - это множество имен элементов данных типа записи  $b_{ij}$  категории типа записи  $a_i$ . Далее, для

$$\forall c_{ijk_{ij}} \in C_{ij} \quad \exists D_{ijk_{ij}} = \{d_{ijk_{ij}1}, d_{ijk_{ij}2}, \dots, d_{ijk_{ij}l_{ijk_{ij}}}, \dots, d_{ijk_{ij}L_{ijk_{ij}}}\} \quad (3.1.14)$$

где  $i = \overline{1, I}$ ;  $j_i = \overline{1, J_i}$ ;  $k_{ij} = \overline{1, K_{ij}}$ ;  $l_{ijk_{ij}} = \overline{1, L_{ijk_{ij}}}$  и  $D_{ijk_{ij}}$  - это множество реализаций элемента данных  $c_{ijk_{ij}}$  типа записи  $b_{ij}$  категории типа записи  $a_i$ . Все связи

иерархической модели данных являются связями функционального типа и не помечаются. Следовательно, необходимо хранить только реализации наборов-связей. Тогда, пусть

$$Z = \{z_1, z_2, \dots, z_m, \dots, z_M\}, \quad (3.1.15)$$

где  $m = \overline{1, M}$ ;  $Z$  - это множество реализаций наборов-связей, соответствующих типу "корневой". Также, пусть

$$Y = \{y_1, y_2, \dots, y_n, \dots, y_N\}, \quad (3.1.16)$$

где  $n = \overline{1, N}$ ;  $Y$  - это множество реализаций наборов-связей, соответствующих типу "порожденный". Тогда, все наборы-связи описываются следующим множеством двоек:

$$\exists \{ \langle z_m, y_n \rangle \mid z_m \in Z, y_n \in Y \}. \quad (3.1.17)$$

### 3.1.5. ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ СПД МОДЕЛИ "СУЩНОСТЬ-СВЯЗЬ"

Модель Чена ("сущность-связь", ER-модель)- это "семантическая реляционная модель данных, в основе которой лежит деление реального мира на отдельные различимые сущности, находящиеся в определенных связях друг с другом..." [Л. 226]. С другой стороны, "такие модели имеют много общего с иерархическими и сетевыми моделями данных; более того, в силу своей ориентации на процесс проектирования они могут рассматриваться как обобщение и развитие иерархических и сетевых моделей" [Л. 316]. Базовыми структурами в ER-модели являются типы сущностей и типы связей. **Тип сущности** носит название множества сущностей и представляет общую структуру сущности. **Тип связи** называется множеством связей и представляет общую структуру связей между множествами сущностей. Тип отображения, соответствующий множеству связей, явным образом специфицируется.

Множество сущностей, участвующее во множестве связей играет некоторую роль. Домен в ER-модели называется множеством значений. "Атрибутом называется отображение между множеством сущностей или связей и множеством значений в контексте множества сущностей или связей" [Л. 316]. Некоторые атрибуты могут быть многозначными, но в контексте данной работы, ограничимся тем, что в таких случаях будем рассматривать не один многозначный атрибут, а несколько подобных, схожих или одинаковых однозначных атрибутов. Таким образом, можно выделить следующие СПД модели "сущность-связь":

- 1) классы сущностей,
- 2) множества сущностей,
- 3) классы (размерности) связей,
- 4) множества связей,
- 5) роли сущностей в связях,
- 6) множество атрибутов,
- 7) множество значений.

Рассмотрим формализованное описание СПД модели "сущность- связь". Пусть  $A$  - множество имен классов сущностей:

$$A = \{a_1, a_2, \dots, a_i, \dots, a_I\}, \quad (3.1.18)$$

где  $i = \overline{1, I}$ . Тогда, для

$$\forall a_i \in A \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ}\}, \quad (3.1.19)$$

где  $i = \overline{1, I}$ ;  $j = \overline{1, J_i}$  и  $B_i$  - это множество имен сущностей класса  $a_i$ . Далее, пусть

$$\exists C = \{c_1, c_2, \dots, c_k, \dots, c_K\}, \quad (3.1.20)$$

где  $k = \overline{1, K}$ ;  $C$  - множество имен атрибутов. Кроме того, пусть

$$\exists D = \{d_1, d_2, \dots, d_l, \dots, d_L\}, \quad (3.1.21)$$

где  $l = \overline{1, L}$ ;  $D$  - множество значений. Далее, пусть

$$\exists Z = \{z_1, z_2, \dots, z_m, \dots, z_M\}, \quad (3.1.22)$$

где  $m = \overline{1, M}$ ;  $Z$  - множество имен классов связей. Тогда,

$$\forall z_m \in Z \quad \exists Y_m = \{y_{m1}, y_{m2}, \dots, y_{mn_m}, \dots, y_{mN_m}\}, \quad (3.1.23)$$

где  $m = \overline{1, M}$ ;  $n_m = \overline{1, N_m}$  и  $Y_m$  - множество имен связей класса  $z_m$ .

Рассмотрим описание сущностей, для

$$\forall b_{ij} \in B_i \quad \exists C_{ij} = \{c_{ij1}, c_{ij2}, \dots, c_{ijp_{ij}}, \dots, c_{ijP_{ij}}\}, \quad (3.1.24)$$

где  $i = \overline{1, I}$ ;  $j = \overline{1, J_i}$ ;  $p_{ij} = \overline{1, P_{ij}}$ . Далее, для

$$\forall c_{ijp_{ij}} \in C_{ij} \quad \exists D_{ijp_{ij}} = \{d_{ijp_{ij}1}, d_{ijp_{ij}2}, \dots, d_{ijp_{ij}r_{ijp_{ij}}}, \dots, d_{ijp_{ij}R_{ijp_{ij}}}\} \quad (3.1.25)$$

где  $i = \overline{1, I}$ ;  $j = \overline{1, J_i}$ ;  $p_{ij} = \overline{1, P_{ij}}$ ;  $r_{ijp_{ij}} = \overline{1, R_{ijp_{ij}}}$ .

В модели данных "сущность-связь", каждая связь может описываться некоторым набором атрибутов, которые могут иметь определенные множества значений. Другими словами, для

$$\forall y_{mn_m} \in Y_m \quad \exists C_{mn_m} = \{c_{mn_m1}, c_{mn_m2}, \dots, c_{mn_ms_{mn_m}}, \dots, c_{mn_mS_{mn_m}}\}, \quad (3.1.26)$$

где  $m = \overline{1, M}$ ;  $n_m = \overline{1, N_m}$ ;  $s_{mn_m} = \overline{1, S_{mn_m}}$ . Далее, для

$$\forall c_{mn_ms_{mn_m}} \in C_{mn_m} \quad \exists D_{mn_ms_{mn_m}} = \{d_{mn_ms_{mn_m}1}, d_{mn_ms_{mn_m}2}, \dots, d_{mn_ms_{mn_m}t_{mn_ms_{mn_m}}}, \dots, d_{mn_ms_{mn_m}T_{mn_ms_{mn_m}}}\} \quad (3.1.27)$$

где  $m = \overline{1, M}$ ;  $n_m = \overline{1, N_m}$ ;  $s_{mn_m} = \overline{1, S_{mn_m}}$ ;  $t_{mn_ms_{mn_m}} = \overline{1, T_{mn_ms_{mn_m}}}$ .

Пусть,  $E$  - это множество имен ролей множеств сущностей в связях,

$$E = \{e_1, e_2, \dots, e_u, \dots, e_U\}, \quad (3.1.28)$$

где  $u = \overline{1, U}$ . Введем следующее обозначение:

$$\mathbf{B} = \bigcup_{i=1}^I \mathbf{B}_i . \quad (3.1.29)$$

Итак, каждой связи соответствует некоторый набор доек, описывающих некоторые сущности и их роли, т.е.:

$$\forall y_{mn_m} \in Y_m \quad \exists \{ \langle b_{mn_m 1}, e_{mn_m 1} \rangle, \langle b_{mn_m 2}, e_{mn_m 2} \rangle, \dots, \langle b_{mn_m v_{mn_m}}, e_{mn_m v_{mn_m}} \rangle, \dots, \langle b_{mn_m v_{mn_m}}, e_{mn_m v_{mn_m}} \rangle \}, \quad (3.1.30)$$

где  $b_{mn_m v_{mn_m}} \in B$ ;  $e_{mn_m v_{mn_m}} \in E$ ;  $m = \overline{1, M}$ ;  $n_m = \overline{1, N_m}$ ;  $v_{mn_m} = \overline{1, V_{mn_m}}$ .

### 3.1.6. ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ БИНАРНЫХ СПД

Бинарная модель данных - это "реляционная модель данных, в которой каждое отношение содержит по два атрибута" [Л. 226]. "Бинарная модель данных - это графовая модель, в которой вершины являются представлениями простых однозначных атрибутов, а дуги - представлениями бинарных связей между атрибутами" [Л. 316]. Вершины графа соответствуют обобщению экземпляров данных в типы, называемые категориями, а дуги - бинарным отношениям категорий. Обоим направлениям бинарного отношения присваиваются уникальные имена, которые называются функциями доступа. Расширение семантического бинарного графа типов состоит из объектов и связей между ними. Объект это реализация категории. Объекты относятся к одному из двух видов - абстрактным или конкретным объектам. Объекты соединены связями, которые являются реализацией бинарного отношения. Каждая связь помечается в обоих направлениях именами соответствующих функций доступа. Выделим следующие бинарные СПД: **категории, объекты, реализации категорий, бинарные отношения категорий, связи, реализации бинарных отношений, функции доступа бинарного отношения**. Рассмотрим формализованное описание бинарных СПД. Пусть  $A$  - множество имен категорий:

$$A = \{a_1, a_2, \dots, a_i, \dots, a_I\}, \quad (3.1.31)$$

где  $i = \overline{1, I}$ . Тогда, для

$$\forall a_i \in A \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ_i}\}, \quad (3.1.32)$$

где  $i = \overline{1, I}$ ;  $j_i = \overline{1, J_i}$  и  $B_i$  - это множество имен объектов-реализаций категории  $a_i$ .

Пусть  $C$  - это множество имен бинарных отношений категорий:

$$C = \{c_1, c_2, \dots, c_k, \dots, c_K\}, \quad (3.1.33)$$

где  $k = \overline{1, K}$ . Пусть  $D$  - это множество имен функций доступа бинарных отношений:

$$D = \{d_1, d_2, \dots, d_l, \dots, d_L\}, \quad (3.1.34)$$

где  $l = \overline{1, L}$ . Введем обозначение:

$$\mathbf{B} = \bigcup_{i=1}^I \mathbf{B}_i , \quad (3.1.35)$$

где  $\mathbf{B}$  - это множество объектов-реализаций категорий. Тогда, для

$$\forall c_k \in C \quad \exists E_k = \{e_{k1}, e_{k2}, \dots, e_{km_k}, \dots, e_{kM_k}\}, \quad (3.1.36)$$

где  $k = \overline{1, K}$ ;  $m_k = \overline{1, M_k}$ ;  $E_k$  - это множество реализаций бинарного отношения  $c_k$ .

При

этом,

$$\forall e_{km_k} \in E_k \quad \exists \langle b_{i_j}, d_1, b_{i_j}, d_1 \rangle, \quad (3.1.37)$$

где  $b_{i_j} \in B$ ;  $d_1 \in D$ ;  $b_{i_j} \in B$ ;  $d_1 \in D$ ;  $i = \overline{1, I}$ ;  $i' = \overline{1, I}$ ;  $j = \overline{1, J_i}$ ;  $j' = \overline{1, J_{i'}}$ ;  $l = \overline{1, L}$ ;  $\Gamma = \overline{1, L}$ . Таким образом, каждой реализации  $e_{km_k}$  некоторого бинарного отношения  $s_k$  соответствует некоторая четверка, следующего вида:

**<объект-образ, функция доступа объекта-образа,  
объект-прообраз, функция доступа объекта-прообраза>.**

### 3.1.7. ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ СПД СЕМАНТИЧЕСКИХ СЕТЕЙ

"Семантическая модель. Представление понятий в семантической памяти в виде графа, в вершинах которого расположены понятия, в терминальных вершинах - элементарные понятия, а дуги представляют отношения между понятиями" [Л. 226]. Структура, поддерживаемая любой семантической сетевой моделью данных, представляет собой граф. "Категории вершин устанавливаются в соответствии с представляемыми ими предметами. Пример - система, предусматривающая четыре категории вершин: концепты (понятия), события, характеристики (свойства) и значения" [Л. 316]. Дополнительным средством обеспечения выразительной силы семантических сетевых моделей данных является распределение вершин по типам. Кроме того, вводится понятие иерархии классов, которая используется для того, чтобы указать на наследование свойств одного класса другим. Вершины семантической сети могут представлять собой: экземпляры, классы и метаклассы, а дуги - утверждения, порождение экземпляров, бинарные отношения. Выделим следующие основные СПД семантических сетей: **категории вершин, иерархия классов вершин, имена вершин, категории дуг, имена дуг**. Рассмотрим формализованное описание СПД семантических сетей. Пусть  $A$  - множество имен категорий вершин:

$$A = \{a_1, a_2, \dots, a_i, \dots, a_l\}, \quad (3.1.38)$$

где  $i = \overline{1, I}$ . Тогда, для некоторого

$$a_i \in A \quad \exists B_i = \{b_{i_1}, b_{i_2}, \dots, b_{i_j}, \dots, b_{i_{J_i}}\}, \quad (3.1.39)$$

где  $i = \overline{1, I}$ ;  $j = \overline{1, J_i}$  и  $B_i$  - это множество имен классов вершин категории  $a_i$ .

Вместе с этим, пусть существует иерархическое множество двоек:

$$\exists B_i = \{ \langle b_{i_j}, j_i \rangle \mid b_{i_j} \in B_i, j_i = \overline{1, J_i} \} \quad (3.1.40)$$

где  $B_i$  - это иерархически упорядоченное множество имен классов вершин категории  $a_i$ . Далее, пусть для

$$\forall b_{i_j} \in B_i \quad \exists C_{i_j} = \{c_{i_j k_{ij} 1}, c_{i_j k_{ij} 2}, \dots, c_{i_j k_{ij} k_{ij}}, \dots, c_{i_j k_{ij} k_{ij}}\}, \quad (3.1.41)$$

где  $i = \overline{1, I}$ ;  $j = \overline{1, J_i}$ ;  $k_{ij} = \overline{1, K_{ij}}$  и  $C_{i_j}$  - это множество имен вершин класса вершин  $b_{i_j}$  категории  $a_i$ . Пусть  $D$  - множество имен категорий дуг:

$$D = \{d_1, d_2, \dots, d_l, \dots, d_L\}, \quad (3.1.42)$$

где  $l = \overline{1, L}$ . Тогда, для

$$\forall d_l \in D \quad \exists E_l = \{e_{l m_1}, e_{l m_2}, \dots, e_{l m_{m_1}}, \dots, e_{l m_{m_1}}\}, \quad (3.1.43)$$

где  $l = \overline{1, L}$ ;  $m_1 = \overline{1, M_1}$  и  $E_l$  - множество имен дуг категории дуг  $d_l$ . Введем следующие обозначения:

$$C = \bigcup_{i=1}^I \bigcup_{j_i=1}^{J_i} C_{i_j} . \quad (3.1.44)$$

Тогда, можно записать, что:

$$\exists C = \{c_1, c_2, \dots, c_n, \dots, c_N\}, \quad (3.1.45)$$

где  $n = \overline{1, N}$ . Каждая дуга соединяет две вершины, следовательно,

$$\forall e_{lm} \in E_1 \quad \exists E_1 = \{ \langle c_n, c_{n'} \rangle \mid c_n \in C, c_{n'} \in C \}. \quad (3.1.46)$$

### 3.1.8. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРЕДСТАВЛЕНИЕ ДАННЫХ

Обзор современной литературы по теории баз данных показал, что объектно-ориентированные базы данных (ООБД) являются новым этапом развития информационных технологий [Л. 1-2, 36, 50, 63, 136, 200-202, 205, 209, 216, 263, 269-271, 289, 326, 335, 339, 350-352, 362, 374-376, 388-390, 396, 399, 407-408, 411]. В АСОИ, построенных на основе объектно-ориентированной концепции представления данных, функционирование системы рассматривается как взаимодействие связанных объектов путем посылки пользователями сообщений объектам, обработки вызванных объектов, перехода к обработке других объектов и получения требуемых результатов. Особенно важно, что при этом стирается грань между данными и методами (процедурами, правилами) их обработки [Л. 263]. Системы управления ООБД обеспечивают как традиционные функции управления данными, так и функции управления приложениями. Объектно-ориентированное проектирование баз данных является альтернативой структурным методам проектирования и отличается тем, что предполагает инкапсуляцию в одном объекте, как данных, так и методов (правил, процедур) их обработки. Такой подход позволяет создавать **базы данных открытой архитектуры**, с присущими ей свойствами переносимости, мобильности и интероперабельности, а также возможности повторного использования наследованных объектов [Л. 263]. Объектно-ориентированное проектирование баз данных включает те же этапы, что и традиционное структурное проектирование, но отличается от него содержанием работ на каждом этапе. Прежде всего, результатом анализа предметной области является каноническая структура БД, представленная в виде объектной модели. При этом, под **объектом** предметной области понимают некоторую совокупность информационных элементов и методов (процедур) их обработки, а также отношений между ними, составляющих единое целое с точки зрения семантических и процедурных аспектов предметной области.

В объектно-ориентированной базе данных информация хранится в форме объектов, а сама ООБД характеризуется свойствами инкапсуляции, наследования и полиморфизма [Л. 36, 50, 202, 263, 289, 350-352, 362, 374-376, 411]. Свойство инкапсуляции означает, что объекты наделяются некоторой структурой и обладают определенным набором операций, т.е. поведением. Благодаря инкапсуляции объекты можно рассматривать как самостоятельные сущности, отделенные от внешнего мира. Свойство наследования обеспечивает возможность создавать из объектов новые объекты, которые наследуют структуру и поведение своих предшественников, добавляя к ним черты, отражающие их собственную индивидуальность. Свойство полиморфизма означает, что различные объекты могут получать одинаковые сообщения, но реагировать на них по-разному, в соответствии с тем, как реализованы у них методы, реагирующие на сообщения [Л. 36, 50, 202, 263,

289, 388-390, 396, 399, 407-408, 411]. Объектно-ориентированная технология призвана устранить ограничения реляционной технологии проектирования БД, предоставить разработчикам более адекватные, естественные и совершенные средства моделирования предметной области. К их числу относятся: классификация (индивидуальный объект может рассматриваться как частный случай общего понятия), подклассы и суперклассы (экземпляры некоторого класса могут образовывать подмножество другого класса, подклассы наследуют атрибуты и поведение своих суперклассов), реализация концепции обобщения-конкретизации (построение иерархии классов), агрегирование (создание сложных объектов из объектов-компонентов, с возможностью определения отношений "часть-целое"). В объектно-ориентированных СУБД технология объектов охватывает концептуальную и логическую стадии проектирования, а механизмы моделирования данных находятся в распоряжении разработчиков вплоть до этапа конкретной реализации модели данных во внешней памяти. Цель концептуального проектирования заключается в том, чтобы определить относительную структуру объектов. Существуют два важных метода образования объектов: **обобщение** и **агрегация** [Л. 50, 202, 263, 289, 388-390, 396, 411]. Обобщение образует объект из класса других объектов. Агрегация образует объект как отношение между другими объектами. Формально обобщение и агрегация соответствуют теоретико-множественным операциям "объединения" и "декартова произведения".

Следует особо отметить, что в настоящее время, несмотря на огромное количество специальной литературы [Л. 1-2, 36, 50, 63, 136, 200-202, 205, 209, 216, 263, 269-271, 289, 326, 335, 339, 350-352, 362, 374-376, 388-390, 396, 399, 407-408, 411], отсутствуют единые взгляды и стандарты на объектно-ориентированные СПД и модели данных. Сейчас развиваются одновременно несколько различных объектно-ориентированных баз данных. Попытки их интегрирования с реляционными БД привели к созданию промежуточных, так называемых **"объектно-реляционных систем"**, и показали необходимость разработки стандартов для этих моделей данных [Л. 200, 201, 263, 269-270, 335]. Большинство исследователей БД считает [Л. 136, 200-202, 205, 209, 216, 263, 269-271, 289, 326, 335, 339, 350-352, 362, 407-408, 411], что в общем виде объектно-ориентированные СПД должны сочетать в себе лучшие черты СПД модели "сущность-связь" и реляционных СПД, которые были описаны выше. Как будет доказано ниже, реляционные СПД являются частным случаем СПД модели "сущность-связь". Объектно-ориентированная модель представления данных опирается на четыре основных понятия [Л. 1-2, 36, 50, 63, 136, 200-202, 205, 209, 216, 263, 269-271, 289, 326, 335, 339, 350-352, 362, 374-376, 388-390, 396, 399, 407-408, 411]: **объект, характеристика объекта (включая методы его обработки), связь и характеристика связи**. Таким образом, **объектно-ориентированные СПД**, в некотором роде являются **частным случаем СПД модели "сущность-связь"**. Следовательно, в дальнейшем при изучении основных СПД нет необходимости в особом выделении объектно-ориентированных СПД, а достаточно исследовать СПД наиболее общей модели данных "сущность-связь".

### 3.1.9. СИМВОЛЬНЫЕ, НЕСТРУКТУРИРОВАННЫЕ И ДРУГИЕ СПД

Как известно, существуют неструктурированные представления данных в виде некоторого текста, например, в сети Интернет используется **гипертекстовое представление данных** в виде HTML-страниц. База данных является **неструктурированной**, если каждый новый факт о предметной области обладает собственной структурой и схема БД не может быть фиксированной. Документальные данные и компоненты баз знаний являются характерными примерами неструктурированных данных [Л. 1-2, 9, 23, 26, 63-66, 82, 84, 91, 97-98, 123, 124, 142, 171, 178, 184, 190-194, 205, 232-233, 248-250, 255, 263, 266, 296, 304-306, 326-328, 348, 392]. Неструктурированные данные хранятся в БД в виде специальным образом организованных структур - **ассоциативных кортежей**, представляющих собой отдельный **факт**, набор фактов или совокупность индивидуальных характеристик некоторого объекта. В случае представления совокупности характеристик объекта ассоциативный кортеж связывается в БД с относящимся к этому же объекту кортежем отношения, образуя комплекс данных - информационную модель некоторого объекта.

Ассоциативный кортеж представляет собой список произвольного числа термов и ассоциативных кортежей, в качестве термов выступают константы и структуры. Структура представляется функтором, за которым в круглых скобках следуют компоненты, разделяемые запятыми. Компонентами структур могут быть термы и ассоциативные кортежи. В том случае, когда в качестве структур выступают кортежи бинарных отношений, а функтор при этом есть имя бинарного отношения, ассоциативный кортеж может быть интерпретирован **семантической сетью** [Л. 263].

**Символьная модель данных** изначально разрабатывалась в качестве основы для модели представления знаний (МПЗ), базирующейся на процедурной интерпретации систем Поста. При этом исходят из того, что любая СУБД - операционная компонента банка данных - реализует некоторое отображение:

$$\varphi: \Sigma \times \Omega \rightarrow \Sigma \times A, \quad (3.1.47)$$

где  $\Sigma$  - множество состояний информационной компоненты БД,  $\Omega$  - множество возможных обращений к базе данных (называемых также "входным языком"),  $A$  - множество возможных ответов СУБД на обращение [Л. 326-327].

Отметим, что все модели данных предикатного типа в плане представления информации отличаются друг от друга видом базовой предикатной протоструктуры (либо набора таких протоструктур) и содержательной трактовкой ее (их) элементов. Например, для реляционной модели данных такими структурами являются:

**< имя отношения, < имя атрибута<sub>1</sub>, ... , имя атрибута<sub>n</sub>>> и  
< значение атрибута<sub>1</sub>, ... , значение атрибута<sub>n</sub>>**,

т.е. схема отношения и кортеж соответственно; для иерархической и сетевой моделей:

**< адрес записи, запись >**,

где запись: < агрегат<sub>1</sub>, ... , агрегат<sub>n</sub>>, агрегат (сегмент) это - < элемент<sub>1</sub>, ... , элемент<sub>n</sub>>, а элемент в свою очередь - скалярное значение либо адрес записи - члена [Л. 326-327].

В символьной модели данных на уровне представления данных **никаких ограничений на структуру** элементов БД не накладывается, а единственным исходным понятием помимо **базы данных** является **факт**. Структура факта при необходимости может определяться пользователем самостоятельно. В символьной базе данных

все данные хранятся в виде элементов - **страниц**, каждая из которых представляет собой некоторую страницу символов с заданным значением количества строк и количества позиций в строке. Кроме того, в символьной модели данных существует еще база метаданных (БМД), которая представляет собой такую же символьную базу данных. Отличие базы метаданных в том, что ее элементы имеют специфическую структуру и интерпретируются системой управления базой данных (СУБД) как элементы описания данных. Обычно база метаданных - это совокупность сведений, используемых СУБД для контроля целостности базы данных и санкционирования доступа к ней, а также для оптимизации хранения и поиска информации.

Возможно, представление базы метаданных в виде множества контекстно-свободных порождающих правил. Каждое такое правило есть конструкция вида:

$$\alpha \rightarrow \beta, \quad (3.1.48)$$

где  $\alpha$  - нетерминальный символ;  $\beta$  - строка, состоящая из символов алфавита  $V$  и нетерминалов, которые представляют собой наименования их подэлементов. При этом, аксиома контекстно-свободной грамматики  $G_t$ , множество правил которой есть  $D_t$ , обозначается  $\alpha_0$  и имеет смысл "факт". Существенным моментом является то, что в символьной модели данных принята так называемая закрытая интерпретация мира, согласно которой отсутствие факта в БД равносильно его отсутствию в предметной области [Л. 326].

Следовательно, **символьная модель данных** относится к **неструктурированным** моделям данных. Таким образом, неструктурированные и символьные СПД могут быть сведены к структурам представления данных **семантической сети**, которая, в свою очередь, сама является частным случаем модели данных "**сущность-связь**".

Кроме выше описанных, существует еще достаточно много различных разновидностей моделей данных (позиционных множеств, активные, временные, графоориентированные, слабоструктурированные, инфологические, концептуальные, плоские, неструктурированные дескрипторные и фреймовые, графовые, формально-логические, алгебраические, теоретико-множественные и т.д.) [Л. 1-2, 9, 11, 17, 23-24, 33, 36, 47, 48, 50, 54-57, 63, 72, 78-82, 91, 97-104, 107-118, 132, 136, 158-164, 170-176, 197-213, 215-216, 219, 227-228, 231-233, 235-239, 241-244, 248-253, 255-258, 261-264, 268-271, 274-280, 287-289, 292-296, 306, 314-319, 321-328, 333, 335, 338-340, 342-356, 358-368, 374-383, 387-413, 492-501]. В связи с ограниченным объемом данной монографии не представляется возможным подробно рассматривать все существующие СПД.

Целесообразным является выделение наиболее известных "традиционных" и "показательных" моделей данных, тем более, что как будет доказано ниже, практически все остальные модели данных могут быть сведены либо к модели данных "сущность-связь", либо к другим основным структурам представления данных традиционных моделей данных, которые будут проанализированы ниже. Перейдем к анализу основных СПД традиционных моделей данных.

## 3.2. АНАЛИЗ СПД ТРАДИЦИОННЫХ МОДЕЛЕЙ ДАННЫХ

### 3.2.1. УРОВЕНЬ ПРЕДСТАВЛЕНИЯ ДАННЫХ

Прежде всего, введем понятие "уровень представления данных".

**Определение 3.2.1.** Если каждому элементу одного множества соответствует некоторое другое множество элементов, тогда будем говорить, что существует двухуровневое представление данных.

Если, в свою очередь, каждому элементу множества второго уровня соответствует некоторое множество элементов, то можно говорить о трех уровнях представления данных. И так далее, по аналогии, можно ввести понятие  $N$ -го уровня представления данных.

Понятие "уровень представления данных" необходимо для сравнительного анализа возможностей по представлению данных СПД различных моделей данных. В общем случае, может существовать  $N$ -уровневое представление данных.

Отметим, что большее количество уровней представления данных одной модели, говорит и о больших "семантических" возможностях такой модели, т.е. о более богатых возможностях отражения семантики предметной области.

Формулы 3.1.1 и 3.1.2 можно рассматривать, с точки зрения введенного понятия, и как формальное определение двухуровневого представления данных.

### 3.2.2. АНАЛИЗ РЕЛЯЦИОННЫХ СПД

Как было показано, выделяют следующие реляционные СПД: отношения; атрибуты; значения. В реляционной модели не выделяют специально структуры для представления данных о связях сущностей. Проанализируем реляционные СПД. Формулы 3.1.1 и 3.1.2 показывают, что существует двухуровневое представление данных. Далее, формулы 3.1.2 и 3.1.3 показывают, что имеет место три уровня представления данных. Итак, реляционные СПД являются трехуровневыми, причем, без разделения описания данных о сущностях от описания данных о связях сущностей.

### 3.2.3. АНАЛИЗ СЕТЕВЫХ СТРУКТУР ПРЕДСТАВЛЕНИЯ ДАННЫХ

В сетевой модели выделяют следующие СПД о сущностях: типы записей; элементы данных типов записей; реализации типов записей. Анализ формул 3.1.4, 3.1.5 и 3.1.6 показывает, что сетевые СПД о сущностях имеют три уровня. Выделяют следующие сетевые СПД о связях сущностей: типы наборов-связей; реализации наборов-связей. При этом каждому элементу множества  $Z$  из формулы 3.1.7 сопоставляется два различных множества, описанных формулами 3.1.8 и 3.1.9. Такое описание, непосредственно, не соответствует понятию уровня представления данных. В этом случае, можно говорить, что каждому элементу из множества  $Z$  соответствует некоторое количество множеств. Каждому такому множеству может быть определенным образом присвоено некоторое имя (идентификатор). Затем, можно рассмотреть некоторое новое множество имен множеств, которые и соответствовали элементу  $Z$ . Таким образом, получается, что каждому элементу  $z$  множества  $Z$  соответствует некоторое множество  $W_1$  - множество имен множеств, которое на самом деле, состоит только из двух элементов, например:  $Y$  и  $X$ . Далее, каждому элементу  $w_{1p}$  множества  $W_1$  соответствует некоторое множество,

обозначаемое  $V_{1p_1}$ . Следовательно, можно представить формулы 3.1.7 - 3.1.10 в следующей записи:

$$Z = \{z_1, z_2, \dots, z_l, \dots, z_L\}, \quad (3.2.1)$$

где  $l = \overline{1, L}$ . Тогда для

$$\forall z_l \in Z \quad \exists W_l = \{w_{1l}, w_{12}, \dots, w_{1p_1}, \dots, w_{1p_1}\}, \quad (3.2.2)$$

где  $l = \overline{1, L}$ ;  $p_1 = \overline{1, P_1}$  и  $W_l$  - это множество имен множеств реализаций записей набора-связи  $z_l$ . Далее, для

$$\forall w_{1p_1} \in W_l \quad \exists V_{1p_1} = \{v_{1p_1 1}, v_{1p_1 2}, \dots, v_{1p_1 q_{p_1}}, \dots, v_{1p_1 Q_{p_1}}\}, \quad (3.2.3)$$

где  $l = \overline{1, L}$ ;  $p_1 = \overline{1, P_1}$ ;  $q_{1p_1} = \overline{1, Q_{1p_1}}$  и  $V_{1p_1}$  - это множество реализаций записей типа  $w_{1p_1}$  набора-связи типа  $z_l$ . Отметим, что

если  $p=1$ , то  $V_{1p_1} = Y_1$ , а

если  $p=2$ , то  $V_{1p_1} = X_1$ ;

где  $Y_1$  и  $X_1$ , берутся в смысле формул 3.1.8 и 3.1.9, соответственно. Анализ формул 3.2.1 - 3.2.3 показывает, что существует трехуровневое сетевое представление данных о связях сущностей. Таким образом, получаем, что в сетевой модели данных существуют трехуровневые представления данных о сущностях и о связях сущностей.

#### 3.2.4. АНАЛИЗ ИЕРАРХИЧЕСКИХ СТРУКТУР ПРЕДСТАВЛЕНИЯ ДАННЫХ

В иерархической модели выделяют следующие СПД о сущностях: категория типа записи (корневая или порожденная); типы записей; элементы данных типов записей; реализации типов записей. Анализ формул 3.1.11, 3.1.12, 3.1.13 и 3.1.14 показывает, что имеет место четырехуровневое представление данных о сущностях. В иерархической модели выделяют следующие СПД о связях сущностей: реализации наборов-связей. Анализ формул 3.1.15, 3.1.16 и 3.1.17, по аналогии с формулами 3.2.1 - 3.2.3, показывает, что можно ввести некоторое множество  $W$ , т.е. множество имен множеств реализаций наборов-связей. Далее, множества  $Z$  и  $Y$ , из формул 3.1.15 и 3.1.16, соответственно, можно рассматривать, как некоторые множества  $V_1$  и  $V_2$ . Тогда получим, что

$$\exists W = \{w_1, w_2, \dots, w_p, \dots, w_P\}, \quad (3.2.4)$$

где  $p = \overline{1, P}$ . Тогда для

$$\forall w_p \in W \quad \exists V_p = \{v_{p1}, v_{p2}, \dots, v_{p q_p}, \dots, v_{p Q_p}\}, \quad (3.2.5)$$

где  $p = \overline{1, P}$ ;  $q_p = \overline{1, Q_p}$ . Тогда,

если  $p=1$ , то  $V_1=Z$ , а

если  $p=2$ , то  $V_2=Y$ .

Анализ формул 3.2.4 и 3.2.5 показывает, что в иерархической модели существует два уровня представления данных о связях сущностей. Таким образом, получаем, что в иерархической модели данных существуют четырехуровневое представление данных о сущностях и двухуровневое представление данных о связях сущностей.

### 3.2.5. АНАЛИЗ СПД МОДЕЛИ "СУЩНОСТЬ-СВЯЗЬ"

В модели данных "сущность-связь" выделяют следующие СПД о сущностях: классы сущностей; множества сущностей; множество атрибутов и множество значений. Рассмотрим формулы 3.1.18, 3.1.19, 3.1.24 и 3.1.25. На основании анализа данных формул можно сделать вывод, что имеет место четырехуровневое представление данных о сущностях. Выделяют следующие СПД о связях сущностей: классы связей; множества связей; множество атрибутов и множество значений. Проанализировав формулы 3.1.22, 3.1.23, 3.1.26 и 3.1.27, приходим к выводу, что имеет место 4-уровневое представление данных о связях сущностей. Кроме того, выделяют еще и такие СПД, как - роли множеств сущностей в связях. С этой точки зрения, каждая связь описывается формулой 3.1.30, т.е. некоторым множеством двоек, следующего вида: **<сущность, роль>**. В данном случае, целесообразно поступить аналогично описанию связей в формулах 3.2.1-3.2.3. То есть, первый уровень - множество имен связей; второй уровень - множество двоек; третий уровень - атрибуты (составляющие) каждой двойки; четвертый уровень - это значения атрибутов двоек, что в данном случае является значением сущностей и значением ролей сущностей в связях. Таким образом, такое описание связей также является четырехуровневым. В модели данных "сущность-связь" имеют место 4-уровневые представления данных о сущностях и о связях сущностей.

### 3.2.6. АНАЛИЗ БИНАРНЫХ СТРУКТУР ПРЕДСТАВЛЕНИЯ ДАННЫХ

В бинарной модели данных выделяют следующие СПД о сущностях: категории; реализации категорий. На основе анализа формул 3.1.31 и 3.1.32, можно сделать вывод, что имеет место двухуровневое представление данных о сущностях. Также, в бинарной модели данных выделяют следующие СПД о связях сущностей: бинарные отношения категорий; реализации бинарных отношений; функции доступа бинарного отношения. Проведя анализ формул 3.1.33 и 3.1.36, можно сделать вывод о существовании двух уровней. Но, затем, на основании анализа формулы 3.1.37 можно сделать вывод, принимая во внимание формулы 3.2.1, 3.2.2 и 3.2.3, что существует еще два уровня. Следовательно, имеет место (существует) четырехуровневое представление данных о связях сущностей. Таким образом, в бинарных моделях данных имеют место двухуровневое представление данных о сущностях и четырехуровневое представление данных о связях сущностей.

### 3.2.7. АНАЛИЗ СПД СЕМАНТИЧЕСКИХ СЕТЕЙ

В семантических сетях выделяют следующие СПД о сущностях: категории вершин; иерархия классов вершин и вершины. На основании анализа формул 3.1.38, 3.1.39 и 3.1.41, можно сделать вывод, что существует трехуровневое представление данных о сущностях. Выделяют следующие СПД о связях сущностей: категории дуг и дуги. На основании анализа формул 3.1.42, 3.1.43, 3.1.44, 3.1.45 и 3.1.46, с учетом формул 3.2.1, 3.2.2 и 3.2.3, представляется возможным сделать вывод о том, что существует четырехуровневое представление данных о связях сущностей. Таким образом, в семантических сетях имеют место трехуровневое представление данных о сущностях и четырехуровневое представление данных о связях сущностей.

### 3.3. СРАВНЕНИЕ ОСНОВНЫХ СПД ТРАДИЦИОННЫХ МОДЕЛЕЙ ДАННЫХ

На основании анализа основных СПД традиционных моделей данных можно сделать следующие выводы:

- в реляционных моделях данных имеет место, общее, как для сущностей, так и для связей, 3-уровневое представление данных;
- в сетевых моделях имеют место 3-уровневое представление данных о сущностях и 3-уровневое представление данных о связях;
- в иерархических моделях имеют место 4-уровневое представление данных о сущностях и 2-уровневое представление данных о связях;
- в моделях "сущность-связь" имеют место 4-уровневое представление данных о сущностях и 4-уровневое представление данных о связях;
- в бинарных моделях имеют место 2-уровневое представление данных о сущностях и 4-уровневое представление данных о связях;
- в семантических сетях имеют место 3-уровневое представление данных о сущностях и 4-уровневое представление данных о связях.

Наиболее приближенными к файловым структурам современных ЭВМ являются реляционные СПД. При явном программном разделении реляционных 3-уровневых таблиц на таблицы описания сущностей и на таблицы описания связей, получаем аналог сетевых СПД. Следовательно, "семантические" возможности, по представлению данных в современных ЭВМ, сетевых и реляционных моделей данных являются практически идентичными. Сразу же подчеркнем, что реляционный подход предъявляет гораздо более жесткие требования к процессу проектирования конкретных баз данных и к непосредственной реализации БД. Но, кроме того, реляционный подход, за счет больших усилий разработчиков, позволяет создавать более эффективные программные продукты для современных ЭВМ, чем сетевой подход. Возможно, этим и объясняется столь высокая популярность реляционных баз данных по сравнению с другими БД.

Рассматривая другие модели данных, получаем, что если программно вводить для реляционных СПД четвертый уровень представления данных (например, путем явного указания названий классов сущностей, категорий вершин или дуг) и ввести разделение описания сущностей от описания связей, то на таких реляционных структурах можно реализовать и поддерживать любые другие структуры традиционных моделей данных, в том числе и структуры модели "сущность-связь". Проведенный анализ СПД показал, что СПД модели "сущность-связь" являются обобщением и развитием структур всех традиционных моделей данных (в том числе и перечисленных выше в разделе 3.1.9, включая объектно-ориентированные, символьные, гипертекстовые, неструктурированные и другие модели данных), так как только в этой модели данных имеют место 4-уровневые представления данных и о сущностях, и о связях. Все остальные традиционные модели являются некоторыми частными случаями наиболее общей модели "сущность-связь". В таблице 3.3.1 показано сопоставление СПД о сущностях различных моделей данных, а в таблице 3.3.2 сопоставляются СПД о связях сущностей основных традиционных моделей данных. Таким образом, **СПД модели "сущность-связь" являются обобщением и развитием всех СПД традиционных моделей данных.**

Таблица 3.3.1.

## Сопоставление СПД о сущностях

№	модель "сущность-связь"	реляционная модель	сетевая модель	иерархическая модель	бинарная модель	семантические сети
1	класс сущностей	-----	-----	категория типа	-----	категории вершин
2	множества сущностей	отношения	типы записей	типы записей	категории	классы вершин
3	атрибуты сущностей	атрибуты	элементы данных	элементы данных	-----	-----
4	значения сущностей	значения	реализации типов	реализации типов	объекты, реализации	вершины

Таблица 3.3.2.

## Сопоставление СПД о связях сущностей

№	модель "сущность-связь"	реляционная модель	сетевая модель	иерархическая модель	бинарная модель	семантические сети
1	класс связей	-----	-----	-----	-----	категории дуг
2	множества связей	отношения	типы наборов	типы записей	бинарные отношения	дуги
3	атрибуты роли сущностей в связи	атрибуты	-----	элементы данных	функции доступа	-----
4	значения сущностей	значения	реализации связей	реализации связей	реализации	-----

Возможно, что при появлении новых архитектур ЭВМ, реляционные модели потеряют свою привлекательность, но в настоящее время, реляционный подход является наиболее эффективным. Важно особо подчеркнуть, что "семантические" возможности реляционных и сетевых СПД являются практически идентичными, равными; т.е. все, что можно представить в сетевых СПД - может быть представлено и в реляционных, а также и наоборот. Это объясняется тем, что как для сетевых, так и для реляционных СПД присуще 3-уровневое представление данных. Конечным пользователям и проектировщикам баз данных удобнее представлять данные в сетевом виде, но для достижения наибольшей эффективности конкретных программных продуктов - необходимо использовать реляционный подход. Однако, уже стали актуальными такие задачи, для решения которых рамки реляционного подхода являются слишком узкими, а следовательно, требуются принципиально новые решения в сфере разработки новых СПД.

Итак, сравнительный анализ СПД традиционных моделей данных показал, что повышение "семантической" описания различных предметных областей достигается путем введения дополнительного уровня представления данных (четвертого в нашей классификации), например - это класс сущностей, категории типов или категории вершин. В соответствии с этим становится понятным, что, при необходимости дальнейшего повышения "семантических" возможностей, можно идти по пути добавления новых уровней представления данных, вплоть до получения некоторых N-уровневых СПД.

### **3.4. ОБОБЩЕНИЕ ОСНОВНЫХ СПД ТРАДИЦИОННЫХ МОДЕЛЕЙ ДАННЫХ**

В настоящее время активно разрабатываются новые направления в области БД, например: создание новых методов и средств организации БД [Л. 101], базы видеоданных [Л. 322], временно- [Л. 122] и объектно-ориентированные [Л. 289] модели, символично-продукционный подход [Л. 326], а также современные методы организации физического уровня хранения данных [Л. 216]. Кроме того, ведутся исследования и в области теоретического развития фундаментальных основ (СПД) традиционных моделей данных.

Как было показано выше, все традиционные модели данных могут быть описаны, с точки зрения СПД, как "графо-табличные" модели данных. Таким образом, получаем, что существует некоторый класс моделей данных, называемых "графо-табличными", для описания СПД которых целесообразно ввести новое понятие: **трехуровневое представление данных в виде одномерных таблиц** специальной структуры. Это понятие является обобщением и развитием СПД всех рассмотренных выше моделей данных.

#### **3.4.1. ОПРЕДЕЛЕНИЕ ОДНОМЕРНОЙ ТАБЛИЦЫ ТРЕХУРОВНЕВОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ**

Существует два основных понятия в представлении данных: **таблицы** и **графы**. "Возможно простейшим способом представления данных являются таблицы. ... Аналогии таблиц - файлы, записи, поля - применяются и в обработке данных. Табличные формы представления характерны и для большинства моделей данных, причем различие в структуре таблиц - часто один из признаков дифференциации моделей" [Л. 316]. В современных ЭВМ все данные хранятся в виде файлов, которые имеют структуру одномерной таблицы данных и состоят из: **названия таблицы, названий столбцов и записей-строк**, в которых и хранятся данные. Любой граф может быть описан некоторой таблицей. С другой стороны, любую таблицу можно представить в виде графа. Выше, в разделе 3.2.1, было введено новое понятие "уровень представления данных". Тогда, все основные СПД "графо-табличных" (традиционных) моделей данных могут быть описаны в виде одномерных таблиц трехуровневого представления данных.

**ОПРЕДЕЛЕНИЕ 3.4.1.** Одномерная таблица трехуровневого представления данных (ОТД-3) - это таблица, предназначенная для хранения данных и состоящая из следующих элементов: **заголовка таблицы, заголовки столбцов и клетки таблицы**.

Где, "заголовок таблицы" - это некоторая совокупность данных, описывающих всю таблицу, например, ее название, тип, класс, и т.д.

"**Заголовок столбца**" - это некоторая совокупность данных, которая описывает соответствующий столбец клеток ОТПД-3. Каждый столбец ОТПД-3 имеет только один "заголовок столбца". Каждый "заголовок столбца" относится только к одному столбцу. Столбец - это аналог атрибута или элемента данных в соответствующих моделях данных. "Заголовок столбца" может содержать, например, данные о названии атрибута-элемента данных, о формате хранения, о типе данных, и т.д. Отметим, что при таком подходе, столбец данных - это заголовок столбца и совокупность всех соответствующих клеток, относящихся к этому столбцу.

Тогда, **таблица данных** - это заголовок таблицы, совокупность всех заголовков столбцов таблицы и все совокупности клеток всех столбцов данной таблицы. Важно отметить, что, если в реляционных таблицах количество клеток в различных столбцах одной таблицы должно быть одинаковым, то в ОТПД-3 это не обязательно, т.е. в одной и той же таблице могут быть столбцы разного размера, содержащие разное количество клеток.

"**Клетки таблицы**" - это некоторая совокупность хранимых данных, которые и образуют "тело" таблицы представления данных, т.е. это, непосредственно, и есть хранимые, накапливаемые данные, организованные в соответствии с заголовками столбцов таблицы.

Рассмотрим формализованное описание СПД ОТПД-3. Пусть  $\mathbf{A}$  - множество имен заголовков таблиц:

$$\mathbf{A} = \{a_1, a_2, \dots, a_i, \dots, a_I\}, \quad (3.4.1)$$

где  $i = \overline{1, I}$ . Тогда, для

$$\forall a_i \in \mathbf{A} \quad \exists \mathbf{V}_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ_i}\}, \quad (3.4.2)$$

где  $i = \overline{1, I}$ ;  $j_i = \overline{1, J_i}$  и  $\mathbf{V}_i$  - это множество имен заголовков столбцов заголовка таблицы  $a_i$ . Далее, для

$$\forall b_{ij} \in \mathbf{V}_i \quad \exists \mathbf{C}_{ij} = \{c_{ij1}, c_{ij2}, \dots, c_{ijk_{ij}}, \dots, c_{ijK_{ij}}\} \quad (3.4.3)$$

где  $i = \overline{1, I}$ ;  $j_i = \overline{1, J_i}$ ;  $k_{ij} = \overline{1, K_{ij}}$  и  $\mathbf{C}_{ij}$  - это множество значений клеток таблицы заголовка столбца  $b_{ij}$  заголовка таблицы  $a_i$ .

Одномерность определяется тем, что после проектирования базы данных количество таблиц и столбцов в них фиксировано. Таким образом, "вширь" структуры данных не изменяются (не растут, не увеличиваются), а добавляются только строки (клетки) таблицы, т.е. все таблицы могут изменяться только в одном направлении - увеличиваться или уменьшаться вниз", при добавлении новых данных или удалении существующих строк. Отметим, что вместо термина "**одномерная таблица трехуровневого представления данных**" можно употреблять равнозначный термин: "**трехуровневая одномерная таблица представления данных**". Если не возникает необходимости специально оговаривать количество уровней представления данных, то будем употреблять термин "одномерная таблица представления данных" (ОТПД), в том числе и как синоним ОТПД-3. Еще раз подчеркнем, что "тело" таблицы - это еще не вся таблица представления данных. Существуют еще: заголовок таблицы и заголовки столбцов. Причем, все или только некоторые заголовки могут непосредственно храниться отдельно от самого тела таблицы, но только наличие всех трех составляющих образует таблицу представления данных. Таким образом, видно, что **одномерная таблица трехуровневого представления данных** - это не одна таблица, в традиционном понимании, а такое представление данных, которое совмещает **три различные таблицы**. Первая таблица описывает ОТПД-3 в целом, вторая описывает столбцы ОТПД-3, и только третья таблица содержит хранимые, накапливаемые дан-

ные, в виде клеток, т.е. строк-записей. Более того, одномерная таблица трехуровневого представления данных - это термин логического уровня представления данных, тогда как на физическом уровне одна ОТПД может отображаться тремя физическими таблицами или, даже, большим количеством таблиц. Например, клетки, относящиеся только к одному столбцу, могут физически храниться в отдельном файле ЭВМ. Следовательно, клетки каждого столбца ОТПД-3 могут храниться в отдельных файлах или таблицах. Подчеркнем, что ОТПД является способом организации хранения данных логического (концептуального) уровня и именно в этом состоит ее главное отличие от традиционных файлов или таблиц памяти ЭВМ. Исходя из смысла, введенного термина "одномерная таблица представления данных", проанализируем СПД основных традиционных графо-табличных моделей данных и определим соотношения между ними и элементами ОТПД-3.

### 3.4.2. СРАВНЕНИЕ РЕЛЯЦИОННЫХ СПД И ОТПД-3

Рассмотрим сравнение ОТПД и реляционных СПД. Имена отношений будут непосредственно содержаться в заголовках таблиц, имена атрибутов будут содержаться в заголовках столбцов, а значения - в клетках ОТПД (рис. 3.4.1). Подчеркнем, что значение величины  $N=Const$ , т.е. она фиксирована, а изменяться может только значение числа  $K$ . Данное замечание справедливо и для всех последующих аналогичных таблиц этого раздела.

### 3.4.3. СРАВНЕНИЕ СЕТЕВЫХ СПД И ОТПД-3

Сетевые СПД могут быть представлены в формализме ОТПД следующим образом. В терминах ОТПД типы записей, элементы данных и реализации типов записей могут быть представлены в виде некоторых таблиц. Но, в связи с тем, что в сетевой модели выделяются два вида: записи и наборы-связи, то необходимо выделить и два вида ОТПД. Итак, типы записей, элементы данных и реализации типов записей будут содержаться в ОТПД, например, вида "записи". Это можно представить следующим образом: заголовок таблицы содержит название вида таблицы: "записи" и название типа записей; заголовок столбца содержит название элемента данных; клетки таблицы содержат реализации типов записей.

Все типы наборов-связей и реализации связей будут содержаться в ОТПД, например, вида "наборы". Это можно представить следующим образом (рис. 3.4.2): заголовок таблицы содержит название вида таблицы: "наборы", а также название типа набора-связей; заголовки столбцов будут двух типов, например: "образ связи" и "прообраз связи"; клетки таблицы будут содержать реализации наборов-связей. Для представления сетевых структур необходимо создать такие ОТПД, в заголовках таблиц которых содержится значение вида, что и позволит делить различные ОТПД по видам. Непосредственное представление в каждой ОТПД СПД сетевой модели, аналогично представлению в ОТПД реляционных структур (рис. 3.4.1 и 3.4.2).

### 3.4.4. СРАВНЕНИЕ ИЕРАРХИЧЕСКИХ СПД И ОТПД-3

Для описания иерархических СПД в терминах ОТПД потребуются выделение трех видов таблиц: "корневой", "порожденный" и "наборы-связи". Тогда необходимо, чтобы "категория типа записи" содержалась в заголовке ОТПД вместе с "типом за-

### СРАВНЕНИЕ СПД РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ И ОТПД-3

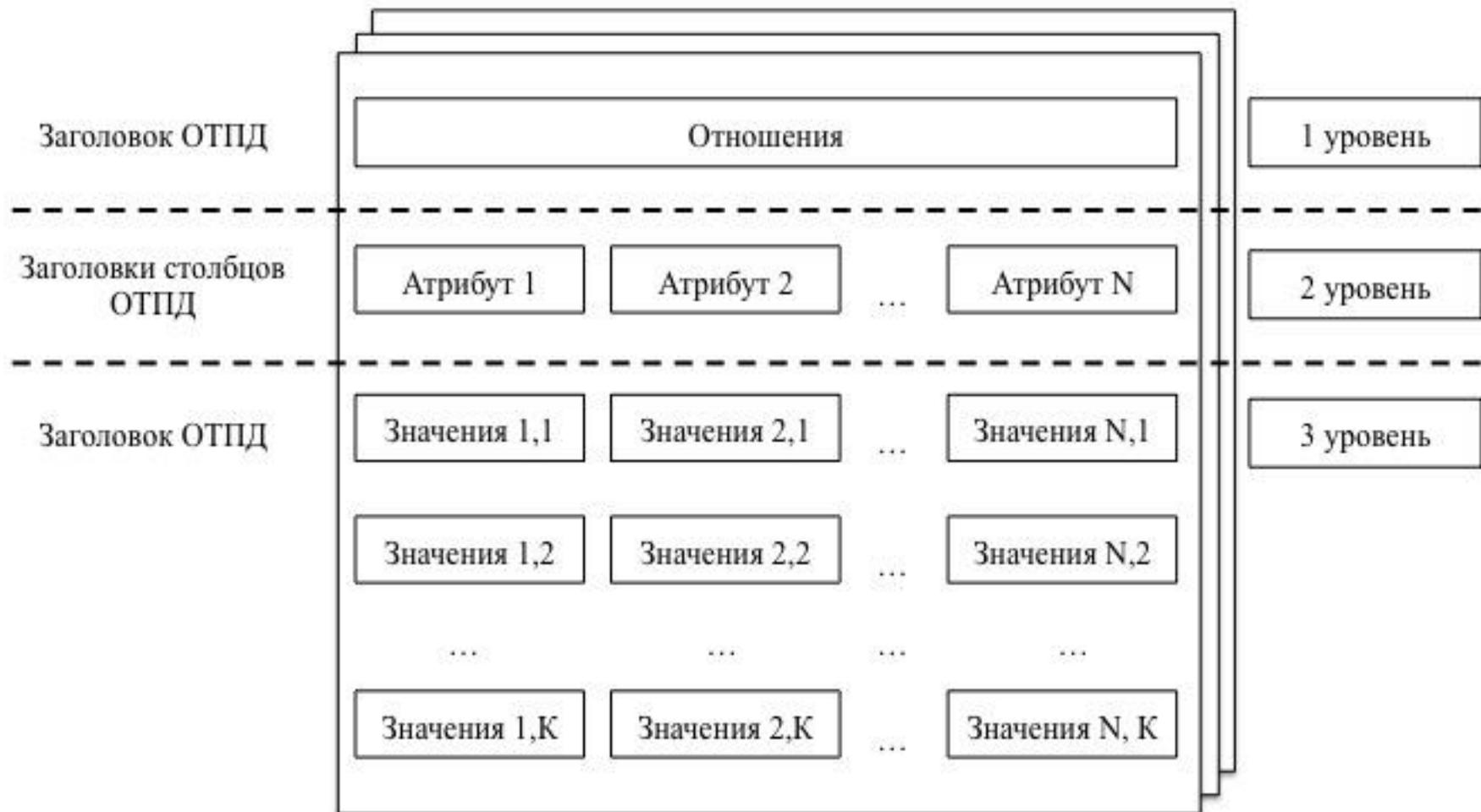


Рис. 3.4.1. СПД реляционной модели данных в формализме ОТПД-3.

### СРАВНЕНИЕ СПД СЕТЕВОЙ МОДЕЛИ ДАННЫХ И ОТПД-3

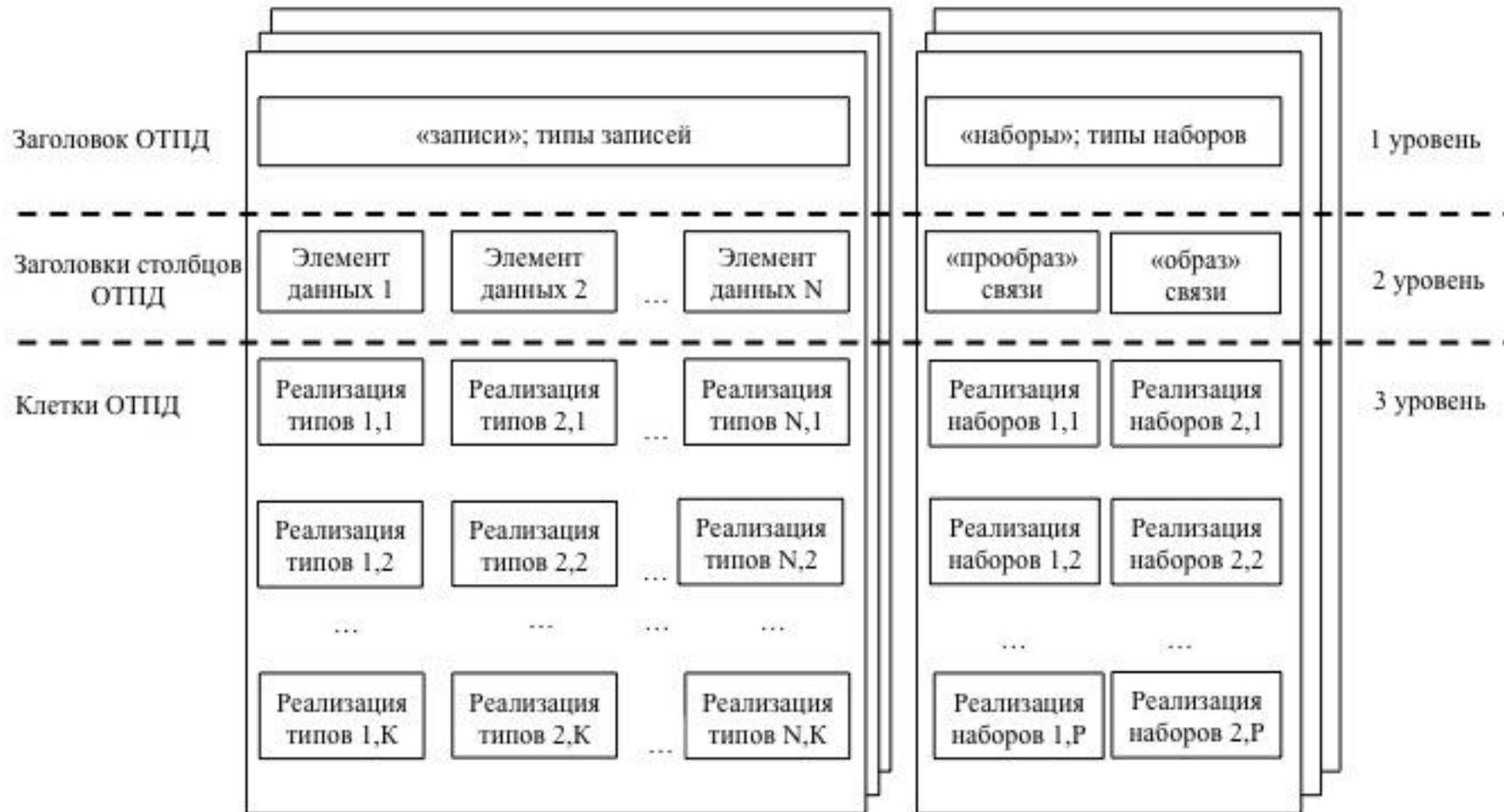


Рис. 3.4.2. СПД сетевой модели данных в формализме ОТПД-3.

писей". "Элементы данных" будут содержаться в заголовках столбцов, а "реализации типов записей" - в клетках ОТПД. В таблице, вида "наборы-связи", создается два заголовка столбцов, а клетки ОТПД будут содержать "реализации наборов-связей". Для представления иерархических структур необходимо создать ОТПД трех различных видов. Признаки каждого вида таблицы должны содержаться в заголовках ОТПД, а в остальном, представление иерархических СПД в формализме ОТПД-3 аналогично представлению в ОТПД сетевых структур (рис. 3.4.2).

### 3.4.5. СРАВНЕНИЕ СПД МОДЕЛИ "СУЩНОСТЬ-СВЯЗЬ" И ОТПД-3

Модель "сущность-связь" практически не реализована на ЭВМ, но и ее можно представить в терминах ОТПД, а, следовательно, можно и реализовать на ЭВМ. Данная модель данных является наиболее "богатой" для отображения и моделирования различных предметных областей. Более того, эта модель является обобщением сетевых, реляционных и иерархических СПД. Представим описание модели "сущность-связь" в терминах ОТПД. Прежде всего, необходимо ввести признак разделения ОТПД для представления сущностей или для представления связей. Для этого введем в заголовки таблиц признак вида таблицы, который, например, принимал бы значения либо "сущность", либо "связь". Далее, для всех ОТПД, представляющих сущности необходимо ввести в соответствующие заголовки таблиц признак класса сущности, а для всех ОТПД, представляющих связи - признак класса связей (размерности связи). Заголовки столбцов ОТПД "сущностей" должны содержать соответствующие названия атрибутов описания, которые описывают каждую конкретную сущность, а также размерность, значность таких атрибутов. Заголовки столбцов ОТПД "связей" должны содержать названия атрибутов описания связей, их размерности, названия ролей множеств сущностей, а также названия этих множеств сущностей. Клетки ОТПД должны содержать конкретные значения сущностей или связей (рис. 3.4.3). Итак, заголовки ОТПД-3 должны содержать либо класс сущностей и название множества сущностей, либо класс связей и название множества связей. Заголовки столбцов ОТПД-3 должны содержать либо атрибуты сущностей, либо атрибуты связей, либо названия ролей сущностей. Возможно включение и других признаков. Клетки ОТПД-3 будут содержать значения либо сущностей, либо связей. Подчеркнем, что введение ОТПД-3 и ее составных частей, т.е. заголовков таблиц, заголовков столбцов и клеток, позволяет описать в этом формализме даже СПД модели данных "сущность-связь". Кроме того, напомним, что числа **N** и **M** - фиксированы, а увеличиваться могут только значения чисел **K** и **P**, так как данные таблицы одномерны.

### 3.4.6. СРАВНЕНИЕ СПД БИНАРНОЙ МОДЕЛИ И ОТПД-3

Структуры данных бинарной модели могут быть представлены в терминах ОТПД следующим образом: категории будут содержаться в заголовках таблиц типа, например, "категории" (эти таблицы будут состоять лишь из одного столбца); объекты - реализации категорий будут отражены в клетках таблицы. Бинарные отношения категорий будут содержаться в заголовках таблиц вида, например: "отношения категорий", функции доступа бинарного отношения будут отражены в заголовках столбцов, а связи - реализации бинарных отношений будут содержаться в клетках таблицы. Итак, для описания СПД бинарной модели достаточно ввести два вида ОТПД: "категории" и "отношения категорий".

### СРАВНЕНИЕ СПД МОДЕЛИ «СУЩНОСТЬ-СВЯЗЬ» И ОТПД-3

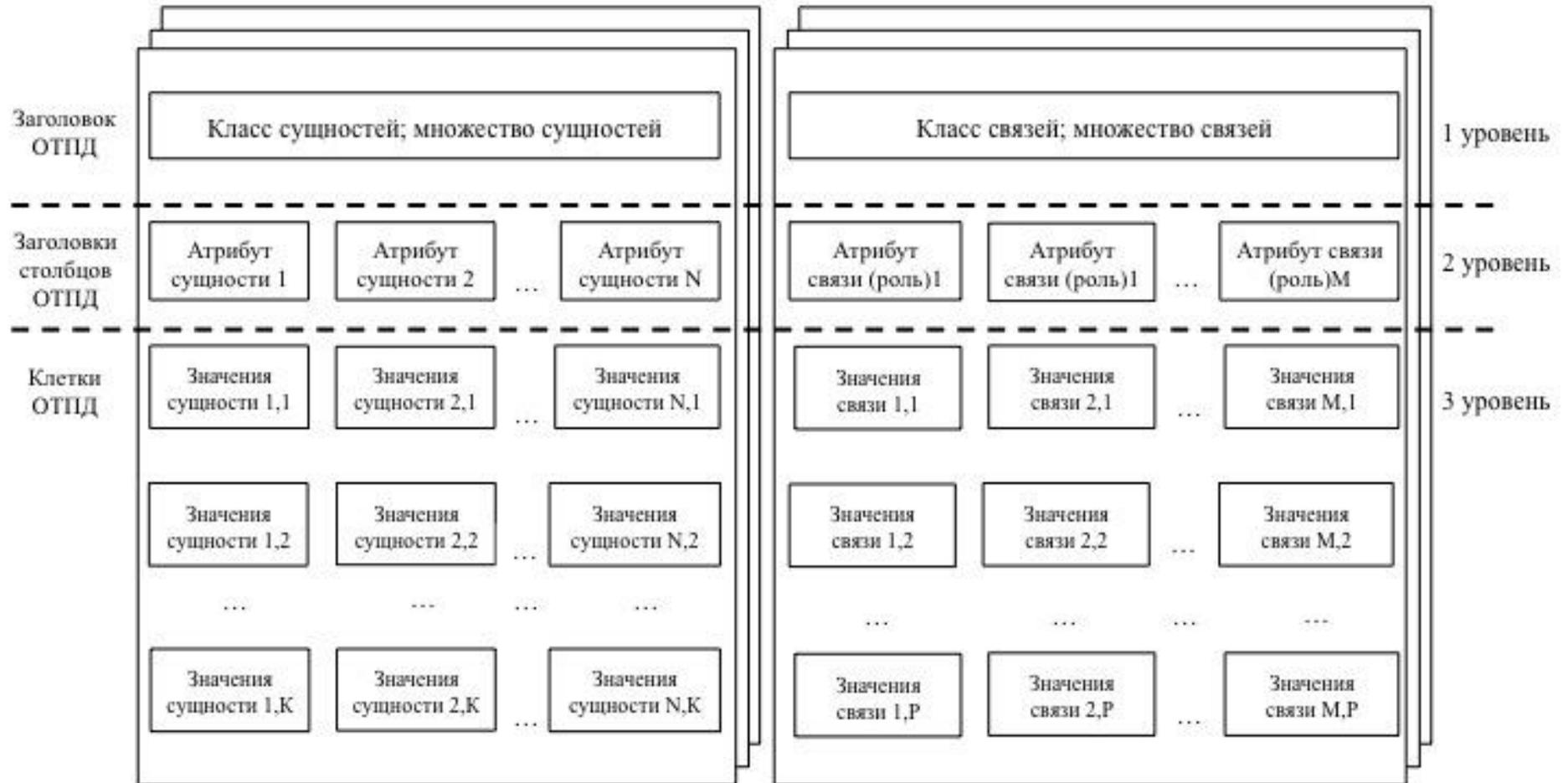


Рис. 3.4.3. СПД модели "сущность-связь" в формализме ОТПД-3.

### 3.4.7. СРАВНЕНИЕ СПД СЕМАНТИЧЕСКИХ СЕТЕЙ И ОТПД-3

Для описания СПД семантических сетей введем следующие два вида ОТПД, например: "вершины" и "дуги". Одномерные таблицы представления данных вида "вершина" должны содержать в заголовках таблиц информацию о категории вершины, о значении в иерархии классов данной вершины, а также об имени вершины. Конкретные значения экземпляров, классов и метаклассов будут содержаться в клетках ОТПД-3.

Одномерные таблицы представления данных, типа "дуга", должны содержать в заголовке таблицы ОТПД-3 информацию о категории дуги и о названии дуги. Конкретные значения утверждений, порождений экземпляров и бинарных отношений будут содержаться в клетках таблицы. Структура самой таблицы этого вида должна описываться при помощи заголовков столбцов ОТПД-3. Следовательно, для описания СПД семантических сетей необходимо, прежде всего, выделить два различных вида одномерных таблиц представления данных: "вершины" и "классы". Информация о виде одномерных таблиц представления данных может быть либо явно отражена в заголовках таблиц, либо это разделение может быть произведено структурно, в неявном виде, например: путем введения некоторых правил задания имен файлов, т.е. некоторые имена файлов для некоторых определенных типов ОТПД. Отметим, что возможны и другие способы выделения различных типов одномерных таблиц представления данных. Рассмотрим обобщение основных традиционных СПД.

### 3.4.8. ОБОБЩЕНИЕ ОСНОВНЫХ ТРАДИЦИОННЫХ СПД

Реляционные структуры описываются в формализме ОТПД-3 наиболее просто из всех традиционных моделей данных. Сетевые структуры, с точки зрения формализма ОТПД-3, отличаются от реляционных только тем, что вводится отдельное описание связей и рассматриваются таблицы двух различных видов (см. рис. 3.4.1 и 3.4.2). Иерархические структуры отличаются от сетевых тем, что вводится различие вершин не только по типам, но и на более высоком уровне. Причем, такое различие, например: по "категориям типов записей", вводится и в семантических сетях: "категории вершин", и в модели "сущность-связь": "класс сущностей" и "класс связей", а также и в других моделях. Наиболее сложное описание СПД имеет место в модели "сущность-связь", поэтому можно сопоставить различные структуры в виде двух таблиц (ОТПД-4). Первая таблица (таблица 3.4.1) соотносит СПД описания сущностей или объектов традиционных моделей данных, а также структуры одномерных таблиц представления данных - ОТПД. Вторая таблица (таблица 3.4.2) сопоставляет СПД описания связей сущностей традиционных моделей и одномерных таблиц представления данных - ОТПД. Из этих таблиц видно, что при описании, с одной стороны - сущностей, объектов, типов записей, вершин, а с другой стороны - связей объектов, бинарных отношений, дуг; возникает много аналогичных СПД. Как уже отмечалось, и сущности, и связи описываются некоторыми таблицами. Например, в реляционной модели, но в этой модели не поддерживается системное разделение видов таблиц, что затрудняет и усложняет описание реальных предметных областей. Многие пользователи баз данных считают, что сетевые модели являются более приемлемыми для их предметных областей. Но на самом деле, все то, что может быть описано средствами сетевой модели, может быть описано и средствами реляционной модели, естественно, с учетом лишь затраты больших усилий при проектировании и эффективной реализации конкретной БД.

Сравнение основных структур представления данных об объектах, сущностях

№ п/п	Одномерная четырехуровневая таблица представления данных ОТПД-4	Модель данных «сущность-связь»	Реляционная модель данных	Сетевая модель данных	Иерархическая модель данных	Бинарная модель данных	Семантические сети
1	вид таблицы	класс сущностей	————	————	категория типа	————	категории вершин
2	заголовок таблицы	множества сущностей	отношения	типы записей	типы записей	категории	классы вершин
3	заголовки столбцов таблицы	атрибуты сущностей	атрибуты	элементы данных	элементы данных	————	————
4	клетки таблицы	значения сущностей	значения	реализации типов	реализации типов	объекты, реализации	вершины

Сравнение основных структур представления данных о связях объектах, сущностях

№ п/п	Одномерная четырехуровневая таблица представления данных ОТПД-4	Модель данных «сущность-связь»	Реляционная модель данных	Сетевая модель данных	Иерархическая модель данных	Бинарная модель данных	Семантические сети
1	вид таблицы	класс связей	————	————	————	————	категории дуг
2	заголовок таблицы	множества связей	отношения	типы наборов-связей	————	бинарные отношения	дуги
3	заголовки столбцов таблицы	атрибуты роли сущностей в связи	атрибуты	————	————	функции доступа	————
4	клетки таблицы	значения связей	значения	реализации наборов-связей	реализации связей	реализации отношений	————

Анализ таблиц 3.4.1 и 3.4.2 позволяет сделать следующий вывод о том, что наиболее полное описание, представление данных либо об объектах, либо о связях объектов достигается с помощью четырехуровневых одномерных таблиц представления данных (ОТПД-4), содержащих следующие уровни представления данных (структуры): вид таблицы, заголовок таблицы, заголовок столбца, клетки таблицы. Например, для описания сетевых СПД в формализме ОТПД-4 можно использовать однотипные таблицы и для записей, и для связей, что показано на рис. 3.4.4.

Для описания СПД модели "сущность-связь" в формализме ОТПД-4 необходимо использовать два типа таблиц (рис. 3.4.5). В некоторых случаях, например, при описании СПД семантических сетей (рис. 3.4.6), бинарных или иерархических моделей, возможно применение разноуровневых одномерных таблиц представления данных. Но в таких случаях можно использовать ОТПД наиболее высокого уровня (например, ОТПД-4 на рис. 3.4.6), потому, что в более общей таблице всегда можно представить другие таблицы, содержащие меньшее количество уровней представления данных. Подчеркнем, что  $N$  и  $M$  - фиксированные числа, т.е. константы.

### 3.5. ПЯТИУРОВНЕВАЯ ОДНОМЕРНАЯ ТАБЛИЦА ПРЕДСТАВЛЕНИЯ ДАННЫХ

Если ввести пятый уровень, называемый "класс таблицы", то это позволит отделять описания объектов от описаний связей этих объектов. **Пятиуровневые ОТПД (ОТПД-5)** позволяют описывать (например, даже для модели "сущность-связь"), как сущности, так и их связи в едином формате (рис. 3.5.1), т.е. в виде однотипных таблиц представления данных. Подчеркнем, что числа  $N$  и  $M$  - фиксированы (константы), а изменяются только числа  $K$  и  $P$ . Причем, в отличие от реляционных структур такое представление (ОТПД-5) позволяет в явном виде сохранять или определять отличия связей от объектов, а также различия по категориям типов, по категориям вершин, по классам сущностей, по категориям дуг и по классам связей, соответственно. Это позволит "не потерять семантическую" как семантическим сетям, так и модели "сущность-связь".

Таким образом, ОТПД-5 представляют собой развитие и обобщение: реляционных, сетевых, иерархических, бинарных, символьных моделей данных, семантических сетей, модели "сущность-связь", а также объектно-ориентированных, "неструктурированных", гипертекстовых и многих других (см. раздел 3.1.9) СПД. Итак, приведем полный список элементов пятиуровневых одномерных таблиц представления данных (таблица 3.5.1): класс таблицы, вид таблицы, заголовок таблицы, заголовок столбца, клетки таблицы. Рассмотрим формализованное описание СПД ОТПД-5. Пусть  $A$  - множество имен классов таблиц:

$$A = \{a_1, a_2, \dots, a_i, \dots, a_I\}, \quad (3.5.1)$$

где  $i = \overline{1, I}$ . Тогда, для

$$\forall a_i \in A \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ}\}, \quad (3.5.2)$$

где  $i = \overline{1, I}$ ;  $j = \overline{1, J}$  и  $B_i$  - это множество имен видов таблиц класса таблиц  $a_i$ . Далее,

для

$$\forall b_{ij} \in B_i \quad \exists C_{ij} = \{c_{ij1}, c_{ij2}, \dots, c_{ijk_{ij}}, \dots, c_{ijK_{ij}}\} \quad (3.5.3)$$

### СРАВНЕНИЕ СПД СЕТЕВОЙ МОДЕЛИ ДАННЫХ И ОТПД-4



Рис. 3.4.4. СПД сетевой модели данных в формализме ОТПД-4.

### СРАВНЕНИЕ СПД МОДЕЛИ «СУЩНОСТЬ-СВЯЗЬ» И ОТПД-4

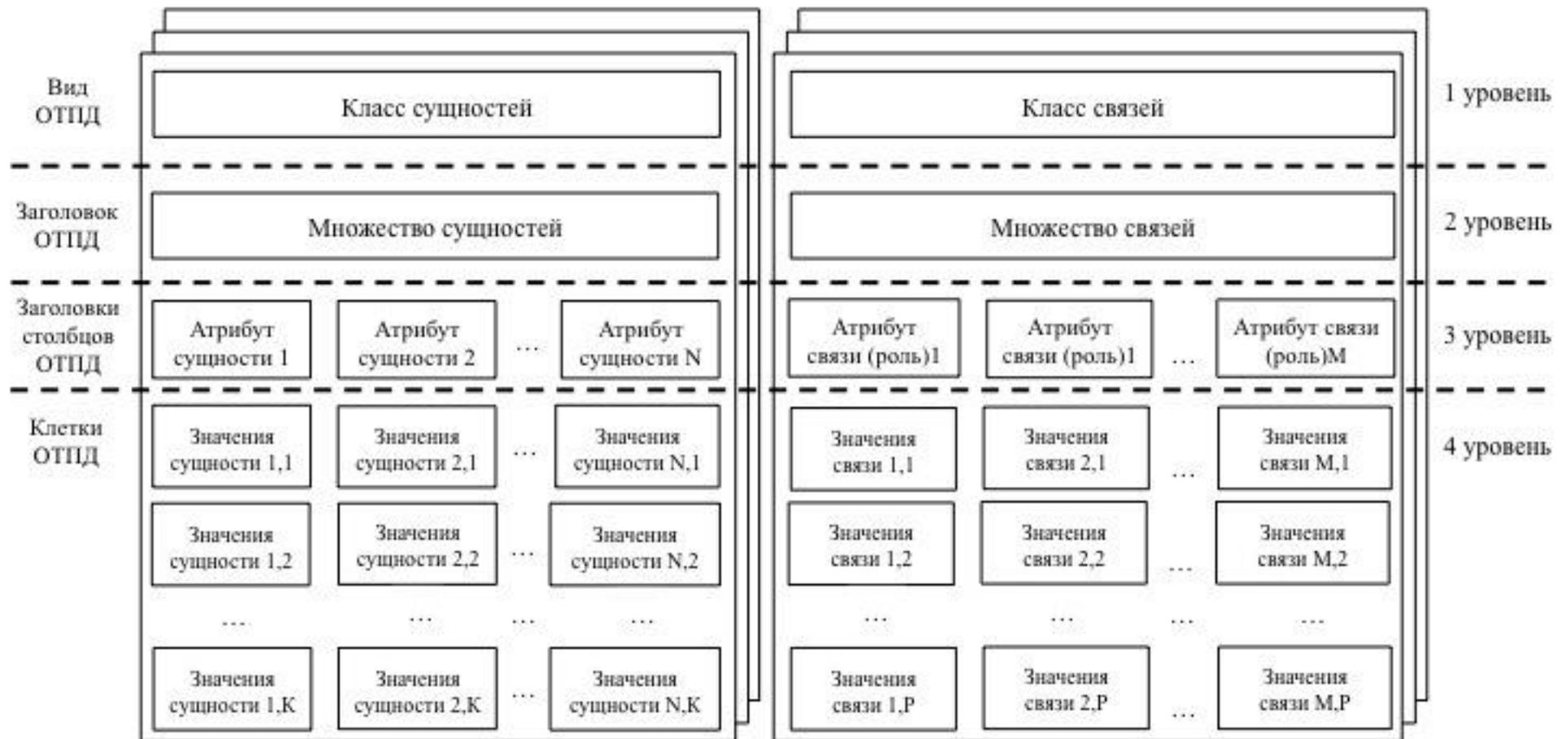


Рис. 3.4.5. СПД модели "сущность-связь" в формализме ОТПД-4

### СРАВНЕНИЕ СПД СЕМАНТИЧЕСКИХ СЕТЕЙ И ОТПД-4

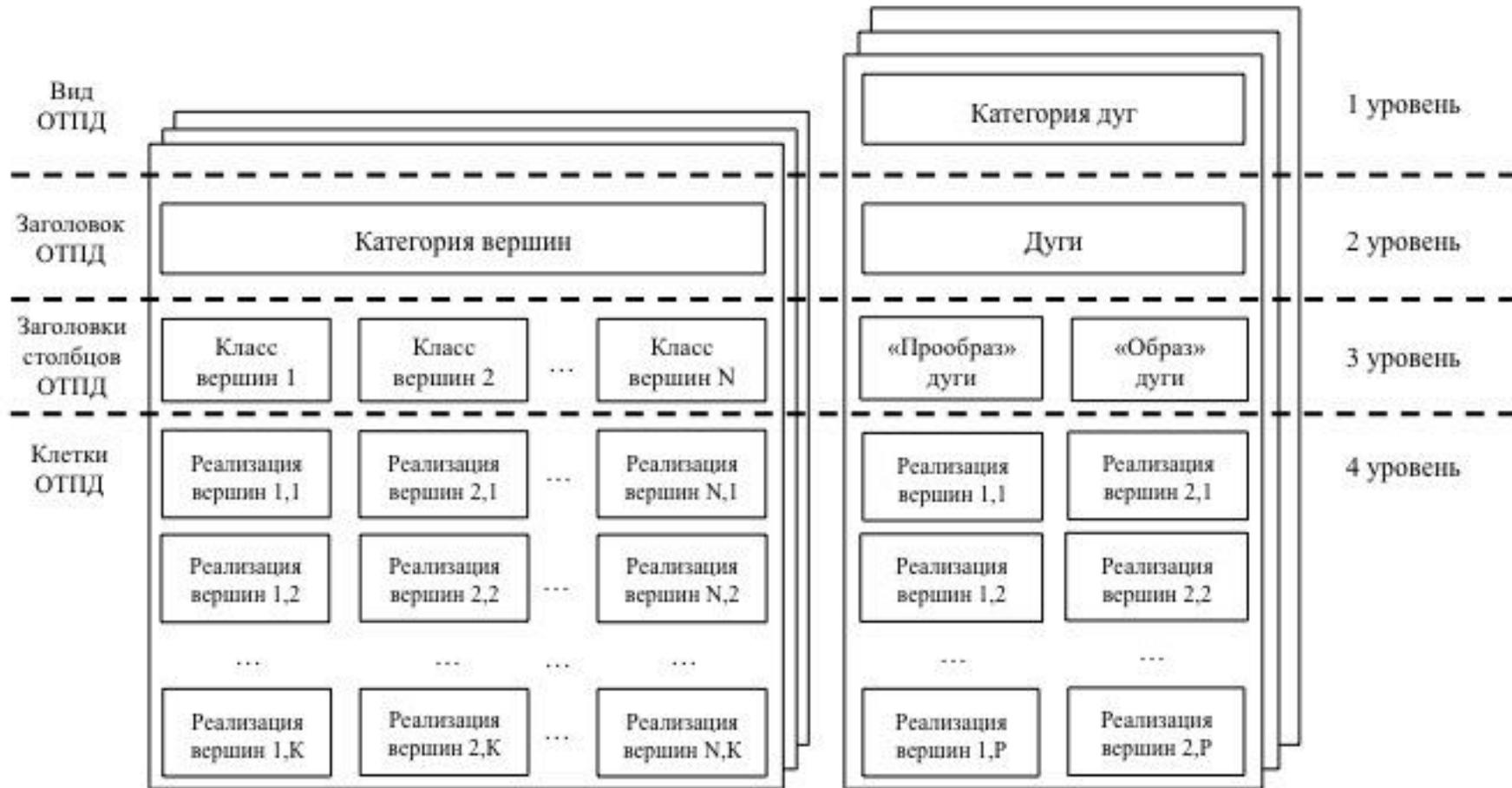


Рис. 3.4.6. СПД семантических сетей в формализме ОТПД-4.

**СРАВНЕНИЕ СПД МОДЕЛИ «СУЩНОСТЬ-СВЯЗЬ» И ОТПД-5**



Рис. 3.5.1. СПД модели "сущность-связь" в формализме ОТПД-5.

**Сравнение основных структур представления данных традиционных моделей данных и  
одномерных пятиуровневых таблиц представления данных ОТПД-5**

№ п/п	Одномерная пятиуровневая таблица представления данных ОТПД-5	Модель данных «сущность-связь»	Реляционная модель данных	Сетевая модель данных	Иерархическая модель данных	Бинарная модель данных	Семантические сети
1	класс ОТПД	—————	—————	—————	—————	—————	—————
2	вид ОТПД	класс сущностей; класс связей	—————	—————	категория типов	—————	категории вершин; категории дуг
3	заголовок ОТПД	множества сущностей; множества связей	отношения	типы записей; типы наборов	типы записей	категории; бинарные отношения	классы вершин; дуги
4	заголовки столбцов ОТПД	атрибуты роли сущностей в связи	атрибуты	элементы данных	элементы данных	функции доступа	—————
	строки-значения ОТПД	значения сущностей; значения связей	значения	реализации типов; реализации наборов	реализации типов; реализации связей	объекты, реализации; реализации отношений	вершины

где  $\mathbf{i} = \overline{1, I}$ ;  $\mathbf{j}_i = \overline{1, J_i}$ ;  $\mathbf{k}_{i_j} = \overline{1, K_{i_j}}$  и  $C_{i_j}$  - это множество имен заголовков таблиц вида таблиц **Ошибка! Невозможно создать объект из кодов полей редактирования.** класса таблиц  $\mathbf{a}_i$ . В свою очередь, для

$$\forall c_{i_j k_{j_i}} \in C_{i_j} \\ \exists D_{i_j k_{j_i}} = \left\{ d_{i_j k_{j_i} 1}, d_{i_j k_{j_i} 2}, \dots, d_{i_j k_{j_i} l_{j_i k_{j_i}}}, \dots, d_{i_j k_{j_i} L_{j_i k_{j_i}}} \right\} \quad (3.5.4)$$

где  $\mathbf{i} = \overline{1, I}$ ;  $\mathbf{j}_i = \overline{1, J_i}$ ;  $\mathbf{k}_{i_j} = \overline{1, K_{i_j}}$ ;  $\mathbf{l}_{i_j k_{j_i}} = \overline{1, L_{i_j k_{j_i}}}$  и  $D_{i_j k_{j_i}}$  - это множество имен заголовков столбцов заголовка таблицы  $c_{i_j k_{j_i}}$  вида таблиц  $b_{i_j}$  класса таблиц  $\mathbf{a}_i$ . И, наконец, для

$$\forall d_{i_j k_{j_i} l_{j_i k_{j_i}}} \in D_{i_j k_{j_i}} \quad \exists E_{i_j k_{j_i} l_{j_i k_{j_i}}} = \\ = \left\{ e_{i_j k_{j_i} l_{j_i k_{j_i}} 1}, e_{i_j k_{j_i} l_{j_i k_{j_i}} 2}, \dots, e_{i_j k_{j_i} l_{j_i k_{j_i}} m_{j_i k_{j_i} l_{j_i k_{j_i}}}}, \dots, e_{i_j k_{j_i} l_{j_i k_{j_i}} M_{j_i k_{j_i} l_{j_i k_{j_i}}}} \right\}, \quad (3.5.5)$$

где  $\mathbf{i} = \overline{1, I}$ ;  $\mathbf{j}_i = \overline{1, J_i}$ ;  $\mathbf{k}_{i_j} = \overline{1, K_{i_j}}$ ;  $\mathbf{l}_{i_j k_{j_i}} = \overline{1, L_{i_j k_{j_i}}}$ ;  $\mathbf{m}_{i_j k_{j_i} l_{j_i k_{j_i}}} = \overline{1, M_{i_j k_{j_i} l_{j_i k_{j_i}}}}$  и  $E_{i_j k_{j_i} l_{j_i k_{j_i}}}$  - это множество значений клеток таблицы заголовка столбца  $d_{i_j k_{j_i} l_{j_i k_{j_i}}}$  заголовка таблицы  $c_{i_j k_{j_i}}$  вида таблиц  $b_{i_j}$  класса таблиц  $\mathbf{a}_i$ . (Подчеркнем, что после проектирования ОТПД-5, числа  $I, J, K$  и  $L$  - фиксируются и больше не изменяются (т.е. они - константы), а изменяется только число  $M$ ).

Практическая реализация ОТПД-5 может быть проведена следующим образом. Во всех заголовках ОТПД-5 выделяются определенные поля под названия класса таблицы и под название вида таблицы, причем эти названия не обязательно должны иметь символьный вид, понимаемый пользователями, более того, эти обозначения должны быть как можно более краткими. Помимо разделения объектов и связей, пять уровней повышают "семантичесность" такого представления данных. Отметим, что деление по классам таблиц и по видам таблиц может быть реализовано структурно, т.е. выделяются различные уровни хранения ("директории") для каждого класса, а затем и для каждого вида таблиц. Кроме того, при реализации ОТПД-5 "возможны варианты" и вариантов этих достаточно много. Выбор конкретного варианта практической реализации зависит от поставленной задачи, от возможностей ЭВМ, от желания пользователя и от многих других факторов. Итак, **наиболее общей формой представления данных в виде таблиц** для традиционных моделей данных является **формализм одномерных таблиц пятиуровневого представления данных**.

Таким образом, ОТПД-5 является развитием и обобщением формализмов СПД всех традиционных моделей данных. ОТПД-5 позволяет наглядно показать ограничения многоуровневых одномерных таблиц и их отличие от многомерных таблиц, которые могут быть созданы на основе простых одномерных таблиц представления данных. Пятиуровневое представление данных открывает новые перспективы по расширению возможностей существующих моделей данных, а также определяет новое направление по созданию и использованию (на базе как шести- или семиуровневых одномерных таблиц представления данных, так, в перспективе, и **N-мерных представлений данных** и, особо отметим, **знаний**) новых моделей данных и знаний.

Одним из наиболее перспективных направлений развития теории представления данных являются миварные (эволюционные, адаптивные, синергетические, многомерные, динамические, объектно-ориентированные) структуры унифицированного представления данных и правил, описание которых приведено в следующей главе данной монографии. Перейдем к изложению миварной концепции представления данных.

#### 4. МИВАРНАЯ КОНЦЕПЦИЯ МНОГОМЕРНОГО ЭВОЛЮЦИОННОГО ОБЪЕКТНО-СИСТЕМНОГО УНИФИЦИРОВАННОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ И ПРАВИЛ

##### 4.1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МИВАРНОГО ПРОСТРАНСТВА ПРЕДСТАВЛЕНИЯ ДАННЫХ И ПРАВИЛ

Прежде всего, отметим, что данные - это "информация, представленная в виде, пригодном для обработки автоматическими средствами при возможном участии человека" [Л. 226]. Следовательно, для различных автоматизированных систем сбора и обработки информации (АССОИ) должны быть разработаны соответствующие концепции представления (накопления) информации в виде некоторым образом организованных баз данных. Традиционная графо-табличная концепция представления данных, получившая свое наибольшее развитие в модели данных "сущность-связь" [Л. 316], как было показано выше, ориентирована на моделирование таких предметных областей, которые могут быть описаны в виде изменяющихся данных некоторой фиксированной структуры или вообще без структуры. Это означает, что могут быть описаны только такие предметные области, в которых можно однозначно зафиксировать, определить некоторые сущности, затем, для каждой сущности однозначно зафиксировать определенные свойства и некоторые связи, отношения между различными сущностями. В случае выявления либо новой сущности, либо нового свойства, либо нового отношения СПД должны быть спроектированы заново или перепроектированы, т.е. практически каждый раз создается новая информационная модель (структура представления) предметной области. Неструктурированные модели данных хорошо подходят для накопления данных в виде гипертекста, но совершенно не предназначены для моделирования и выявления внутренней структуры исследуемых объектов. Поэтому, неструктурированные модели данных далее подробно рассматриваться не будут, как не предназначенные для решения требуемых, исследуемых в данном разделе задач.

Однако, кроме традиционного графо-табличного описания данных (и указанного выше неструктурированного подхода к накоплению данных) представляется возможным использовать **другой подход**, основанный на использовании **эволюционного (адаптивного) динамического многомерного объектно-системного дискретного пространства унифицированного представления данных и правил**, называемого в дальнейшем, для краткости и однозначности толкования терминов, **миварным пространством представления данных**, или: миварным пространством. Подчеркнем, что термин "**мивар**" является специально созданным словом, обозначающим только то, для чего его придумали и применяют, а история его возникновения не представляет собой научного интереса. Основой миварного подхода к представлению данных является целостное, адаптивное (эволюционное), динамичное, многомерное, а при необходимости и объектно-ориентированное, описание изучаемой (моделируемой) предметной области, при котором сущности (вещи, объекты), свойства и отношения могут переходить друг в друга, в зависимости от предмета исследования, т.е. сущность может быть свойством другой сущности или сущность может быть отношением других сущностей. Таким образом, будем исходить из того, что некоторое "нечто" может быть и сущностью, и свойством, и отношением, в зависимости от исследуемого предмета и его роли в этом исследовании. Например, понятие "стол" - с одной стороны это сущность, но в некотором смысле это может быть и отношение составных частей: столешницы, ножек, крепежа и т.п. Кроме того, это может рассматриваться даже

как свойство, например, "ящик" или "пень" могут иметь свойство, в некоторых ситуациях, "быть столом". Обсудим более подробно необходимость разработки миварной концепции представления данных.

#### 4.1.1. НЕОБХОДИМОСТЬ РАЗРАБОТКИ НОВОЙ КОНЦЕПЦИИ ПРЕДСТАВЛЕНИЯ ДАННЫХ

Существует целый класс познающих систем (например - АССОИ), задачей которых является изучение и моделирование сложных (реальных) динамичных предметных областей. Основное отличие познающих АССОИ от других систем, заключается в том, что происходит накопление и обработка разнородной информации. Накопление информации приводит к необходимости обучения, постоянного уточнения и изменения СПД: выявляются новые объекты, появляются новые свойства, возникают новые отношения. Кроме того, возможно и удаление или замена устаревших понятий. В познающих АССОИ необходимо описывать поведение сложной исследуемой предметной области или некоторой развивающейся системы. Существуют различные подходы к решению этой проблемы. Например, рассмотрим вопросы применения "скользящей аксиоматической системы" [Л. 251-252]. Отметим, что в кибернетике поведение системы обычно рассматривается в фиксированном пространстве  $X^i$  [Л. 251-253], причем **поведение системы** - это изменение ее состояний во времени, исходом которого является некоторый результат [Л. 241, с.26]. Однако, использование фиксированного пространства состояний не позволяет описать поведение сложной развивающейся системы. Поэтому для создания моделей развивающихся систем, способных более адекватно отображать изменение системы, ее развитие (т.е., движение объективной реальности), необходимо применять более совершенный аппарат. В частности, одним из возможных средств математического моделирования развивающихся систем, является применение **математической структуры Н. Бурбаки**, на основе которой можно синтезировать целый класс различных моделей [Л. 252, с.34]. В соответствии с работой И.М. Яглома [Л. 342, с. 50] математическая структура Н. Бурбаки - это система:

$$S = \langle M, R \rangle, \quad (4.1.1)$$

где  $M = \{a, b, c, \dots\}$  - основное множество, а  $R = \{R_1, R_2, R_3, \dots\}$  - множество унарных, бинарных и других отношений. Однако, такая структура (формула 4.1.1) не описывает динамики развития системы. Следовательно, для учета фактора времени в систему (4.1.1) необходимо добавить еще и время. Тогда эта математическая система будет описываться следующим выражением:

$$S(t) = \langle M(t), R(t) \rangle, \quad (4.1.2)$$

где  $t$  - текущее время [Л. 252, с. 34-35]. Впрочем, понятно также, что описать изменение структуры в непрерывном времени едва ли возможно. Поэтому Ю.Г. Ростовцев в [Л. 252] предлагает, что более целесообразно будет ввести понятие **квазидинамической структуры**, которая не изменяет свои элементы  $M$  и  $R$  в пределах интервала времени  $[t_0, t_1]$ . После окончания этого интервала параметры структуры изменяются скачками. При этом надо лишь определить правило изменения параметров структуры. Следовательно, для создания математических моделей, обеспечивающих более высокую степень адекватности по сравнению с существующими, необходимо разрабатывать новый математический аппарат, например, основанный на применении "скользящей аксиоматической системы", отслеживающей (с помощью человека) изменения, происходящие в реальном мире [Л. 252].

Кроме создания новых математических аппаратов представления накапливаемых данных, существуют различные подходы и к созданию новых, в том числе - динамических и изменяемых, систем логической обработки данных с различными параметрами, а также разрабатываются новые технические средства и аппаратные реализации таких математических систем (многомерных, динамических, нейросетевых, многопроцессорных, с распараллеливанием, на основе формальных грамматик и т.п.) [Л. 5-13, 17, 20-28, 31-40, 43-48, 54-57, 62-66, 71-80, 84, 88-98, 107-118, 120-132, 135-152, 155-167, 170-180, 189-216, 219, 224-228, 231-253, 255-258, 261-264, 268-271, 274-280, 284-289, 292-300, 304- 328, 331, 333, 337-340, 342-413]. Такие примеры дают нам основания для разработки новых математических подходов к моделированию сложных развивающихся систем с применением технологий баз данных и логической обработки информации.

Например, для динамической логической обработки информации может использоваться система операторных алгоритмов Ван Хао [Л. 79, с. 57]. Такая система задается последовательностью приказов, каждому из которых присваивается вполне определенный номер. Эта последовательность включает в себя следующие данные: тип операции, которую надо выполнить над заданным объектом, и номер приказа, который следует выполнять дальше. Таким образом, каждая команда имеет форму:

$$i: \begin{array}{|c|c|c|} \hline w & a & b \\ \hline \end{array} \quad (4.1.3)$$

где  $i$  -номер приказа;  $w$  - символ некоторой одноместной частичной функции;  $a$ ,  $b$  - номера приказов. Содержательно команда трактуется так: найти величину  $w(x)$ , далее выполнить приказ  $a$  над величиной  $w(x)$ , если она определена и приказ  $b$  над величиной  $x$ , если значение  $w(x)$  не определено.

Доказательство того, что система операторных алгоритмов Ван Хао образует алгоритмическую систему приведено в работе А.И. Мальцева [Л.180]. Тем не менее, возможности этих операторных алгоритмов Ван Хао ограничены, но идея динамичности описания предметной области может быть нами использованы в дальнейших исследованиях. Таким образом, даже с учетом всего известного математического аппарата, для более адекватного отображения объектов реального мира целесообразно разрабатывать новый, более совершенный математический аппарат, в том числе и в области баз данных и правил (знаний), что обуславливает необходимость разработки новой концепции представления данных. Применение традиционной концепции представления данных для познающих АССОИ приводит к большим затратам различных ресурсов, но не позволяет эффективно решать многие, сравнительно несложные, задачи, как правило, связанные с необходимостью обучения АССОИ. Следовательно, возникает **проблема разработки такой концепции представления данных**, которая сочетала бы в себе достоинства традиционных подходов и новые возможности по обучению и моделированию предметных областей с **изменяемой** (переменной, адаптивной, эволюционной) **структурой представления данных**. Эта задача является **актуальной**, прежде всего, для **познающих систем сбора и обработки информации**, которые должны одновременно и контролировать состояние изучаемого объекта, т.е. для **познающе-диагностических систем, например, для автоматизированных систем сбора и обработки информации**.

Рассмотрим миварную концепцию представления данных, которая позволяет решить сформулированную проблему по накоплению и обработке данных с изменяемой структурой без дополнительного проектирования, то есть, фактически по обучению и накоплению данных с эволюционной, адаптивной, динамичной структурой.

#### 4.1.2. ОСНОВНЫЕ ПОЛОЖЕНИЯ МИВАРНОЙ КОНЦЕПЦИИ ПРЕДСТАВЛЕНИЯ ДАННЫХ

Кратко изложим то, в чем состоит сущность миварной концепции представления данных. Будем исходить из того, что реально существует некоторая предметная область, а определенная познающая система изучает и моделирует данную область. Полученные сведения о предметной области накапливаются в некоторой системе представления данных. Для решения этой задачи введем три равнозначных и равнозависимых понятия: **вещь**, (объект, сущность); **свойство**, (атрибут, характеристика); **отношение**, (связь, взаимодействие). Этим трех выделенных понятий-категорий, как было показано выше (в разделе 2), достаточно для представления любой информации о любой предметной области. Особо отметим, что такие понятия как, например: элемент, объект, система, функция, структура, организация и подобные им, являются вторичными, производными по отношению к трем основным, а, следовательно, могут быть выражены через них.

В частности, из этого следует, что любое правило (процедура, связь, взаимодействие и т.п.) вне зависимости от формы его представления, является всего лишь - частным случаем отношения, а, следовательно, нет необходимости в специальном выделении в системах хранения "правил" и формирования "базы знаний", так как все это может быть описано и представлено в едином формализме (унифицированном формате) в базах данных, построенных, например, на основе миварного подхода. Итак, в процессе познания выделяется некоторая вещь в предметной области, а затем присваивается ей некоторое уникальное название. Если требование уникальности не будет выполнено, то это приведет к неоднозначности и к ошибкам. После этого, определяются свойства вещи, на основе которых и произошло ее выделение из предметной области. Каждому свойству присваивается определенное уникальное название (имя, идентификатор). Естественно предположить, что близкие, однородные свойства будут названы похожими, частично совпадающими именами. Затем, изучаются все отношения выделенной вещи с другими вещами данной предметной области. Каждому отношению присваивается уникальное название (идентификатор).

Рассмотрим некоторое множество **V** названий вещей. Так как, каждое название является уникальным, то количество элементов этого множества будет равно количеству выделенных вещей в рассматриваемой предметной области. Затем, некоторые одинаковые свойства могут быть обнаружены у различных вещей. Тогда, можно рассмотреть некоторое множество **S** названий свойств, состоящее из различных и уникальных названий-элементов. Кроме того, существует третье множество **O** названий отношений, состоящее из уникальных названий выявленных отношений вещей данной предметной области.

Итак, можно рассмотреть три различных множества названий: **V**, **S**, **O**. Любая вещь в нашем понимании, т.е. "вещь для нас" - это некоторое название этой вещи, некоторая совокупность ее свойств и некоторая совокупность отношений, в которых эта вещь себя и проявляет. Возникает два вопроса. Можно ли говорить о свойствах вне вещей и отношений? Можно ли говорить об отношениях вне вещей и свойств?

Выше (в разделе 2) и в философской литературе [Л. 246] показано, что в определенных ситуациях вещь может рассматриваться, как свойство или как отношение; свойство может рассматриваться, как вещь либо как отношение и отношение может рассматриваться, как вещь или как свойство. Таким образом, эти три понятия не только взаимосвязаны и взаимозависимы, но и взаимопроникающие, и не-

разрывные. **Вещь не существует вне свойств и вне отношений. Точно также, не существует свойств вне вещей и вне отношений, как не существует и отношений вне вещей и вне свойств.**

Традиционная концепция представления данных оперирует, прежде всего, вещами отдельно, а только потом выделяет свойства для каждой вещи и определяет ее отношения с другими вещами. То есть, необходимо выделить вещи и определить, таким образом, структуру представления данных. Получается, что вещь играет ведущую роль. Но, на самом деле, как было показано и доказано выше (раздел 2), все эти три понятия равноправны, взаимосвязаны и взаимопереходят друг в друга. Как можно разрешить возникшее противоречие? Приведем, казалось бы, отвлеченный вопрос на схожую тему: какое из трех измерений Декартова геометрического пространства  $X$ ,  $Y$  или  $Z$  играет ведущую роль? Ответ прост и понятен: все измерения равноправны, все играют ведущую роль. Более того, вопрос выделения конкретных осей такого пространства является субъективным выбором, надо только соблюдать определенные условия ортогональности. Ведь объективно существует некоторая трехмерная точка, а уж как нам ее отобразить, в каких координатах и на какие оси - это уже дело выбора (вкуса). Просто существует некоторый геометрический объект, который мы описываем с трех разных точек зрения и, например, для определенности, называем эти точки зрения осями координат:  $X$ ,  $Y$  и  $Z$ . Впрочем, всегда можно выбрать другие оси такого пространства.

Основной идеей миварного подхода является то, что **реальный мир существует сам по себе, но при изучении этого мира, в процессе познания, человек (или АСОИ) представляет себе описание этого мира в виде трехмерного пространства, осями которого являются понятия вещи, свойства и отношения.** Другими словами, **понятия вещь, свойство и отношение** - это всего лишь абстракции удобные для описания реального мира. Эти абстракции аналогичны, например, трем осям Декартова геометрического пространства, так как это только три разных взгляда на одно объективно существующее "нечто". Три разных и взаимосвязанных точки зрения на одно и то же "нечто", и позволяют нам выделять из предметной области вещи, свойства и отношения. Эти три абстракции абсолютно равнозначны. В этом и заключается новизна и отличие миварного подхода.

#### 4.1.3. МИВАРНОЕ ТРЕХМЕРНОЕ ИНФОРМАЦИОННОЕ ПРОСТРАНСТВО

Итак, допустим, что все три множества  $V$ ,  $S$  и  $O$  (**вещи, свойства, отношения**) для некоторой конкретной предметной области являются **счетными множествами**. Конечно же, возникает вопрос о границах такой предметной области, но это тема для отдельного исследования. Предположим, что границы такой предметной области достаточно большие, а в некоторых случаях эти границы могут постепенно "расширяться", увеличиваться. Отметим, что опыт научных исследований и практической работы показывает, что таких "больших" и актуальных предметных областей достаточно много. Тогда можно построить **трехмерное дискретное информационное пространство: <вещь, свойство, отношение>**, или, что, то же самое, но более формально: **<V, S, O>**.

Наименьший элемент этого пространства, то есть трехмерную точку, будем называть: **"мивар"**. Выбор этого термина обусловлен соображениями однозначности трактовки и интерпретации. Другими словами, определим мивар следующим образом.

**Определение 4.1.1.** Мивар - это наименьшая адресуемая точка трехмерного пространства **<вещь, свойство, отношение>**.

Подчеркнем, что более никаких других смыслов и интерпретаций в этом слове нет. Конечно, можно сформулировать и другие определения, более конкретные или конструктивные, например, при миварном подходе к описанию вещей с учетом времени и местоположения можно сформулировать другое (более конкретное) определение мивара.

**Определение 4.1.2.** Мивар - это некоторая конкретная вещь, обладающая некоторым конкретным свойством, находящаяся в некотором конкретном отношении в определенный момент времени и в конкретных географических координатах.

Тогда, при таком подходе, вещь - это уникальное название этой вещи, совокупность всех свойств (атрибутов) этой вещи и значения во всех отношениях этой вещи со всеми другими вещами предметной области. Очень важно, что **степень детализации описания вещи может быть различной: от описания всех свойств и отношений до представления лишь сущности вещи.** При этом, существует возможность адаптации данных (повышение адекватности), учета времени, географического расположения, системного уровня, и объектных характеристик при объектно-ориентированном подходе. **"Физический" смысл понятия мивар, с координатами  $\langle v, s, o \rangle$  состоит в том, что это некоторая вещь  $v$ , обладающая некоторым свойством  $s$  и находящаяся в некотором отношении  $o$ .** Подчеркнем, что  $v$  - это элемент счетного множества  $V$  названий вещей;  $s$  - это элемент счетного множества  $S$  названий свойств;  $o$  - это элемент счетного множества  $O$  названий отношений. Каждому мивару, как точке пространства, сопоставляется некоторое значение. Например, в простейшем случае: 0 - "не выявлен" и 1 - "выявлен", или 0 - "не существует" и 1 - "существует". Конечно же, миварам могут соответствовать и более сложные значения, но этот вопрос на данном этапе является второстепенным, требует дополнительных усилий и будет проанализирован в ходе дальнейших исследований. Для наглядности, воспользуемся аналогией с геометрией: **мивар** - это трехмерная точка; **название вещи** - это мивар, принадлежащий оси  $V$ ; **название свойства** - это мивар, принадлежащий оси  $S$ ; **название отношения** - это мивар, принадлежащий оси  $O$ . Тогда, **вещь**, имеющая название  $v$  - это **множество всех миваров**, имеющих некоторые определенные значения и принадлежащих плоскости, которая ортогональна оси  $V$  и проходит через точку-мивар  $\langle v, 0, 0 \rangle$ . Соответственно, **свойство**, имеющее название  $s$  - это **множество всех миваров**, имеющих некоторые определенные значения и принадлежащих плоскости, которая ортогональна оси  $S$  и проходит через мивар  $\langle 0, s, 0 \rangle$ . Наконец, **отношение**, имеющее название  $o$  - это **множество всех миваров**, имеющих некоторые определенные значения и принадлежащих плоскости, которая ортогональна оси  $O$  и проходит через мивар  $\langle 0, 0, o \rangle$ .

Необходимо сделать замечание о том, что следует различать, близкие по написанию, буквы "O" (большое O), "o" (малое o) и цифру 0 (ноль), которые имеют совершенно различную интерпретацию.

Мивар, с координатами  $\langle 0, 0, 0 \rangle$  будем называть **"начало координат миварного пространства"**. Вопрос выбора начала координат имеет важное значение. Возможны различные подходы, например, для описания каждой предметной области использовать некоторый **"относительный центр координат"**. Возможен вариант, когда каждый пользователь, работающий с некоторой областью, вводит свой относительный **"центр мироописания" (представления информации)**, а для общего пользования можно создать некоторую **"энциклопедию" общепризнанных понятий**, от которой и будут отсчитываться начало все пользователи АССОИ. Принципиальным и новым является то, что также как каждый пользователь имеет право

на свой "центр мироздания", точно также у каждого пользователя может быть свой **персональный "горизонт" трехмерного миварного пространства**, что не исключает, а предполагает наличие некоторых общих подпространств, по типу энциклопедий, учебников или каких-либо других. Как известно, "все в мире относительно", следовательно, в этом вопросе, не может быть никаких принципиальных ограничений, кроме разумных соглашений, основанных на здравом смысле. Точно также можно ответить и на вопрос о принципах упорядочивания названий вещей, свойств и отношений в соответствующих множествах. Это будет проблемой самого пользователя познающей (обучающейся) системы, тем более что **миварная система хранения и обработки данных принципиально позволяет в любой момент времени добавить, изменить или уничтожить любой элемент данных**, следовательно, можно и **переупорядочить любое множество**.

Более того, современные подходы к обработке баз данных на основе индексных файлов позволяют одновременно работать с разными ключевыми, индексными файлами ("упорядочиваниями") одного и того же множества. Конечно, все эти возможности будут зависеть от технических характеристик и способностей ЭВМ, а также от вариантов реализаций программных продуктов, основанных на миварной концепции.

#### 4.2. ОСНОВНЫЕ ВОЗМОЖНОСТИ МИВАРНОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ

Прежде всего, миварная концепция представления данных позволяет работать с динамическими (изменяющимися, эволюционными) структурами хранения данных, что открывает новые возможности по созданию эволюционных познающих систем сбора и обработки информации. Миварное представление данных позволяет использовать неявные ассоциативные связи различных понятий и объектов. Это означает, что на основе анализа структуры хранящихся данных можно будет путем **ассоциативного поиска** получать дополнительную информацию, которая не содержится в явном виде в базе данных. Кроме того, в миварном пространстве можно вводить понятие **меры близости - расстояния** либо между отдельными миварами, либо между их совокупностями. Можно использовать **меру подобия**, схожести различных структур. Это открывает принципиально новые возможности для систем, использующих механизм логического вывода или обработку изображений.

Миварное представление данных позволит **обрабатывать неоднородные данные**, организуя связь между ними посредством координат N-мерных точек. Следовательно, подобные по форме представления данные могут храниться в одних наборах данных, а связи между ними будут организованы через общее пространство. Это особенно важно для систем выполняющих комплексную обработку различных по форме представления (мультимедиа, графика, тексты, звук, картографические и т.п.) данных. Кроме того, **правила обработки данных и непосредственно сами данные могут храниться в едином формате**, например, стандартных файлов ЭВМ. Миварное представление включает в себя идеи **объектно-ориентированного** программирования, более того, оно раскрывает **новые возможности и перспективы**. Например, появляется возможность работать с **неполной и противоречивой информацией**, т.е. с различными по степени полноты и достоверности описаниями объектов, причем **в динамике**. В любое время можно либо расширить описание исследуемых объектов, либо уменьшить его. Один и тот же объект в разных отношениях может описываться с **разной степенью детализации**: от одного краткого идентификатора или перечня нескольких наиболее важных

характеристик, до полного описания всех свойств и отношений. Кроме всего прочего, миварная концепция открывает новые возможности по созданию мощных информационных систем, работающих **в реальном масштабе времени без потери динамизма изменения структуры**.

Объединение трехмерного миварного информационного пространства  $\langle V, S, O \rangle$  с трехмерным Декартовым пространством  $\langle X, Y, Z \rangle$  и с учетом времени  $T$ , позволит говорить о едином: **семимерном геометрико-временно-информационном пространстве**, в котором, собственно говоря, и живет любой человек. Такой подход позволяет достаточно просто учитывать время или географические координаты. Причем, все это будет храниться **в едином унифицированном формате**. Это необходимо для систем комплексной обработки информации или для систем, в которых необходимо работать с временными или с геометрическими характеристиками. Миварный подход позволит значительно расширить сферу применения баз данных. Особо отметим, что при необходимости можно **вводить дополнительные оси и размерности**, а это будет уже: **N-мерное миварное представление данных**. Реализация такого подхода вполне возможна даже на современных персональных ЭВМ, не говоря уже о перспективных ЭВМ, в том числе многопроцессорных, и сетях ЭВМ. Миварное пространство может быть разбито на отдельные подпространства, можно и, наоборот, по определенным правилам соединять отдельные представления в более общее. Не исключается возможность создания некоторого глобального, обобщенного пространства описания накопленных человечеством знаний. Причем, принципиальных ограничений для этого не выявлено, остаются лишь вопросы целесообразности и технического решения. Кроме того, миварный подход позволяет проводить **распределенную и параллельную обработку данных**. При этом, миварная концепция представления правил обработки данных позволяет организовать потоковую обработку входной информации на основе **логической сети, управляемой потоком данных**. Важно отметить, что миварная концепция представления данных имеет право на существование и возможны различные пути практической реализации данной концепции.

Миварная концепция появилась как **обобщение и развитие традиционных моделей данных**. С точки зрения **сетевых моделей**, миварное представление - это некоторая сеть, находящаяся в координатном трехмерном пространстве, что только расширяет возможности сетевых моделей. С точки зрения **реляционных моделей**, миварное представление - это трехмерная (или N-мерная) реляционная таблица, находящаяся в трехмерном пространстве, в которой собраны все обычные реляционные таблицы. Дополнительно к этому миварные таблицы представления данных позволяют эффективно хранить правила и отношения между вещами (объектами) типа: "многие к многим" (M:N).

В заключение отметим, что миварная концепция представления данных была получена на основе анализа и обобщения существующих СПД, осмысления и развития философских аспектов проблемы представления информации и на основе выявления потребностей современных познающих и диагностических реальных систем сбора и обработки информации. Подчеркнем, что автором еще в начале 90-х в рамках специальных работ был создан первый прототип - программный продукт (написанный на реляционном языке программирования баз данных Clarion 2.1) для ПЭВМ, названный: "макет миварной базы данных "УЗЕЛОК" версия 2.1". Практическая эксплуатация этого программного комплекса доказывает возможность практической реализации миварной концепции представления данных. Отметим, что миварная концепция представления данных является, в достаточной степени, универсальной, хотя и в строго определенных границах; но, одновременно с этим, вы-

ше доказано и то, что границы эти, по сравнению с традиционным подходом к представлению данных в АССОИ, значительно расширяются.

#### **4.3. ПЕРЕХОД ОТ ОДНОМЕРНЫХ МНОГОУРОВНЕВЫХ ТАБЛИЦ ПРЕДСТАВЛЕНИЯ ДАННЫХ К МНОГОМЕРНОМУ МИВАРНОМУ ПРОСТРАНСТВУ ПРЕДСТАВЛЕНИЯ ДАННЫХ**

Прежде всего, целесообразно проанализировать адекватность моделирования на основе традиционных СПД. Если исходить из того, что база данных является информационной моделью предметной области [Л. 226], то ее необходимо рассматривать как систему математических зависимостей или как программу, которая имитирует структуру или функционирование исследуемого объекта. Основное требование к модели - это ее адекватность объекту, т.е. соответствие или тождественность. Для изучения простых объектов можно применять и простые модели, но для более сложных объектов (предметных областей) необходимо применять более сложные модели. Реляционные, сетевые и иерархические базы данных можно рассматривать как относительно простые модели предметных областей. Семантические сети, бинарные модели и модель данных "сущность-связь" предназначены для моделирования более сложных предметных областей.

Новые области применения баз данных, развитие технических средств, повсеместное внедрение информационно-вычислительных сетей, типа Интернет, а также необходимость повышения адекватности моделирования, привели к тому, что возник целый ряд задач, которые уже не могут быть эффективно решены на основе традиционных баз данных, даже с учетом расширения возможностей современных ЭВМ. Итак, **возникло противоречие**, сущность которого заключается в том, что **базы данных, построенные на основе традиционных моделей данных, не адекватны сложным предметным областям (объектам)**. Следовательно, необходимо разработать принципиально новый подход к информационному моделированию на основе баз данных. Известно, что для моделирования более сложных объектов требуются и более сложные модели. В то же время, для более детального или для более многостороннего изучения объектов, которые считались ранее простыми, также требуются более сложные модели. Сложность моделей данных определяют два основных компонента: СПД (СПД); операции манипулирования данными. Соответственно с этим, можно говорить о **двух направлениях** по созданию более "сложных", более адекватных баз данных:

- 1) **разработка более сложных СПД, обладающих большей адекватностью;**
- 2) **разработка более сложных, многосторонних методов манипулирования данными.**

Путем более сложного манипулирования с простыми по структуре данными можно повысить сложность баз данных. Работы в данном направлении ведутся в настоящее время очень активно. Более перспективным, на наш взгляд, является комбинированный путь: прежде всего - разработка более сложных СПД, а затем применение к ним существующих и создание новых методов манипулирования данными.

Рассмотрим предлагаемые принципиально новые СПД, на основе которых будет создана миварная модель данных. Базы данных, созданные путем реализации миварных принципов организации накопления и хранения данных, т.е. на основе многомерного объектно-ориентированного дискретного информационного пространства унифицированного представления данных о предметной области и правил (процедур

их обработки), позволят значительно повысить адекватность моделирования даже на базе существующих информационно-вычислительных средств. Таким образом, **миварные СПД позволяют разрешить существующее противоречие**: неадекватность традиционных простых СПД и сложных моделируемых объектов - предметных областей. Перейдем к исследованию возможностей и перспектив перехода от одномерных многоуровневых таблиц представления данных, т.е. всех традиционных СПД, к миварному многомерному динамическому эволюционному (адаптивному) пространству унифицированного представления данных и правил.

Выше было показано, что наиболее общей формой представления данных в виде таблиц для традиционных моделей данных (от реляционных и сетевых, вплоть до объектно-ориентированных и модели "сущность- связь") является формализм одномерных таблиц пятиуровневого представления данных (ОТПД-5). Подчеркнем, что в основе традиционного подхода к базам данных лежит выделение из предметной области некоторых объектов, потом осуществляется описание свойств этих объектов, а затем определяются взаимосвязи, отношения между выделенными объектами.

**Основой миварного подхода** к представлению данных является целостное, единое описание изучаемой предметной области с различных сторон, с различных точек зрения. Это осуществляется путем задания, прежде всего, **структуры описания** - выделения основных осей миварного пространства, например: **вещи, свойства, отношения, время и географические координаты**, а возможно и указание некоего системного (**объектно-ориентированного**) месторасположения в терминах метасистемы, системы и подсистемы. Затем, с каждой точки зрения (миварной оси) выделяются совокупности различаемых объектов, каждый из которых получает уникальное краткое наименование (идентификатор), потом производится некоторым образом упорядочение и соотнесение идентификаторов (наименований) объектов с определенными точками определенных координатных осей миварного пространства. Определяются значения выделенных точек, т.е. определяются точки миварного пространства, не принадлежащие его осям, и задаются их значения.

Рассмотрим переход от традиционных реляционных таблиц, т.е. одномерных таблиц трехуровневого представления данных (ОТПД-3), к трехмерному миварному пространству унифицированного представления данных. Как показано выше (в разделе 3.4.1), **ОТПД-3** - это таблица, предназначенная для хранения данных и состоящая из следующих элементов: **заголовок таблицы, заголовки столбцов и клетки таблицы**. Формализованное описание ОТПД-3 на языке теории множеств (аналогично формулам 3.4.1-3.4.3) выглядит следующим образом. Пусть  $A$  - множество имен заголовков таблиц:

$$A = \{a_1, a_2, \dots, a_i, \dots, a_I\}, \quad (4.3.1)$$

где  $i = \overline{1, I}$ . Тогда, для

$$\forall a_i \in A \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ_i}\}, \quad (4.3.2)$$

где  $i = \overline{1, I}$ ;  $j_i = \overline{1, J_i}$  и  $B_i$  - это множество имен заголовков столбцов заголовка таблицы  $a_i$ . Далее, для

$$\forall b_{ij} \in B_i \quad \exists C_{ij} = \{c_{ij1}, c_{ij2}, \dots, c_{ijk_{ij}}, \dots, c_{ijk_{ij_i}}\} \quad (4.3.3)$$

где  $i = \overline{1, I}$ ;  $j_i = \overline{1, J_i}$ ;  $k_{ij} = \overline{1, K_{ij}}$  и  $C_{ij}$  - это множество значений клеток таблицы заголовка столбца  $b_{ij}$  заголовка таблицы  $a_i$ .

В **трехмерном миварном пространстве представления данных (МППД-3)** выделяют элементы следующих четырех типов множеств:

- 1) заголовки таблиц;
- 2) заголовки столбцов;
- 3) заголовки строк;
- 4) клетки таблицы.

При этом, каждым трем элементам из трех различных множеств заголовков может соответствовать только один элемент множества клеток. Рассмотрим **формализованное описание МППД-3**. Пусть существует  $A$  - множество имен заголовков таблиц, т.е.:

$$\exists A = \{a_1, a_2, \dots, a_i, \dots, a_I\}, \quad (4.3.4)$$

где  $i = \overline{1, I}$ . Пусть существует  $B$  - множество имен заголовков столбцов, т.е.:

$$\exists B = \{b_1, b_2, \dots, b_j, \dots, b_J\}, \quad (4.3.5)$$

где  $j = \overline{1, J}$ . Пусть существует  $D$  - множество имен заголовков строк, т.е.:

$$\exists D = \{d_1, d_2, \dots, d_k, \dots, d_K\}, \quad (4.3.6)$$

где  $k = \overline{1, K}$ . Кроме того, пусть существует  $C$  - множество клеток таблиц, т.е.:

$$\exists C = \{c_{111}, c_{112}, \dots, c_{ijk}, \dots, c_{IJK}\}, \quad (4.3.7)$$

где  $i = \overline{1, I}$ ,  $j = \overline{1, J}$ ,  $k = \overline{1, K}$ .

Для любых трех элементов, принадлежащих соответственно множествам  $A$ ,  $B$ ,  $D$  всегда существует единственный элемент, принадлежащий множеству  $C$ , т.е.:

$$\forall a_i \in A, \forall b_j \in B, \forall d_k \in D \quad \exists! c_{ijk} \in C. \quad (4.3.8)$$

где  $i = \overline{1, I}$ ,  $j = \overline{1, J}$ ,  $k = \overline{1, K}$ . И наоборот, для любого элемента множества клеток таблиц  $C$  существует единственный набор трехмерных координат миварного пространства:

$$\forall c_{ijk} \in C \quad \exists! \langle i, j, k \rangle. \quad (4.3.9)$$

где  $i = \overline{1, I}$ ,  $j = \overline{1, J}$ ,  $k = \overline{1, K}$  - трехмерные миварные координаты в пространстве представления данных. Проведем анализ и сравним возможности одномерных таблиц представления данных и миварного многомерного пространства унифицированного представления данных.

#### 4.4. СРАВНЕНИЕ ТРЕХУРОВНЕВЫХ ТАБЛИЦ ПРЕДСТАВЛЕНИЯ ДАННЫХ И ТРЕХМЕРНОГО МИВАРНОГО ПРОСТРАНСТВА ПРЕДСТАВЛЕНИЯ ДАННЫХ

Возьмем некоторую реляционную таблицу. Для любой такой таблицы существует отдельное множество заголовков столбцов - атрибутов описания отношения. При традиционном подходе к реляционным таблицам количество столбцов - атрибутов может быть любым, но это только при проектировании такой таблицы. Как только описание таблицы задано, т.е. зафиксировано некоторое количество атрибутов отношения, после этого **изменять структуру реляционной таблицы**: добавлять, удалять или изменять атрибуты этого отношения **нельзя**. При этом, **наименьшей адресуемой частью реляционной таблицы**, при выполнении операций добавления, удаления или изменения является не отдельная клеточка, а **вся строка**-запись, т.е. все клетки одной строки таблицы. Таким образом, в реляционных таблицах можно лишь накапливать и обрабатывать целые строки-записи таблицы. Следовательно, на самом деле, реально **реляционные таблицы являются одно-**

**мерными**, так как добавление, удаление или изменение производится сразу для всех атрибутов одновременно, т.е. "в длину", а изменение таблицы "в ширину" (изменение структуры таблицы) - запрещено.

Основное отличие МППД-3 от ОТПД-3 заключается в том, что множества заголовков таблиц, столбцов и строк являются независимыми, а, кроме того, появляются элементы четвертого множества - клеток таблицы, в то время как в ОТПД-3 различаются только три типа множеств: заголовки таблиц, заголовки столбцов и строки клеток-записей.

Наименьшей адресуемой частью данных при операциях добавления, удаления и изменения в МППД-3 является отдельная клетка таблицы, связанная с тремя соответствующими заголовками, а в ОТПД-3 - целая строка клеток, которые относятся к одной записи реляционной таблицы. Конечно, в настоящее время многомерные пространства представления данных, в том числе и МППД-3, могут быть построены на основе реляционных таблиц. Однако, возможность построения на основе нескольких различных реляционных таблиц одной многомерной таблицы, не дает никаких оснований для того, чтобы отдельную реляционную таблицу считать многомерной. При построении многоуровневых (например, реляционных) таблиц множества являются, в соответствующем порядке, зависимыми друг от друга. При уничтожении некоторого элемента множества вышестоящего уровня, уничтожаются и все зависимые от него множества. При построении многомерного пространства представления данных, все множества являются независимыми друг от друга, но дополнительно необходимо хранить связи-отношения между элементами различных независимых множеств. В этом одно из главных отличий многоуровневых таблиц от многомерного пространства представления данных.

Проанализируем формулы 4.3.1-4.3.9. Как видно из формул 4.3.2. и 4.3.3, при многоуровневом, т.е. традиционном, реляционном, подходе индексы  $j$  и  $k$  являются зависимыми от индексов более верхнего уровня. При миварном, т.е. многомерном, подходе, как видно из анализа формул 4.3.4-4.3.7, все индексы являются независимыми. Это означает, что размеры, аналогичных реляционным столбцам, миварных СПД могут быть различными, а в реляционных таблицах данных - величины всех столбцов одной таблицы принципиально одинаковы. При практической эксплуатации баз данных, "одинаковость" (равномерность заполнения и хранения данных) всех столбцов таблицы приводит к тому, что, если значения не заданы, то в базе данных все равно хранится запись (пустая) фиксированной длины. Для очень многих предметных областей - это приводит к неоправданной трате вычислительных ресурсов. При миварном подходе реально в базу данных записывается только та информация, которая требуется и имеет заданные значения. Более того, если в некоторых клетках таблиц должна содержаться однотипная информация, то миварный подход позволяет один раз записать и хранить только различные значения таких клеток, а в самих клетках хранить только краткие ссылки (минимальной длины) на соответствующие значения. Допустим, что некоторые трехуровневые одномерные таблицы представления данных ОТПД-3 и трехмерное миварное пространство представления данных МППД-3 описывают одну и ту же предметную область, тогда существуют следующие соотношения между элементами формализованного описания трехуровневых одномерных таблиц представления данных ОТПД-3 (формулы 4.3.1 - 4.3.3) и элементами формализованного описания трехмерного миварного пространства представления данных МППД-3 (формулы 4.3.4 - 4.3.9).

$$\mathbf{J} \leq \sum_{i=1}^{\mathbf{I}} \mathbf{J}_i, \quad (4.4.1)$$

где:  $\mathbf{J}$  - из формулы 4.3.5, а  $\mathbf{j}_i = \overline{\mathbf{1}, \mathbf{J}_i}$  - из формулы 4.3.2.

$$\mathbf{K} \leq \sum_{i=1}^{\mathbf{I}} \sum_{j_i=1}^{\mathbf{J}_i} \mathbf{K}_{i j_i}, \quad (4.4.2)$$

где:  $\mathbf{K}$  - из формулы 4.3.6, а  $\mathbf{k}_{i j_i} = \overline{\mathbf{1}, \mathbf{K}_{i j_i}}$  - из формулы 4.3.3.

$$\forall \mathbf{j} \exists \mathbf{j}_i : \mathbf{b}_j = \mathbf{b}_{i j_i}, \quad (4.4.3)$$

где:  $\mathbf{b}_j \in \mathbf{B}$  (формула 4.3.5),  $\mathbf{b}_{i j_i} \in \mathbf{B}_i$  (формула 4.3.2),  $\mathbf{i} = \overline{\mathbf{1}, \mathbf{I}}$ ,  $\mathbf{j} = \overline{\mathbf{1}, \mathbf{J}}$ ,  $\mathbf{j}_i = \overline{\mathbf{1}, \mathbf{J}_i}$ .

$$\forall \mathbf{i}, \forall \mathbf{j}, \forall \mathbf{k} \exists \mathbf{j}_i, \exists \mathbf{k}_{i j_i} : \mathbf{c}_{i j k} = \mathbf{c}_{i j_i k_{i j_i}}, \quad (4.4.4)$$

где:  $\mathbf{c}_{i j k} \in \mathbf{C}$  (формула 4.3.7),  $\mathbf{c}_{i j_i k_{i j_i}} \in \mathbf{C}_{i j_i}$  (формула 4.3.3),  $\mathbf{i} = \overline{\mathbf{1}, \mathbf{I}}$ ,  $\mathbf{j} = \overline{\mathbf{1}, \mathbf{J}}$ ,  $\mathbf{k} = \overline{\mathbf{1}, \mathbf{K}}$ ,  $\mathbf{j}_i = \overline{\mathbf{1}, \mathbf{J}_i}$ ,  $\mathbf{k}_{i j_i} = \overline{\mathbf{1}, \mathbf{K}_{i j_i}}$ .

$$\mathbf{B} = \bigcup_{i=1}^{\mathbf{I}} \mathbf{B}_i, \quad (4.4.5)$$

где:  $\mathbf{B}$  из формулы 4.3.5,  $\mathbf{B}_i$  из формулы 4.3.2,  $\mathbf{i} = \overline{\mathbf{1}, \mathbf{I}}$ .

$$\mathbf{C} = \bigcup_{i=1}^{\mathbf{I}} \bigcup_{j_i=1}^{\mathbf{J}_i} \mathbf{C}_{i j_i}, \quad (4.4.6)$$

где:  $\mathbf{C}$  - из формулы 4.3.7,  $\mathbf{C}_{i j_i}$  - из формулы 4.3.3,  $\mathbf{i} = \overline{\mathbf{1}, \mathbf{I}}$ ,  $\mathbf{j}_i = \overline{\mathbf{1}, \mathbf{J}_i}$ .

На основании анализа формул 4.4.1 - 4.4.6, можно сделать **вывод** о том, что **трехуровневые одномерные таблицы представления данных ОТПД-3 являются частным случаем более общего представления данных в трехмерном миварном пространстве представления данных МППД-3**. Таким образом, при представлении данных в виде миварного пространства для всех отношений создается единый список атрибутов, что в дальнейшем создает предпосылки для возможности использования структур описания отношений (сущностей, объектов) при манипулировании данными. Например, можно будет сравнивать различные объекты не по значениям атрибутов описания, а просто по наличию таких атрибутов и выявлять похожие или подобные объекты (отношения, сущности).

#### 4.5. СТРУКТУРЫ ПРЕДСТАВЛЕНИЯ ДАННЫХ О МИВАРАХ

При рассмотрении миварного трехмерного дискретного пространства представления данных единичным адресуемым элементом данных является мивар, т.е. трехмерная точка такого пространства с координатами  $\langle \mathbf{i}, \mathbf{j}, \mathbf{k} \rangle$ . Таким образом, **миварное пространство**, с точки зрения теории множеств - это **множество всех трехмерных точек**, задаваемых своими координатами. Все множество миваров можно разделить на следующие пять подмножеств, каждое из которых само является множеством. В зависимости от значения своих координат мивар может принадлежать к следующим множествам:

- 1)  $\langle 0,0,0 \rangle$  - начало координат миварного пространства;
- 2)  $\langle i,0,0 \rangle$  - множество заголовков таблиц;
- 3)  $\langle 0,j,0 \rangle$  - множество заголовков столбцов таблиц;
- 4)  $\langle 0,0,k \rangle$  - множество заголовков строк таблиц;
- 5)  $\langle i,j,k \rangle$  - множество клеток таблицы.

Следовательно, если обозначить мивар " $m(i,j,k)$ ", то:

- мивар  $m(0,0,0)$  - будет соответствовать описанию всего пространства представления данных, например для внутри системных данных;
- множество миваров  $m(i,0,0)$ , при  $i > 0$  - будет соответствовать оси заголовков таблиц, т.е. заголовкам описания двумерных плоскостей (таблиц) представления данных;
- множество миваров  $m(0,j,0)$ , при  $j > 0$  - будет соответствовать оси заголовков столбцов таблиц;
- множество миваров  $m(0,0,k)$ , при  $k > 0$  - будет соответствовать оси заголовков строк таблиц и
- множество остальных миваров  $m(i,j,k)$  - будет соответствовать значениям клеток таблиц, причем сюда же относятся и точки координатных плоскостей, за исключением тех точек, которые принадлежат осям координат.

Применительно к обозначениям формул 4.3.4-4.3.9:

- миварам оси заголовков таблиц соответствуют элементы множества **A**,
- миварам оси заголовков столбцов - элементы множества **B**,
- миварам оси заголовков строк - элементы множества **D**, а
- всем остальным миварам, кроме  $m(0,0,0)$ , соответствуют элементы множества **C**.

В каждой точке миварного пространства может быть определено некоторое **значение**, структура которого сама по себе может быть достаточно **многообразной**: набор символов, некоторый текст, графическое изображение или образ, набор звуков, различные комментарии и т.д. Более того, значений у одного мивара может быть несколько, например: краткое название предмета, некий графический образ этого предмета, детальное описание этого же предмета и цифровая запись некоторых звуков, относящихся к этому предмету. Если некоторому мивару не поставлено в соответствие ни одного значения, то такой мивар существует и считается "неопределенным". Можно специально задавать различные **степени неопределенности**: "значения не может быть", "значение может быть, но оно не известно", "значение уточняется", "значение хранится в другом месте", "значение утрачено" и так далее.

Главное заключается в том, что **миварное пространство:  $M \langle I, J, K \rangle$**  реально является трехмерным в отличие от одномерных многоуровневых реляционных таблиц. Следовательно, миварный подход позволит работать с гораздо более сложными моделями предметных областей, что позволит решить задачу повышения адекватности информационных моделей. На основе анализа формул 4.3.1 - 4.3.9 и 4.4.1 - 4.4.6 можно сделать вывод, что миварные таблицы по сравнению с реляционными являются более общими моделями, обладающими большими возможностями и сложностью по представлению данных.

**В миварных таблицах можно изменять, адаптировать (без перепроектирования) как структуру описания объектов предметной области, так и структуру описания всей предметной области** в целом, причем делать это в рамках одного формализованного описания. Миварные таблицы реализуют все возможности

реляционных таблиц, а, кроме того, они обладают более гибким (адаптивным) и эффективным представлением данных, в том числе, как будет показано ниже, и на физическом уровне. Осуществить физическое хранение миварной таблицы можно различными способами: от одной общей "физической" таблицы и до определенного множества элементарных одномерных (реляционных) таблиц (файлов). Итак, суть миварного представления данных заключается в том, что вводится эволюционное (адаптивное) многомерное дискретное пространство унифицированного представления данных, каждой точке которого может быть присвоено некоторое значение.

Определенные множества элементарных точек пространства, т.е. миваров, будут соответствовать различным описаниям объектов предметной области: от полного описания объектов со всеми комментариями, картино-графическими образами и во времени, а также с учетом объектно-ориентированного (системного) взаимодействия с другими объектами, вплоть до краткого описания (сущности, т.е. наиболее важных свойств и отношений, их значений) таких объектов.

#### 4.6. СТРУКТУРЫ ПРЕДССТАВЛЕНИЯ ДАННЫХ ОБ ОТНОШЕНИЯХ МИВАРОВ

Для представления отношений, связей, взаимодействий различных объектов (вещей, сущностей) предметной области, которые уже описаны в миварном пространстве и которым соответствуют определенные мивары, т.е. для представления отношений миваров вводится отдельное **подпространство представления данных об отношениях миваров**. Напомним, что отношения объектов предметной области сами по себе могут быть представлены как объекты, но это не всегда удобно. Для большей адекватности необходимо ввести отдельное подпространство для представления отношений миваров, что будет являться, в некоторой степени, аналогом сетевых моделей данных. Сразу же подчеркнем, что подпространство миваров и подпространство отношений миваров являются составными элементами общего миварного пространства представления данных о предметной области.

Каждое отношение миваров должно иметь некоторое название. Отношения могут быть разделены по типам, по мерности и т.д. Для одного и того же отношения миваров может существовать несколько различных реализаций, значений. Важно, что для каждого отношения миваров существует множество образов и множество прообразов этого отношения. Во всем остальном, **отношение миваров само может рассматриваться как мивар**, т.е. как точка миварного пространства. Это означает, что отношение миваров может иметь некоторое описание, как и любой мивар. Единственным отличием мивара-точки от мивара-отношения точек заключается в том, что мивару-отношению точек обязательно должны соответствовать определенные типы множеств:

- 1) **множество миваров-образов этого отношения и**
- 2) **множество миваров-прообразов этого отношения.**

Например, для представления наиболее сложного и обобщающего отношения **M:N** достаточно указать уникальное название этого отношения, определить множество миваров-образов, определить множество миваров-прообразов и определить зависимость прообразов от соответствующих образов. В общем случае, можно рассматривать:

- **K-мерное подпространство миваров и**
  - **L-мерное подпространство отношений миваров,**
- которые образуют: **N-мерное миварное пространство**, где  **$K+L=N$** .

Выше было показано, что ОТПД-5 (одномерная таблица пятиуровневого представления данных) является более общим формализмом, чем формализмы СПД реляционных, сетевых, иерархических, бинарных моделей данных, модели данных "сущность-связь" и семантических сетей. На основании формул 2.6.1-2.6.10 доказано, что ОТПД-3 является частным случаем трехмерного миварного пространства представления данных (МППД-3). Таким образом, формализм представления данных в виде трехмерного дискретного пространства является более общим, чем формализм представления данных в виде 3-уровневых таблиц.

Следовательно, **по аналогии** можно сделать вывод о том, что **ОТПД-5 является частным случаем пятимерного миварного пространства представления данных**. Более того, любая N-уровневая таблица представления данных будет лишь частным случаем N-мерного пространства представления данных. Доказательство данного факта можно провести самостоятельно по аналогии с формулами 4.3.1 - 4.3.9 и 4.4.1 - 4.4.6. Перейдем к рассмотрению формализованного описания миварного информационного пространства.

#### 4.7. ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ МИВАРНЫХ СПД

Миварный подход реализуется на практике, прежде всего, путем задания структуры описания данных, т.е. выделением **основных осей миварного пространства: вещи, свойства, отношения**. Кроме того, возможно выделение следующих дополнительных осей: **время, географические координаты, объектно-ориентированное местоположение и системное местоположение**. Подчеркнем, что объектно-ориентированные и системные характеристики могут быть даны в терминах: метасистемы, системы и подсистемы, с учетом наследования свойств, сложности объектов, инкапсуляции и т.д. При необходимости количество дополнительных осей миварного пространства может быть изменено (увеличено или уменьшено) без ограничений. Затем выделяются совокупности различаемых объектов, каждый из которых получает уникальное краткое наименование (идентификатор), потом производится некоторым образом упорядочение и соотнесение наименований объектов с определенными точками конкретных координатных осей миварного пространства. Определяются значения выделенных точек, т.е. определяются точки миварного пространства, не принадлежащие его осям, и задаются их значения. При таком подходе целесообразно сформулировать следующее определение.

**Определение 4.7.1. Мивар - это конкретная вещь, обладающая некоторым конкретным свойством, находящаяся в некотором конкретном отношении** в определенный момент времени и в конкретных географических координатах.

Тогда получаем, что некоторое "нечто", т.е. "вещь" - это уникальное название этой сущности, совокупность всех ее свойств (атрибутов) и ее значения во всех отношениях с другими сущностями исследуемой предметной области. **Степень детализации описания вещи** может быть различной: от полного описания всех свойств и отношений до представления лишь сущности вещи, т.е. минимального набора свойств и отношений, определяющего и выделяющего эту вещь из исследуемой предметной области. Таким образом, формируются не разрозненные различные описания отдельных объектов (что возможно в неструктурированных моделях данных, типа - гипертекст, символьная модель данных), а создается **единая адаптивная, эволюционная динамическая многомерная модель предметной области**, в которой все объекты являются взаимосвязанными и могут рассматри-

ваться в зависимости от требований пользователя: либо как вещи, либо как свойства, либо как отношения. Отметим, что в общем случае, любое **свойство можно рассматривать как унарное** (одноместное) **отношение** некоторой вещи самой к себе. Тогда, имеет смысл рассматривать только два основных вида представления данных: **сущности и отношения**, или: **мивары и отношения миваров**.

Рассмотрим формализованное описание  $N$ -мерного миварного адаптивного пространства унифицированного представления данных. Пусть:

$$\exists A = \{a_1, a_2, \dots, a_n, \dots, a_N\}, \quad (4.7.1)$$

где  $n = \overline{1, N}$ ;  $A$  - множество названий осей миварного пространства,  $N$  - количество осей (возможно динамическое) миварного пространства. Тогда:

$$\forall a_n \in A \quad \exists B_n = \{b_{n1}, b_{n2}, \dots, b_{ni_n}, \dots, b_{ni_n}\}, \quad (4.7.2)$$

$i_n = \overline{1, I_n}$ ;  $B_n$  - множество миваров оси  $a_n$ . Для любых допустимых значений координат всегда существует определенная точка многомерного миварного пространства:

$$\forall i_1, i_2, \dots, i_n, \dots, i_N \quad \exists \langle i_1, i_2, \dots, i_n, \dots, i_N \rangle \in I_1 \times I_2 \times \dots \times I_n \times \dots \times I_N, \quad (4.7.3)$$

где  $\langle i_1, i_2, \dots, i_n, \dots, i_N \rangle$  - координаты  $N$ -мерной точки.

Существует множество значений точек миварного пространства:

$$\exists C = \{c_{i_1 i_2 \dots i_n \dots i_N} \mid i_1 = \overline{1, I_1}; \dots; i_n = \overline{1, I_n}; \dots; i_N = \overline{1, I_N}\}. \quad (4.7.4)$$

Для каждой точки миварного пространства существует единственное значение из множества значений  $C$ :

$$\forall \langle i_1, i_2, \dots, i_n, \dots, i_N \rangle \quad \exists! c_{i_1, i_2, \dots, i_n, \dots, i_N} \in C. \quad (4.7.5)$$

Конкретные значения точек миварного пространства, размерность миварного пространства и структуры конкретных таблиц представления данных зависят от моделируемых предметных областей, а также от возможностей программного и аппаратного обеспечения. Теоретически никаких ограничений не накладывается. Более того, размерности различных таблиц в одном пространстве могут различаться, а также предоставляется возможность для каждого пользователя на основе единой общей БД создавать свои индивидуальные представления и образы предметной области.

#### 4.8. ПРИМЕР ОПИСАНИЯ ДАННЫХ В МИВАРНОМ ПРОСТРАНСТВЕ ПРЕДСТАВЛЕНИЯ ДАННЫХ И ПРАВИЛ

Для иллюстрации различных концепций в моделях данных в [Л. 316, с. 124-126, 318-320] используется специальный пример описания медицинских данных. Для того, чтобы не описывать новый пример представления данных для каждой из перечисленных выше моделей данных (что займет очень много места в книге) представляется целесообразным использовать в качестве образца уже готовый пример из литературы [Л. 316, с. 124-126, 318-320] и дополнительно показать только реализацию этого примера на основе миварного подхода к представлению данных.

Впрочем, для удобства читателей очень кратко приведем здесь основные положения данного примера описания данных. Итак, в качестве примера рассматривается ряд сведений по некоторым аспектам функционирования больницы [Л. 316]. Конечно же, это упрощенная версия описания реальной предметной области. Для обеспечения полной однозначности интерпретации дадим краткое описание нашего тестового примера, согласно приложению 1 в книге [Л. 316, с. 318]. В таблице 4.8.1

представлены типы сущностей и их атрибуты, а в таблице 4.8.2 показаны типы связей данного тестового примера. Более подробное описание данного примера (таблицы 4.8.3-4.8.8) приведено в [Л. 316, с. 124-126, 318-320]. Отметим, что в связи с ограниченным объемом данной работы в дальнейшем мы будем приводить не полные данные из этого примера, а только наиболее информативные, показывающие главные особенности миварного подхода. Однако, при желании читатель может и сам построить подобные примеры, либо дополнить этот пример новыми значениями и записями, либо увеличить объем рассматриваемых данных.

Таблица 4.8.1.

Тип сущности	Атрибуты
Больница	Код больницы, Название, Адрес, Телефон, Число коек
Палата	Код палаты, Название, Число коек
Персонал	Номер служащего, Фамилия, Должность, Смена, Зарплата
Врач	Номер врача, Фамилия, Специальность
Пациент	Регистрационный номер, Номер койки, Фамилия, Адрес, Дата рождения, Пол, НСС (номер социального страхования)
Диагноз	Код диагноза, Тип диагноза, Осложнения, Предупреждающая информация
Лаборатория	Номер лаборатории, Название, Адрес, Телефон
Анализ	Код анализа, Назначенная дата, Назначенное время, Номер варианта/направления, Состояние

Таблица 4.8.2.

Тип связи	Типы сущностей	Отображение
Больничная палата	Больница, палата	функциональное
Штатный врач	Больница, врач	функциональное
Больница - лаборатория	Больница, лаборатория	многие - ко - многим
Персонал палаты	Палата, персонал	функциональное
Размещение	Палата, пациент	функциональное
Пациент - врач	Врач, пациент	многие - ко - многим
Анализ пациента	Пациент, анализ	функциональное
Диагноз пациента	Пациент, диагноз	функциональное
Направленный анализ	Лаборатория, анализ	функциональное

Рассмотрим краткое описание этого примера (табл. 4.8.3 - 4.8.8).

Таблица 4.8.3.

Больница				
Код	Название	Адрес	Телефон	Число коек
13	Центральная	Шербурн 333	964-4264	502
18	Общая	Колледж 101	595-3111	987
22	Клиническая	Брунsvик 45	923-5411	412
45	Детская	Университетская 555	597-1500	845

Таблица 4.8.4.

Лаборатория			
Номер	Название	Адрес	Телефон
16	Эткон	Мэйн 14	532-4453
42	Клини	Кинг 55	447-6448
56	Альфа	Киплинг 18	929-9611
84	Нукро	Лайонс 62	368-9703

Таблица 4.8.5.

<b>Врач</b>		
Номер врача	Фамилия	Специальность
435	Ли	Кардиолог
585	Миллер	Гинеколог
982	Расс	Кардиолог
386	Стоун	Психиатр
398	Бест	Уролог
453	Глэсс	Педиатр
522	Адамс	Невропатолог
607	Эшби	Педиатр

Таблица 4.8.6.

<b>Больничные палаты</b>		
Код палаты	Название	Число коек
3	Интенсивной терапии	21
6	Психиатрическая	67
3	Интенсивной терапии	10
4	Кардиологическая	53
1	Реанимационная	10
2	Родильная	34
6	Психиатрическая	118
1	Реанимационная	13
2	Родильная	24
4	Кардиологическая	55

Таблица 4.8.7.

<b>Пациент</b>					
Регистрационный номер	Фамилия	Адрес	Дата рождения	Пол	НСС
63827	Рэски	Бэтхорст 60	01.06.1945	М	100973253
36658	Домб	Патина 55	08.04.1954	М	660657471
64823	Фрэзер	Мэсси 11	03.05.1960	Ж	985201776
74835	Бауэр	Онтарио 15	16.10.1933	М	654811767
18004	Шу	Айви 14	22.01.1916	Ж	914991452
59076	Миллер	Лоутон 80	04.06.1971	Ж	611969044
24024	Фурье	Донора 40	09.07.1966	Ж	321790059
10995	Листа	Олсен 58	07.11.1963	М	980862482
39217	Байрз	Даллас 51	20.08.1958	М	740294390
38702	Нил	Хэлси 65	03.11.1949	Ж	380010217

Для реляционного представления данных примера [Л. 316, с. 124-126], кроме приведенных в таблицах 4.8.2 - 4.8.8 таблиц, еще необходимо хранить следующие реляционные таблицы сущностей (таблица 4.8.9), подробное описание которых опускается для сокращения объема работы. Кроме того, для накопления информации о связях сущностей необходимо хранить данные о следующих отношениях (таблица 4.8.10), которые также здесь подробно не описываются, так как они полностью соответствуют примеру в [Л. 316, с. 124-126, 318-320].

Таблица 4.8.8.

<b>Персонал</b>				
Номер служащего	Фамилия	Должность	Смена	Зарплата
3106	Хьюджес	Санитар	Д	13500
3754	Дэлэджи	Сиделка	Д	17400
6357	Карплюс	Стажер	У	18300
6065	Ритчи	Сиделка	В	20200
7379	Колони	Сиделка	У	16300
9901	Ньюпорт	Стажер	У	17000
1009	Холмс	Сиделка	У	18500
8422	Белл	Санитар	У	12600
8526	Франк	Сиделка	Д	19400
1280	Андерсон	Стажер	В	17000

Таблица 4.8.9.

Название таблицы	Атрибуты
<b>Диагноз</b>	<b>Код диагноза, Тип диагноза, Осложнения, предупреждающая информация</b>
<b>Анализ</b>	<b>Код анализа, Тип, Назначенная дата, Назначенное время, Номер варианта/ направления, состояние</b>
<b>Врач - пациент</b>	<b>Номер врача, Регистрационный номер</b>
<b>Больница - лаборатория</b>	<b>Код больницы, Номер лаборатории</b>

Таблица 4.8.10.

Название отношения	Таблица 1	Таблица 2
Больничные палаты	Больница	Палата
Персонал палаты	Палата	Персонал
Размещение	Палата	Пациент
Штатный врач	Больница	Врач
Пациенты врача	Врач	Врач-пациент
Диагноз пациента	Пациент	Диагноз
Назначенные анализы	Пациент	Анализ
Направленные анализы	Лаборатория	Анализ
Используемые лаборатории	Больница	Больница-лаборатория
Обслуживаемые больницы	Лаборатория	Больница-лаборатория

Для рассмотрения возможностей, особенностей и преимуществ миварного подхода представления данных, приведем конкретный пример описания базы медицинских данных в миварном трехмерном пространстве, аналогичный указанному выше примеру, который Цикритзис Д. и Лоховски Ф. показали в своей книге [Л. 316, с. 124-126, 318-320].

#### 4.8.1. ОПРЕДЕЛЕНИЕ ОСЕЙ МИВАРНОГО ПРОСТРАНСТВА

Прежде всего, определим оси, т.е. типы заголовков, трехмерного миварного пространства. Заголовки будут следующих трех типов: заголовки таблиц, заголовки столбцов и заголовки строк. В исходном примере, как показано выше, всего 8 таблиц [Л. 316, с. 124]. При этом, заголовки таблиц будут соответствовать сущностям или

отношениям, заголовки столбцов - атрибутам, а заголовки строк - номерам конкретных значений. Итак, получаем следующие 8 заголовков таблиц (**ТБЛ**):

1. **ТБЛ1** больница;
2. **ТБЛ2** палата;
3. **ТБЛ3** персонал;
4. **ТБЛ4** врач;
5. **ТБЛ5** пациент;
6. **ТБЛ6** диагноз;
7. **ТБЛ7** лаборатория;
8. **ТБЛ8** анализ.

Заголовки столбцов (**СЛБ**), по количеству различных атрибутов описания сущностей (всего 23), можно представить в следующем виде:

1. **СЛБ1** код;
2. **СЛБ2** название;
3. **СЛБ3** адрес;
4. **СЛБ4** телефон;
5. **СЛБ5** число коек;
6. **СЛБ6** фамилия;
7. **СЛБ7** номер;
8. **СЛБ8** должность;
9. **СЛБ9** смена;
10. **СЛБ10** зарплата;
11. **СЛБ11** специальность;
12. **СЛБ12** номер койки;
13. **СЛБ13** дата рождения;
14. **СЛБ14** пол;
15. **СЛБ15** номер социального страхования;
16. **СЛБ16** тип диагноза;
17. **СЛБ17** осложнения;
18. **СЛБ18** предупреждение;
19. **СЛБ19** тип анализа;
20. **СЛБ20** назначенная дата;
21. **СЛБ21** назначенное время;
22. **СЛБ22** номер направления;
23. **СЛБ23** состояние.

Перейдем к определению СПД.

#### 4.8.2. ОПРЕДЕЛЕНИЕ СТРУКТУР ПРЕДСТАВЛЕНИЯ ДАННЫХ

Пусть все мивары будут одного типа. Определим в двумерной плоскости "заголовки таблиц, заголовки столбцов" точки пересечения, т.е. определим структуру медицинских данных, по которой будут накапливаться данные. Для краткости, будем обозначать:

- ось заголовков таблиц - **ТБЛ** =  $\mathbf{M}(\mathbf{x}, \mathbf{0}, \mathbf{0}) = \mathbf{M}(\mathbf{x})$ ;
- ось заголовков столбцов - **СЛБ** =  $\mathbf{M}(\mathbf{0}, \mathbf{y}, \mathbf{0}) = \mathbf{M}(\mathbf{y})$ ;
- ось заголовков строк - **СТР** =  $\mathbf{M}(\mathbf{x}, \mathbf{y}, \mathbf{0}) = \mathbf{M}(\mathbf{x}, \mathbf{y})$ ;
- мивары - клетки таблицы  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  или  $\mathbf{M}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ .

Отсутствие третьей координаты при обозначении заголовков строк необходимо только для удобства графического представления структуры данных. На самом деле, у всех заголовков строк  $\mathbf{M}(\mathbf{x}, \mathbf{y})$  значение по третьей координате ( $\mathbf{z}$ ) просто

равно нулю, т.е., в некотором смысле:  $СТР = M(x, y) = M(x, y, 0)$ . И только поэтому третью координату можно и не указывать на рисунках. Конечно же, аналогично вместо знака СЛБ можно указывать значение  $M(y)$  только с одной переменной, так как все остальные значения по следующим двум координатам  $(x, z)$  у заголовков столбцов в данном случае будут равны нулю, т.е., в некотором смысле:  $СЛБ = M(0, y, 0) = M(y)$ .

Аналогично можно объяснить и появление записи:  $ТБЛ = M(x, 0, 0) = M(x)$ . Однако, при появлении конкретных цифр обозначения ТБЛ и СЛБ через  $M(\text{цифра})$  не будут уникальными, поэтому они в дальнейшем и не используются, а указывается конкретный идентификатор ТБЛ или СЛБ. Тогда, в двумерной плоскости  $\langle ТБЛ, СЛБ \rangle$  получим 37 следующих клеток, соответствующих заголовкам строк  $M(x, y)$ , ортогонально которым и будут накапливаться все данные в соответствующих клетках (строках) трехмерной таблицы.

1. **M(01,01)** код больницы;
2. **M(01,02)** название больницы;
3. **M(01,03)** адрес больницы;
4. **M(01,04)** телефон больницы;
5. **M(01,05)** число коек в больнице;
6. **M(02,01)** код палаты;
7. **M(02,02)** название палаты;
8. **M(02,05)** число коек в палате;
9. **M(03,06)** фамилия служащего палаты;
10. **M(03,07)** номер служащего палаты;
11. **M(03,08)** должность служащего палаты;
12. **M(03,09)** смена служащего палаты;
13. **M(03,10)** зарплата служащего палаты;
14. **M(04,06)** фамилия врача;
15. **M(04,07)** номер врача;
16. **M(04,11)** специальность врача;
17. **M(05,03)** адрес пациента;
18. **M(05,06)** фамилия пациента;
19. **M(05,07)** регистрационный номер пациента;
20. **M(05,12)** номер койки пациента;
21. **M(05,13)** дата рождения пациента;
22. **M(05,14)** пол пациента;
23. **M(05,15)** номер социального страхования пациента;
24. **M(06,01)** код диагноза;
25. **M(06,16)** тип диагноза;
26. **M(06,17)** осложнения;
27. **M(06,18)** предупреждающая информация;
28. **M(07,02)** название лаборатории;
29. **M(07,03)** адрес лаборатории;
30. **M(07,04)** телефон лаборатории;
31. **M(07,07)** номер лаборатории;
32. **M(08,01)** код анализа;
33. **M(08,19)** тип анализа;
34. **M(08,20)** назначенная дата анализа;
35. **M(08,21)** назначенное время анализа;
36. **M(08,22)** номер варианта / направления;
37. **M(08,23)** состояние.

Указанные двумерные точки  $M(x, y)$  можно изобразить на рисунке (см. рис. 4.8.1). Подчеркнем, что в данном случае, третье измерение: заголовки строк - это номера соответствующих значений строк (миваров), которые на рис. 4.8.1 уходят вглубь страницы, т.е. ортогональны плоскости страницы и поэтому не показаны. Кроме того, у всех миваров на рис. 4.8.1 значение 3-й координаты равно 0 и поэтому, для наглядности, не указано.

Отметим, что на рисунке 4.8.1. отражены только первые значения-заголовки таких строк в формате: **M(координата ТБЛ, координата СЛБ)**.

Данные могут храниться в таких строках в разных форматах, например, в виде набора последовательных данных (рис. 4.8.2). Для каждого такого файла тип хранимых данных указан в названии файла. Возможно и другое представление данных в строках, когда заголовок столбца одномерной таблицы формата  $M(x, y)$  определяет двумерные координаты заголовка строки накопления данных, а графа "номер" в каждой таблице фактически определяет значение третьей координаты точек по оси заголовков строк (рис 4.8.3).

Например, на рис 4.8.3 в первой таблице  $M(1, 1)$  на 3 (третьей) строке записано значение "22", которое соответствует трехмерной точке с координатами (1, 1, 3). Отметим, что мивары обозначаются указанием трехмерной точки  $(x, y, z)$  или буквой "M" с указанием в скобках всех трех значений координат по осям заголовков таблиц, столбцов и строк.

Далее, рассмотрим пример формирования таблицы "Больница" (рис. 4.8.4). На рисунке 4.8.4 в скобках  $(x, y, z)$  приведены координаты элементов структуры в миварном трехмерном пространстве представления данных. Для сравнения миварной таблицы с реляционными таблицами можно посмотреть таблицу "Больница". При графо-табличном представлении эта миварная таблица (рис. 4.8.4) будет иметь вид, показанный в таблице 4.8.3. Также, приведем пример таблицы "Лаборатория" [Л. 316] (таблица 4.8.4) в миварном (рис. 4.8.5) виде. Составим таблицу описания связей, в которой укажем порядковый номер связи, название связи, образ связи, прообраз связи и тип связи (рис. 4.8.6). Рассмотрим фрагмент миварного трехмерного описания медицинских данных представленного выше примера (рис. 4.8.7) и пример описания связей в миварном пространстве.

Если каждой связи присвоить некоторый универсальный префикс, например - VEC, а также ее уникальный номер, то получим уникальные идентификаторы связей, например: VEC01, VEC02, ..., VEC09. При задании образа и прообраза целесообразно делать так.

1. Если в качестве образа (прообраза) задается принадлежащая оси координат точка - заголовок таблицы, то образом (прообразом) является все множество всех точек, которые относятся к этой таблице, т.е. образом (прообразом) является вся таблица.

2. Если в качестве образа (прообраза) задается точка, соответствующая определенному заголовку (столбца или строки) определенной таблицы (т.е. точка лежащая в одной из координатных плоскостей), то в таком случае образом (прообразом) является множество всех точек (миваров), которые образуют значение столбца данной таблицы, т.е. по аналогии можно назвать это: "луч", ортогональный данной точке-образу (прообразу).

СЛБ									
23									M(8,23)
22									M(8,22)
21									M(8,21)
20									M(8,20)
19									M(8,19)
18						M(6,18)			
17						M(6,17)			
16						M(6,16)			
15					M(5,15)				
14					M(5,14)				
13					M(5,13)				
12					M(5,12)				
11				M(4,11)					
10			M(3,10)						
9			M(3,9)						
8			M(3,8)						
7			M(3,7)	M(4,7)	M(5,7)			M(7,7)	
6			M(3,6)	M(4,6)	M(5,6)				
5	M(1,5)	M(2,5)							
4	M(1,4)							M(7,4)	
3	M(1,3)				M(5,3)			M(7,3)	
2	M(1,2)	M(2,2)						M(7,2)	
1	M(1,1)	M(2,1)				M(6,1)			M(8,1)
0	1	2	3	4	5	6	7	8	ТБЛ

Рис. 4.8.1. Двумерная схема миварного накопления данных.

1. M(01,01) код больницы 

13	18	22	45	...
----	----	----	----	-----
2. M(01,02) название больницы 

центральная	общая	клиническая	детская	...
-------------	-------	-------------	---------	-----
3. M(01,03) адрес больницы 

Шербурн333	Колледж101	Брунсвик45	Университетская555	...
------------	------------	------------	--------------------	-----
- ...
6. M(02,01) код палаты 

3	6	3	4	1	2	6	1	2	4	...
---	---	---	---	---	---	---	---	---	---	-----
- ...
8. M(02,05) число коек в палате 

21	67	10	53	10	34	118	13	24	55	...
----	----	----	----	----	----	-----	----	----	----	-----
- ...
29. M(07,03) адрес лаборатории 

Мэйн 14	Кинг 55	Киплинг 18	Лайонс 62	...
---------	---------	------------	-----------	-----

Рис. 4.8.2. Пример представления данных в строках таблицы.

M(01,01) код больницы		M(01,02) название больницы		M(01,03) адрес больницы		...
Номер	Значение	Номер	Значение	Номер	Значение	
1	13	1	Центральная	1	Шербурн 333	
2	18	2	Общая	2	Колледж 101	
3	22	3	Клиническая	3	Брунsvик 45	
4	45	4	Детская	4	Университетская 555	
...	...	...	...	...	...	

Рис. 4.8.3. Пример представления данных в строках таблицы.

ТБЛ1 (1,0,0)	СЛБ1 (0,1,0)	СЛБ2 (0,2,0)	СЛБ3 (0,3,0)	СЛБ4 (0,4,0)	СЛБ5 (0,5,0)
Больница	Код	Название	Адрес	Телефон	Число коек
(1,0,1) Значение 1	(1,1,1) 13	(1,2,1) Центральная	(1,3,1) Шербурн 333	(1,4,1) 964-4264	(1,5,1) 502
(1,0,2) Значение 2	(1,1,2) 18	(1,2,2) Общая	(1,3,2) Колледж 101	(1,4,2) 595-3111	(1,5,2) 987
(1,0,3) Значение 3	(1,1,3) 22	(1,2,3) Клиническая	(1,3,3) Брунsvик 45	(1,4,3) 923-5411	(1,5,3) 412
(1,0,4) Значение 4	(1,1,4) 45	(1,2,4) Детская	(1,3,4) Университетская 555	(1,4,4) 597-1500	(1,5,4) 845
...	...	...	...	...	...

Рис. 4.8.4. Пример миварного представления таблицы "Больница".

ТБЛ7 (7,0,0)	СЛБ7 (0,7,0)	СЛБ2 (0,2,0)	СЛБ3 (0,3,0)	СЛБ4 (0,4,0)
Лаборатория	Номер	Название	Адрес	Телефон
(7,0,1) Значение 1	(7,7,1) 16	(7,2,1) Эткон	(7,3,1) Мэйн 14	(7,4,1) 532-4453
(7,0,2) Значение 2	(7,7,2) 42	(7,2,2) Клини	(7,3,2) Кинг 55	(7,4,2) 447-6448
(7,0,3) Значение 3	(7,7,3) 56	(7,2,3) Альфа	(7,3,3) Киплинг 18	(7,4,3) 929-9611
(7,0,4) Значение 4	(7,7,4) 84	(7,2,4) Нукро	(7,3,4) Лайонс 62	(7,4,4) 368-9703
...	...	...	...	...

Рис. 4.8.5. Пример миварной таблицы "Лаборатория".

Номер	Название	Образ	Прообраз	Тип связи
1	Больничная палата	Больница	Палата	функциональная
2	Штатный врач	Больница	Врач	функциональная
3	Больница-лаборатория	Больница	Лаборатория	многие - ко - многим
4	Персонал палаты	Палата	Персонал	функциональная
5	Размещение	Палата	Пациент	функциональная
6	Пациент - Врач	Врач	Пациент	многие - ко - многим
7	Анализ пациента	Пациент	Анализ	функциональная
8	Диагноз пациента	Пациент	Диагноз	функциональная
9	Направленный анализ	Лаборатория	Анализ	функциональная
...	...	...	...	...

Рис. 4.8.6. Фрагмент таблицы описания связей.

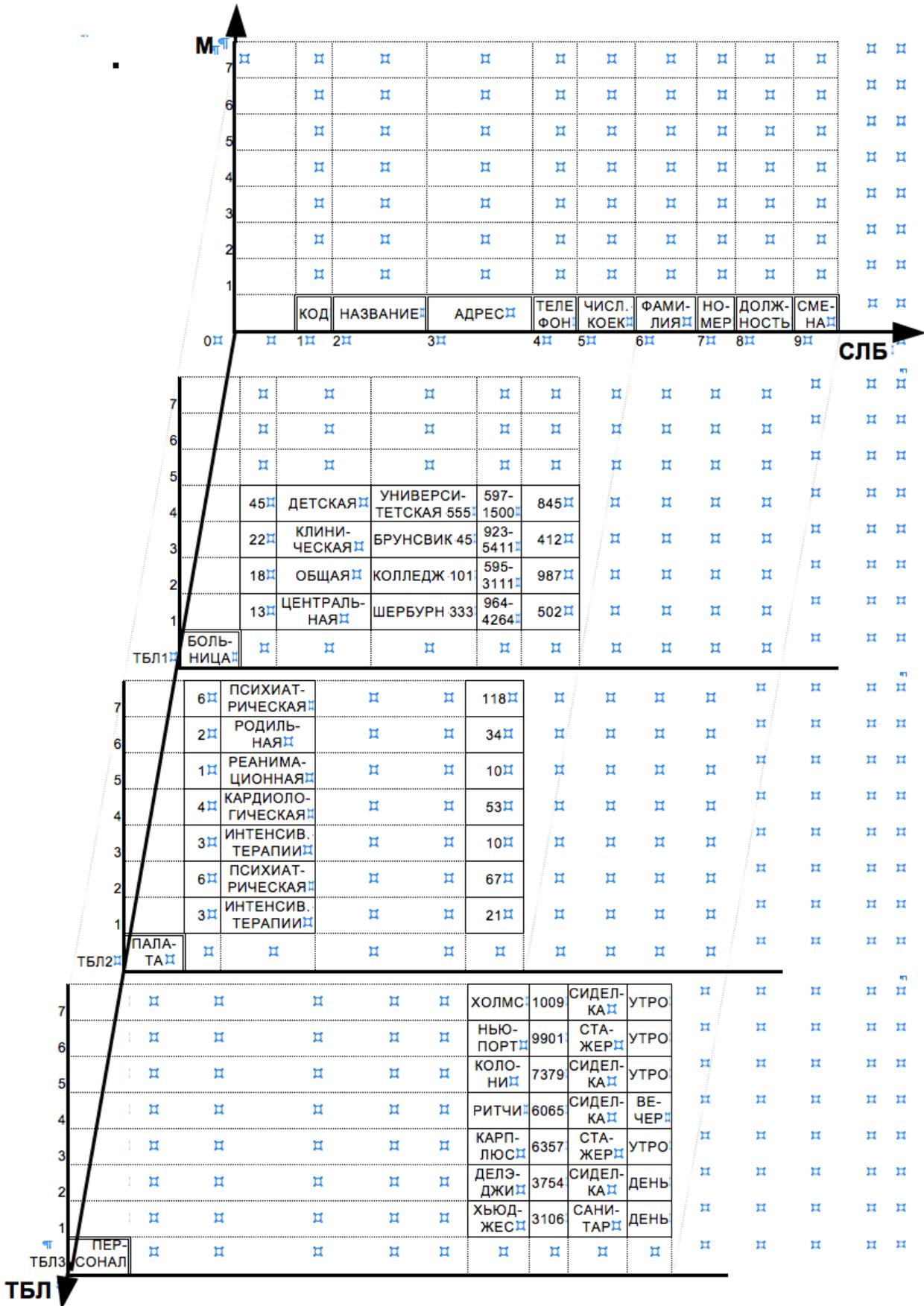


Рис. 4.8.7. Фрагмент миварного 3-мерного описания медицинских данных.

3. Если в качестве образа (прообраза) задается точка не принадлежащая ни одной из координатных плоскостей (т.е. все ее координаты отличны от 0), то образом (прообразом) является множество, состоящее только из одной этой точки.

Тогда, для однозначного описания конкретной связи представляется целесообразным рассмотреть таблицу, в которой определены как связи, так и каждая конкретная реализация связи. Для этого необходимо задать: уникальный номер связи, идентификатор связи, значение связи (реализация), точка-образ (т.е. если трехмерное пространство -это три координаты мивара), реализация образа связи (три координаты), прообраз связи (три координаты), значение реализации прообраза связи (три координаты).

Приведем пример представления связей базы медицинских данных [Л. 316] в миварном трехмерном пространстве (рис. 4.8.8). Отметим, что для задания значений конкретной связи необходимо всего три элемента: идентификатор связи и его конкретная реализация; идентификаторы образа и прообраза с их значениями.

Номер реализации связи	Идентификатор связи	Реализация связи	Идентификатор образа	Значение образа	Идентификатор прообраза	Значение прообраза
1	VEC01	1	(1,0,0)	(1,1,1)	(2,0,0)	(2,1,1)
2	VEC01	2	(1,0,0)	(1,1,1)	(2,0,0)	(2,1,2)
3	VEC01	3	(1,0,0)	(1,1,2)	(2,0,0)	(2,1,3)
4	VEC01	4	(1,0,0)	(1,1,2)	(2,0,0)	(2,1,4)
5	VEC01	5	(1,0,0)	(1,1,3)	(2,0,0)	(2,1,5)
6	VEC01	6	(1,0,0)	(1,1,3)	(2,0,0)	(2,1,6)
7	VEC01	7	(1,0,0)	(1,1,3)	(2,0,0)	(2,1,7)
8	VEC01	8	(1,0,0)	(1,1,4)	(2,0,0)	(2,1,8)
9	VEC01	9	(1,0,0)	(1,1,4)	(2,0,0)	(2,1,9)
10	VEC01	10	(1,0,0)	(1,1,4)	(2,0,0)	(2,1,10)
...	...	...	...	...	...	...
19	VEC03	1	(1,0,0)	(1,0,1)	(7,0,0)	(7,0,1)
20	VEC03	2	(1,0,0)	(1,0,1)	(7,0,0)	(7,0,2)
21	VEC03	3	(1,0,0)	(1,0,2)	(7,0,0)	(7,0,1)
22	VEC03	4	(1,0,0)	(1,0,2)	(7,0,0)	(7,0,2)
23	VEC03	5	(1,0,0)	(1,0,2)	(7,0,0)	(7,0,3)
24	VEC03	6	(1,0,0)	(1,0,2)	(7,0,0)	(7,0,4)
25	VEC03	7	(1,0,0)	(1,0,3)	(7,0,0)	(7,0,3)
26	VEC03	8	(1,0,0)	(1,0,3)	(7,0,0)	(7,0,4)
27	VEC03	9	(1,0,0)	(1,0,4)	(7,0,0)	(7,0,1)
...	...	...	...	...	...	...
38	VEC05	1	(2,0,0)	(2,0,1)	(5,0,0)	(5,0,1)
39	VEC05	2	(2,0,0)	(2,0,2)	(5,0,0)	(5,0,2)
40	VEC05	3	(2,0,0)	(2,0,3)	(5,0,0)	(5,0,3)
41	VEC05	4	(2,0,0)	(2,0,4)	(5,0,0)	(5,0,4)
42	VEC05	5	(2,0,0)	(2,0,5)	(5,0,0)	(5,0,5)
43	VEC05	6	(2,0,0)	(2,0,6)	(5,0,0)	(5,0,6)
44	VEC05	7	(2,0,0)	(2,0,7)	(5,0,0)	(5,0,7)
45	VEC05	8	(2,0,0)	(2,0,8)	(5,0,0)	(5,0,8)
46	VEC05	9	(2,0,0)	(2,0,9)	(5,0,0)	(5,0,9)
47	VEC05	10	(2,0,0)	(2,0,10)	(5,0,0)	(5,0,10)
...	...	...	...	...	...	...

Рис. 4.8.8. Таблица представления связей медицинских данных.

### 4.8.3. ОПРЕДЕЛЕНИЕ ПРОСТРАНСТВА ДАННЫХ

Таким образом, как было показано выше, практически, представляется возможным работать с трехмерным миварным пространством представления данных:

<ТБЛ, СЛБ, СТР>.

Первая ось **ТБЛ** - ось заголовков таблиц, вторая ось **СЛБ** - ось заголовков столбцов; третья ось **СТР** - ось заголовков строк. Отметим, что именно ось **СТР** не показана на рис. 4.8.1. В данном конкретном случае, заголовки строк - это просто номера значений, соответствующих элементов двумерной схемы накопления данных. Единицей измерения на каждой оси является одна запись данных (клетка таблицы), т.е. **клетка - наименьший адресуемый элемент накапливаемых данных**. Следовательно, с учетом того, что в соответствии с примером, наибольшее количество записей в описании пациента: 10 записей [Л. 316] получим следующее количество адресуемых элементов данных:

- по оси **ТБЛ** будет 8 единиц,
- по оси **СЛБ** будет 23 единицы, а
- по оси **СТР** - будет 10 единиц данных .

После рассмотрения этого конкретного примера представления данных в миварном пространстве данных, целесообразно проанализировать эволюционные, адаптивные возможности миварного пространства представления данных.

### 4.9. ЭВОЛЮЦИОННОСТЬ МИВАРНЫХ СПД

Отметим, что в миварном пространстве могут изменяться не только значения переменных, но и количество осей пространства, т.е. сама структура представления данных. Таким образом, предлагается рассмотреть **новый класс моделей данных с изменяемой (эволюционной, адаптивной) структурой представления данных**.

Динамические эволюционные (адаптивные) многомерные миварные СПД являются наиболее унифицированными (универсальными) структурами представления данных.

Подчеркнем, что любые "плоские" графо-табличные многоуровневые СПД являются частным случаем многомерных СПД. Таким образом, **основы построения миварного пространства представления данных** заключаются в следующем. Формируется минимальное пространство представления данных (ППД) путем определения основных осей и фиксации основных объектов и их отношений. Названия объектов и отношений в некотором порядке фиксируются на соответствующих осях пространства представления данных, а на соответствующих пересечениях точках ППД, называемых миварами, записываются конкретные значения свойств и отношений объектов.

В дальнейшем при появлении новых данных миварное пространство представления данных либо просто накапливает данные при неизменной структуре. Либо происходит изменение СПД путем добавления, изменения или удаления любого отдельного мивара, любой точки на любой оси пространства или даже самой оси пространства представления данных, конечно же, с соблюдением определенных процедур поддержания целостности данных.

Подчеркнем, что для различных случаев, в зависимости от технических средств, программного обеспечения и т.п., могут применяться различные варианты физического хранения миварных баз данных и правил. Важной особенностью миварного подхода является то, что отношения объектов (правила обработки)

хранятся в том же едином миварном пространстве и могут представлять собой отношения, правила или процедуры обработки объектов, которые могут функционировать при выполнении некоторых условий. Таким образом, **миварный подход позволяет хранить в едином пространстве представления данных и сами данные, и правила их обработки - отношения объектов.**

Отметим, что любое отношение также может быть добавлено, удалено или изменено в любое время без остановки обработки, но с соблюдением определенной процедуры поддержания целостности адаптивного миварного пространства представления данных.

Таким образом, миварное пространство предоставляет возможность эволюционного многомерного структурированного (при необходимости - объектно-ориентированного) накопления, обработки и изменения любых данных и правил без перепланирования баз данных. Следовательно, именно миварное пространство обеспечивает эволюционность развития любых автоматизированных систем обработки информации или, более широко - любых интеллектуальных многопроцессорных систем, т.е. любых компьютерных систем.

Перейдем к описанию обработки данных и к исследованию новых возможностей по обработке данных в миварном пространстве представления данных и правил.

## 5. ОБРАБОТКА ДАННЫХ В МИВАРНОМ ИНФОРМАЦИОННОМ ПРОСТРАНСТВЕ

### 5.1. ОБРАБОТКА ДАННЫХ НА ОСНОВЕ АДАПТИВНОЙ ЛОГИЧЕСКОЙ СЕТИ ПРАВИЛ

#### 5.1.1. ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ В МИВАРНЫХ МОДЕЛЯХ ДАННЫХ

После анализа возможностей и особенностей миварного пространства по представлению данных необходимо рассмотреть оставшиеся компоненты модели данных: ограничения целостности и операции над данными [Л. 15, 19, 23, 27-28, 51, 71, 84, 87, 91, 100-104, 110, 120-124, 139-142, 175-176, 183, 190, 206, 209, 213, 216, 226, 247, 263, 269-271, 284, 293, 305, 314-316, 319, 326-327, 339, 349-356, 362-368, 372-383, 387-390, 392, 396-403, 407-413, 492-501]. Рассмотрим сначала **ограничения целостности** миварной модели данных. Как было показано выше, миварная модель данных является развитием и обобщением практически всех традиционных, прежде всего графо-табличных, моделей данных. На данном этапе исследования целесообразно считать, что все основные ограничения целостности, которые характеризуют традиционные модели данных, в том числе и наиболее общую, "выразительную" модель данных "сущность-связь" [Л. 19, 28, 87, 91, 100-104, 121, 123, 183, 190, 216, 263, 284, 305, 314-316, 319, 326, 368, 396-403], могут быть применены (реализованы) и для миварной модели данных.

В случае необходимости, все дополнительные и основные ограничения целостности могут быть включены либо в отношения, либо в свойства (характеристики, атрибуты) любого мивара или множества миваров. Более того, если в графо-табличных моделях нельзя накладывать ограничения целостности на часть описания объекта, на часть описания атрибутов и т.п. [Л. 19, 28, 87, 91, 100-104, 121, 123, 183, 190, 216, 263, 284, 305, 314-316, 319, 396-403], то к особенностям ограничений целостности миварных моделей можно отнести то, что миварные структуры представления данных, как было показано выше, достаточно просто и очень органично позволяют характеризовать (описывать, моделировать), а, следовательно, и накладывать ограничения целостности на любые отдельные мивары, любые совокупности миваров, любые множества миваров и на отношения миваров.

Таким образом, возможности по ограничению целостности в миварной модели данных позволяют реализовать все традиционные ограничения целостности, а также дополнительно реализовать более конкретные, "адресные" ограничения целостности, путем задания специальных "отношений по ограничению целостности".

#### 5.1.2. ОПЕРАЦИИ НАД ДАННЫМИ В МИВАРНЫХ МОДЕЛЯХ ДАННЫХ

После анализа возможностей и особенностей ограничений целостности, рассмотрим оставшуюся компоненту миварной модели данных, а именно **операции над данными** [Л. 226, 316, 319]. Прежде всего, поступим по аналогии с ограничениями целостности, изложенными выше. Так как миварная модель данных является обобщением и развитием традиционных моделей данных, прежде всего модели данных "сущность-связь", то все операции над данными [Л. 19, 28, 87, 91, 100-104, 121, 123, 183, 190, 216, 263, 284, 305, 314-316, 319, 326, 368, 396-403], ко-

торые есть в модели "сущность - связь", могут быть аналогично (возможно с небольшими модификациями) реализованы и в миварной модели данных. Целесообразно в дальнейшем уделить основное внимание на особенности реализации традиционных операций и на новые операции над данными, которые присущи только миварной модели данных, а рассмотрение реализации традиционных операций над данными провести по аналогии с тем, как эти операции реализованы в графо-табличных моделях данных, прежде всего в модели "сущность - связь".

Итак, рассмотрим новые операции над данными, а также особенности обработки данных в миварном адаптивном многомерном динамическом объектно-ориентированном пространстве унифицированного представления данных и правил. Прежде всего, проанализируем особенности представления правил (процедур, отношений) обработки данных в миварном пространстве.

### 5.1.3. СПИСОЧНОЕ ПРЕДСТАВЛЕНИЕ АДАПТИВНОЙ ЛОГИЧЕСКОЙ СЕТИ ПРАВИЛ

Миварное пространство унифицированного представления данных и правил существенно расширяет возможности по адаптивности, универсальности и оперативности представления и обработки информации в автоматизированных системах обработки информации, к которым относятся и системы адаптивного синтеза информационно-вычислительных конфигураций - САС ИВК. В тоже время, миварное пространство позволяет реализовать адаптивную обработку информации. Следовательно, целесообразно проанализировать предлагаемый новый метод обработки данных в миварном пространстве представления данных и правил на основе построения активной адаптивной логической сети правил, управляемой потоком данных.

Существо предлагаемого метода в том, что **в миварном пространстве унифицированного представления данных из всех возможных правил (отношений) и заданных (известных) значений формируется логическая сеть вывода (обработки), управляемая потоком данных.** Оперативность (быстродействие) нового метода определяется тем, что сначала определяют маршрут логического вывода на основе специального набора "метаправил", а только потом осуществляют запуск необходимых процедур и правил обработки данных.

Такой подход позволяет значительно сократить время обработки, так как непосредственно на обработку не запускаются лишние и не нужные для получения вывода правила. Кроме того, в зависимости от имеющихся значений переменных и правил обработки (отношений), подграфы обработки-вывода могут динамически изменяться. При этом возможно формирование новых правил (процедур) обработки миварного пространства представления данных и правил, существенно сокращающих процесс вывода.

Все правила в миварном пространстве могут быть пронумерованы единым списком правил. Все переменные, входящие в состав различных правил, также образуют некоторый единый список переменных. Для каждого правила выявляются, определяются входные и выходные переменные. Если одно и тоже правило используется для получения различных переменных, в зависимости от потока входных данных, например - обычные уравнения тригонометрии и т.п., то существует несколько возможностей по представлению таких правил. Но самым простым и традиционным является разбиение такого правила в зависимости от набора входных данных на несколько простейших, "однозначных" процедур - правил обработки входных данных.

Тогда априорно, т.е. изначально, вся логическая сеть может быть представлена в виде трех взаимосвязанных списков: входных переменных, правил и выходных переменных. Такое представление логической сети позволяет в любой момент времени добавить, изменить или удалить любые переменные или правила, т.е. такая логическая сеть становится адаптивной. В этих списках переменных для каждой переменной определяется номер или идентификатор соответствующего правила. Аналогично, для каждого правила соответственно определяются все входные и выходные переменные. Таким образом, получаем простой граф взаимосвязей всех переменных и правил. Отметим, что именно табличная или, другими словами, "списочная" форма представления такого графа позволяет в любое время изменить его состав и конфигурацию.

#### **5.1.4. АКТИВНАЯ ОБРАБОТКА ПОТОКА ВХОДНЫХ ПЕРЕМЕННЫХ АДАПТИВНОЙ ЛОГИЧЕСКОЙ СЕТИ**

Проанализируем возможности и особенности активной обработки потока входных данных (переменных) адаптивной логической сети. В определенный момент времени начинается логический вывод или обработка входного потока переменных. Пусть на вход нашей логической сети в некоторый момент времени поступили значения нескольких переменных. Тогда происходит присваивание значений, т.е. "означивание" этих переменных, а затем - происходит поиск таких правил обработки, для которых заданы ("означены") все входные переменные. Подчеркнем, что, таким образом, эта логическая сеть (и ей подобные) управляется потоком входных данных, т.к. весь дальнейший вывод ведется только на основе потока известных значений входных переменных.

Если таких правил, для которых "означены" все входные переменные, нет, то обработка на этом завершается и формируется сигнал о том, что не определено достаточное количество переменных и требуется дополнительная информация. При этом существует возможность выделить такие правила, для полного "означивания", т.е. "запуска", которых не хватает минимального количества переменных. Такие переменные могут быть отсортированы по количеству правил, в которых они встречаются, а затем в этом ранжированном порядке список переменных выдается подсистеме сбора информации для уточнения и определения значений требуемых переменных. Отметим, что формирование таких списков "потребностей" является признаком "активности" системы обработки, т.е. такая автоматизированная система самостоятельно, но по определенным правилам, определяет и ранжирует важность необходимых входных данных.

Если есть такие правила, у которых заданы все входные переменные, то происходит их обработка ("запуск"). Обработка правил может быть одновременной, параллельной, если для этого достаточно вычислительных ресурсов. В противном случае, необходимо выбрать те правила, которые будут запущены в первую очередь.

Предлагается следующий порядок определения приоритета правил. Прежде всего, выделяют все правила, в результате обработки которых будут известны требуемые (искомые) переменные, и называют их "последними". Чем больше искомых переменных определяет правило, тем выше его приоритет обработки. В первую очередь обрабатывают наиболее приоритетные правила, в результате выполнения которых может быть получено требуемое решение. Если ни одно из таких "последних" правил не может быть запущено, то определяют те переменные, которые необходимы для запуска "последних" правил. Ранжирование таких переменных производится аналогично приведенному выше, т.е. чем больше правил

требует эту входную переменную, тем выше ее приоритет. Затем, подобным образом, определяют "предпоследние" правила, т.е. те, которые "означивают" (вычисляют) требуемые переменные. Аналогичные действия повторяют, при необходимости, несколько раз. В тоже время, исходя от известных переменных и выбранных "первых" правил, формируют список обработки правил 1-го этапа. При этом если общее количество этапов обработки принять за N, то все "последние" правила образуют этап обработки № N.

Среди правил 1-го этапа, если все вышесказанное не выполнено и ответ не получен, наибольший приоритет присваивается тем правилам, в результате обработки которых получается максимальное количество новых означенных переменных. Однако, в некоторых случаях, когда заранее известно требуемое время обработки каждого правила, приоритетность может быть рассчитана относительно минимального времени обработки, т.е. запускать в первую очередь надо самые быстрые правила. Конкретные критерии приоритетности определяются априорно исходя из заданных условий, специфики решаемой задачи и особенностей моделируемой, исследуемой предметной области.

Получаем, что с двух противоположных сторон, т.е. от известных и от искомых переменных, выстраивается - формируется некоторая многоэтапная структура обработки правил. Определение очередности обработки правил, возможности их запуска и подсчет вычисляемых каждым правилом переменных производят без выполнения самих правил. При этом, исходят только из того, что если у правила заданы все входные переменные, то могут быть однозначно определены и все выходные переменные. Отметим особо, что время и ресурсы обработки каждого правила могут существенно различаться друг от друга. Следовательно, время определения (вычисления) этапов обработки и очередности запуска правил не зависит от самих правил и существенно меньше времени обработки самих правил. В том случае, если у нескольких правил совпадают все рассчитанные параметры, то более высокий приоритет получает правило с меньшим первоначальным номером в списке всех правил. В процессе определения порядка и очередности выполнения правил может быть выявлена такая ситуация, когда "известных" и "вычисленных" переменных не достаточно для определения искомых переменных, тогда обработка завершается без искомого результата. Аналогично указанному выше, сообщается о невозможности получения искомого результата и определяют наиболее важные для завершения обработки переменные, перечень которых выдается в качестве запроса для добавления и уточнения входных данных. Если же подсистема обработки вычисляет, что результат может быть получен, то осуществляется собственно запуск машины логического вывода с учетом "вычисленного" порядка обработки. Таким образом, исходя из начального списочного формирования сети, "вычисляется" и формируется логическая сеть обработки данных.

#### **5.1.5. АДАПТИВНОСТЬ ЛОГИЧЕСКОЙ ОБРАБОТКИ ПОТОКА ВХОДНЫХ ПЕРЕМЕННЫХ**

В процессе функционирования (обработки) логической сети могут быть получены новые исходные данные: новые значения уже известных переменных или значения новых переменных. В случае появления новых значений уже известных переменных, фиксируется время их появления и, при необходимости, осуществляется новый запуск всей цепочки логического вывода. При этом, одновременно могут выполняться, "вычисляться" разные значения одних и тех же переменных, в зависимости от времени начала их обработки, что должно корректно учитываться.

В случае появления значений новых переменных процесс вывода, при необходимости, приостанавливается и, исходя из известных переменных, к числу которых добавляются значения новых переменных и уже вычисленные переменные, осуществляется новое планирование логического вывода. После определения нового маршрута вывода процесс обработки запускается.

Подчеркнем, что на одном наборе правил может осуществляться одновременно, но с соблюдением специальных мер поддержания корректности вывода, несколько выводов, в зависимости от времени поступления значений переменных. Кроме того, определение и вычисление маршрутов логического вывода, что является "метавычислениями" для системы, также могут одновременно осуществляться на общей вычислительной системе. Получаем, что, сформировав набор правил и "метаправил", в зависимости от подаваемых на вход системы значений входных переменных, "запущена" такая машина логического вывода, которая затем может функционировать не только в автоматизированном режиме, но и полностью автономно, т.е. без вмешательства человека.

Миварный подход позволяет без перепланировки вводить новые правила и новые переменные. Для сформированной системы логического вывода такие изменения также возможны. Фиксируется момент времени ввода новых правил и /или переменных, что необходимо для поддержания целостности вывода предшествующих значений переменных. Затем новые правила и переменные вводятся в соответствующие списки, и процессы формирования (вычисления) маршрутов вывода с этого момента, просто работают с новыми списками без всяких дополнительных усилий и указаний. Модернизация и удаление правил и переменных также возможны, но в зависимости от особенностей предметной области уже (и все еще) вычисляемые процессы обработки должны быть либо запущены заново, либо должны быть завершены на основе старых данных. Таким образом, получаем реальную адаптивную логическую сеть обработки, управляемую потоком входных данных.

Полученная адаптивная логическая сеть, кроме всего прочего, самостоятельно (по заданным алгоритмам, но автоматически) формирует запросы на уточнение и дополнение входных данных. Следовательно, такая сеть обладает свойством активности. В некоторых случаях, после построения подсистемы сбора информации и после задания, определения или формирования правил и переменных подсистемы обработки, полученная система может функционировать без вмешательства человека или других внешних воздействий, если все подсистемы будут корректно взаимодействовать друг с другом. Кроме того, могут быть заданы такие "метаправила", которые позволят создаваемой системе самостоятельно формировать новые правила обработки данных. В отличие от известных традиционных систем логического вывода, в которых вывод (обработка данных) происходит только по запросу, создаваемая логическая сеть постоянно обрабатывает (все) новые входные переменные.

Для однозначности дальнейшего наименования будем **называть создаваемую активную адаптивную логическую сеть, управляемую потоком данных, "миварной логической сетью"**. Такое название является вполне естественным, если учесть, что в ее основе лежит миварное пространство унифицированного представления данных и правил. Подчеркнем, что такую или аналогичную сеть можно создать и без использования миварного подхода к представлению данных и правил. В таком случае это будет менее эффективно, чем при использовании миварного подхода, но также (что важно) будет действовать и на основе существующих баз данных и систем логического вывода. Таким образом, активную

адаптивную логическую сеть представляется возможным создавать и эксплуатировать как на основе миварной концепции представления данных, так и независимо от нее на основе существующих подходов к обработке данных.

## 5.2. ОСОБЕННОСТИ ОБРАБОТКИ ДАННЫХ НА ОСНОВЕ МИВАРНОЙ ЛОГИЧЕСКОЙ СЕТИ

Прежде всего, рассмотрим особенности диагностики и распознавания на основе миварной логической сети. Если на основе миварной логической сети создать некоторую информационную (логическую) модель предметной области и затем постоянно подавать на вход поток данных, то такая миварная сеть самостоятельно будет вести всю обработку, определяя постоянно новые значения всех переменных. Целесообразно для некоторых предметных областей задать некоторое количество "диагностических" правил обработки результатов моделирования, которые в случае перехода заданных переменных из одной области значений в другую, выполняло бы заданные действия. Например, подавала бы сигнал тревоги или просто посылала сообщение пользователю о новых состояниях модели. В таком случае, получаем логичное **совмещение распознавания и диагностики** в одной автоматизированной системе. Кроме того, на одной миварной логической сети может одновременно функционировать несколько различных информационных моделей, результаты работы которых целесообразно сравнивать, а также выбирать из них наиболее адекватные по итогам экспериментальной проверки.

Теперь обсудим **эволюционность** и обучаемость миварной логической сети. Возможности по изменению, адаптации описания предметной области, т.е. правил и данных, придают миварной сети способности к ее обучению. Режимы обучения могут быть различными, но главное, что миварная логическая сеть и все АССОИ, созданные на ее основе, являются обучаемыми, эволюционными. Так как на само миварное пространство не налагается ни каких теоретических ограничений по накоплению и обработке данных, следовательно, и обучаемость, и эволюционность миварной логической сети ничем не ограничена, т.е. она может быть глобальной. Вопросы самообучения миварных систем могут быть реализованы в автономном режиме, но требуют дополнительного исследования, являются предметом дальнейших исследований, а, следовательно, выходят за рамки данной работы. Адаптивная логическая сеть обработки, управляемая потоком данных, объединяет и обработку известных данных на основе известных правил, и одновременно осуществляет обучение, изменение системы правил со своевременным внесением изменений в соответствующие процессы обработки, что обеспечивает эволюционное, синергетическое развитие всей компьютерной системы в целом. Кроме того, такая активная логическая сеть позволяет максимально распараллелить и ускорить обработку любого потока данных, а также повысить эффективность использования технических средств.

Для повышения оперативности обработки данных на миварной логической сети правил представляется возможным и целесообразным разработать метод "графового" поиска маршрута вывода на логической сети правил, который предлагается рассмотреть в следующем разделе.

### 5.3. ПРИНЦИПЫ "ГРАФОВОГО" ПОИСКА МАРШРУТА ВЫВОДА НА ЛОГИЧЕСКОЙ СЕТИ ПРАВИЛ

#### 5.3.1. ЛОГИЧЕСКИЕ СЕТИ ПРАВИЛ И МНОГОПОЛЮСНЫЕ СЕТИ ТЕОРИИ ГРАФОВ

Разработка метода "графового" поиска маршрута логического вывода основана на использовании результатов теории графов в области построения многополюсных сетей и поиска минимального разреза (или максимального потока). Суть предлагаемого метода состоит в преобразовании логической сети правил миварного пространства в многополюсную сеть теории графов и поиска ее минимального разреза. Подчеркнем, что предлагаемый метод может применяться не только для миварных логических сетей, но и для любых традиционных логических сетей, возможно с учетом небольших модификаций и конкретизации условий его применения.

Данный метод поиска маршрута логического вывода называется "графовым", потому что для поиска этого маршрута строится граф обработки, с одной стороны от исходных, известных значений, а с другой стороны - от требуемых, искомых значений. Затем осуществляется анализ связности этого графа путем поиска его минимального разреза.

Если минимальный разрез больше или равен единице, то решение существует и определяется кратчайший, минимальный путь достижения требуемых результатов, а затем на этой основе непосредственно запускается механизм вывода и обработки данных, что позволит значительно экономить вычислительные ресурсы.

Если минимальный разрез равен нулю, то цепочки вывода нет, но можно определить значения каких именно переменных (объектов) являются определяющими (важными и критическими) для последующего уточняющего запроса пользователю, что и является признаком "активности" подсистемы обработки данных.

Внедрение этого метода, в том числе и в системы адаптивного синтеза информационно-вычислительных конфигураций, которые будут более подробно рассмотрены ниже, позволит значительно экономить время и вычислительные ресурсы, активно формулировать уточняющие запросы и определять недостающие значения переменных. Особо подчеркнем, что предлагаемый способ "графового" (быстрого) поиска маршрута логического вывода является достаточно универсальным и, возможно при небольшой модификации, может применяться для любых логических систем или машин логического вывода, в том числе и для всех известных, традиционных систем продукционного типа, а также для сетей Петри и им подобным.

В настоящее время известно достаточно много методов, методик и способов поиска маршрута вывода на логической сети. В основном, эти традиционные работы [Л. 9, 11, 23, 28, 31, 33, 65, 66, 92, 97-98, 103, 117, 123, 155, 171, 173, 176, 194, 219, 232, 248, 251, 263, 270, 276, 280, 292, 310, 314, 326, 348, 358, 409] исследуют различные способы организации или выбора кратчайшего пути логического вывода, например, различные способы поиска на различных деревьях вывода: вывод "вглубь", "в ширину", бинарные деревья и т.д. При этом, как правило, все логические правила обработки реально "запускаются", обрабатываются в некотором специально заданном порядке [Л. 19, 28, 87, 91, 100-104, 121, 123, 183, 190, 216, 263, 284, 319, 326, 396-403]. Известно, что логическая обработка занимает гораздо больше вычислительных ресурсов и времени, чем обычные арифметические операции. Отметим, что практически все логические системы или машины вывода

могут быть представлены в виде графов. В то же время, в теории графов разработано достаточно много методов и способов анализа связности графа или поиска изолированных подграфов [Л. 41-42, 55-56, 146, 173, 308, 310, 416-426]. Если логическую систему вывода представить в виде некоторого графа, то задача поиска кратчайшего маршрута логического вывода может быть представлена в следующем виде. Возьмем за основу построения (моделирования) **многоуровневый (многоярусный) многотиповой динамический граф**. Прежде всего, аналогично теории сетей Петри, выделим два типа узлов графа: вершины-переменные и вершины-процедуры обработки. Взаимосвязи между ними будут представлять собой направленные вектора - ребра графа. Если переменная является входной для процедуры, вектор направлен от переменной к процедуре. Если переменная является выходной для процедуры, т.е. получается в результате выполнения процедуры, то вектор направлен от процедуры к переменной. Допустим, что выполняется следующее условие: каждая логическая процедура может быть готова к выполнению тогда и только тогда, когда известны все ее входные переменные. Если процедура выполнена, то все ее выходные переменные получают некоторые значения.

Логическую систему вывода представим в виде графа, в котором выделим группу входных переменных и группу искомым переменных. Кроме этих двух групп, в общем случае, еще существуют выводимые переменные, т.е. те переменные, значения которых могут быть определены (вычислены, выведены), и невыводимые переменные, аналогично, т.е. те, которые не могут быть вычислены, определены в ходе выполнения логической обработки. Если путь логического вывода, от входных до искомым переменных, есть, значит, как минимум, эти переменные принадлежат некоторому связанному подграфу. Если граф является связанным, значит его минимальный разрез больше либо равен единице. Следовательно, для **определения существования маршрута логического вывода на таком графе достаточно просто найти его минимальный разрез**. Если минимальный разрез больше либо равен единице, значит маршрут логического вывода существует. Для определения непосредственно маршрута вывода необходимо просто найти кратчайший путь от группы-подграфа известных переменных к группе-подграфу искомым переменных.

На основании графа можно построить многополюсную сеть. Если в таком графе выделены некоторые переменные, то их называют полюсами многополюсной сети, а сам граф образует многополюсную сеть. Подчеркнем, что термины "сеть" в теории графов и в математической логике имеют различное, но похожее значение. В дальнейшем необходимо четко определять контекст и понимать значение этого термина - "сеть".

Перейдем к анализу алгоритмов поиска минимального разреза многополюсных сетей.

### **5.3.2. АЛГОРИТМЫ ПОИСКА МИНИМАЛЬНОГО РАЗРЕЗА МНОГОПОЛЮСНЫХ СЕТЕЙ ТЕОРИИ ГРАФОВ**

В теории графов известно достаточно много способов и алгоритмов поиска минимального разреза многополюсных сетей (или максимального потока) различной вычислительной сложности [Л. 41-42, 55-56, 146, 173, 308, 310, 416-426]. Таким образом, задача поиска маршрута логического вывода может быть сведена в теории графов к поиску минимального разреза многополюсной сети и поиску кратчайшего пути от одной группы вершин графа к другой.

Отметим, что решение задачи теории графов непосредственно не позволят найти конкретные значения искомым логическим переменным, так как для этого необходимо осуществить логический вывод. Но решение этих задач позволит однозначно доказать существование или отсутствие маршрута логического вывода, а также определить непосредственный маршрут искомого логического вывода. Конечно, решение задач теории графов может быть отнесено к дополнительным затратам времени и вычислительного ресурса, но, как уже отмечалось выше, эти задачи имеют строгое, математически доказанное, решение за определенный интервал времени, т.е., другими словами, эти задачи имеют некоторую вычислительную сложность. Решение таких задач связано, как правило, с целочисленными вычислениями типа элементарного сложения на различных матрицах, что очень эффективно выполняется на любых традиционных ЭВМ. Кроме того, в отличие от логических задач, алгоритмы поиска кратчайшего пути и минимального разреза теории графов имеют хорошо определенную вычислительную сложность, а значит - гарантированное время решения таких задач, которое может быть быстро вычислено, определено заранее.

Известно, что вычислительная сложность алгоритмов поиска минимального разреза многополюсных сетей определяется относительно общего количества узлов многополюсной сети, причем такая зависимость количества операций от узлов сети, как правило, является степенной [Л. 41-42, 55-56, 146, 173, 308, 310, 416-426]. В теории графов известно достаточно много различных методов, алгоритмов и способов поиска минимального разреза двухполюсных и многополюсных сетей кубической вычислительной сложности. Наиболее быстрым из известных алгоритмов [Л. 41-42, 55-56, 146, 173, 308, 310, 416-426] поиска минимального разреза двухполюсных сетей считается алгоритм авторов Карив-Арбиб, который характеризуется вычислительной сложностью  $2,5$  (или, что то же самое:  $O(n^{2,5})$ , но это просто более формализованное обозначение). Однако, в ходе проведенных автором научных исследований, с учетом особенностей моделирования адаптивных миварных логических сетей правил, **разработан новый квадратичной сложности (т.е.  $O(n^2)$  сложности) метод поиска минимального разреза многополюсных сетей по вершинам и ребрам ее графа [Л. 41-42]**, который более подробно будет рассмотрен после анализа возможности преобразования логической сети правил миварного пространства в многополюсную сеть теории графов.

### 5.3.3. ПРЕОБРАЗОВАНИЕ ЛОГИЧЕСКОЙ СЕТИ ПРАВИЛ МИВАРНОГО ПРОСТРАНСТВА В МНОГОПОЛЮСНУЮ СЕТЬ

Преобразование логической сети правил миварного пространства в многополюсную сеть теории графов с целью поиска ее минимального разреза осуществляется следующим образом. В многополюсной сети выделяют два типа узлов: переменные и процедуры. Логические переменные миварного пространства преобразуются в узлы-переменные многополюсной сети.

Логические правила и процедуры (отношения) миварного пространства преобразуются в узлы-процедуры многополюсной сети. Логические взаимосвязи переменных и процедур миварного пространства преобразуются в вектора или направленные ребра многополюсной сети. Ребра многополюсной сети могут связывать только узлы разных типов: "переменные" и "процедуры", а связи - ребра между однотипными вершинами запрещены по определению. Таким образом, формируют **"двудольный ориентированный граф"**, в некотором смысле похожий на сети Петри [Л. 173, 228, 308, 310, 416-426]. Затем выделяют две группы

узлов-переменных: "входные" и "искомые" переменные. Построение многополюсной сети начинают с группы входных переменных, которые образуют собственно полюса этой сети. Искомые переменные образуют другую группу полюсов этой сети. От "входных" полюсов осуществляют постепенное построение многополюсной сети до "искомых" полюсов.

Выделив все "входные" переменные на первом этапе построения многополюсной сети, на втором этапе осуществляют отбор таких узлов-процедур, для которых известны, т.е. являются "входными", все их "входные" переменные. Таким образом, отбирают такие процедуры, которые могут быть "запущены", выполнены. Подчеркнем, что выполнение логических процедур не производится. После завершения второго этапа, на следующем этапе определяют все узлы-переменные, которые могут быть определены, вычислены или "выведены" после выполнения всех процедур предыдущего этапа.

На следующем этапе из всех "означенных" узлов-переменных, т.е. "входных" и "вычисленных" переменных, формируют множество "известных" переменных. Осуществляют проверку признака завершения построения многополюсной сети. Если все "искомые" узлы-переменные вошли во множество "известных" переменных, значит построение многополюсной сети можно завершить, а все оставшиеся узлы (переменные и процедуры) могут быть размещены в произвольном порядке, так как они уже не оказывают никакого влияния на вывод "искомых" переменных. Другим признаком завершения построения многополюсной сети будет такой, когда для построения многополюсной сети использованы все узлы и все ребра, т.е. построение полностью завершено. Если построение многополюсной сети не завершено, то на следующем этапе на основе множества всех "известных" узлов-переменных осуществляют отбор таких новых узлов-процедур, для которых известны все их "входные" переменные, т.е. отбирают такие процедуры, которые могут быть "запущены", выполнены на следующем этапе построения многополюсной сети.

На следующем этапе, аналогично вышеизложенному, определяют все узлы-переменные, которые могут быть определены, вычислены или "выведены" после выполнения всех новых процедур предыдущего этапа. Далее, на следующем этапе все новые "означенные", "выведенные" или "вычисленные" узлы-переменные добавляют к множеству "известных" переменных. Аналогично осуществляют проверку признака завершения построения многополюсной сети. Если построение многополюсной сети не завершено, то аналогично указанным выше этапам циклически проводят дальнейшее формирование многополюсной сети до полного завершения.

В результате формирования многополюсной сети представляется целесообразным представить ее в виде некоторой матрицы. На сформированной многополюсной сети известными методами теории графов осуществляется поиск ее минимального разреза между "входными" и "искомыми" узлами-переменными. Так как сформированный граф является ориентированным, то поиск минимального разреза должен осуществляться с учетом направленности всех ребер многополюсной сети таким образом, чтобы от "входных" переменных можно было вывести все "искомые" значения. Поиск на двудольном ориентированном графе многополюсной сети является частным случаем поиска минимального разреза многополюсной сети по ребрам и вершинам ее графа. Если минимальный разрез больше или равен единице, то переходят к поиску кратчайшего маршрута логического вывода. Задача поиска кратчайшего пути в многополюсных сетях так же является хорошо изученной в теории графов. Отметим, что некоторые методы по-

иска минимального разреза многополюсной сети фактически сразу определяют и кратчайший путь в такой многополюсной сети, что сокращает необходимый объем вычислений и предоставляет возможность непосредственного определения кратчайшего, оптимального пути логического вывода одновременно с поиском минимального разреза многополюсной сети. Рассмотрим конкретный пример реализации метода "графового" поиска маршрута вывода путем построения адаптивного механизма логического вывода на конкретной реализации миварной сети правил в виде активной эволюционной сети гиперправил с мультиактивизаторами, управляемой потоком данных.

#### **5.4. АДАПТИВНЫЙ МЕХАНИЗМ ЛОГИЧЕСКОГО ВЫВОДА НА ЭВОЛЮЦИОННОЙ АКТИВНОЙ СЕТИ ГИПЕРПРАВИЛ С МУЛЬТИАКТИВИЗАТОРАМИ, УПРАВЛЯЕМОЙ ПОТОКОМ ДАННЫХ**

В настоящее время, в научной области создания систем искусственного интеллекта (СИИ), для повышения интеллектуальности существующих и перспективных автоматизированных систем сбора и обработки информации (АССОИ) актуальной является проблема разработки обучаемых, адаптивных механизмов логического вывода. Одним из наиболее перспективных формализмов представления знаний являются гиперправила с мультиактивизаторами [Л. 326-327]. В общем виде **гиперправило** (ГП) представляет собой конструкцию с детерминированной операционной семантикой: после запуска происходит выполнение ГП "слева-направо" и "сверху-вниз". Благодаря естественному совмещению процедурной и непроцедурной компонент гиперправила представляют собой достаточно удобное средство описания локальных функций обработки, увязываемых в единый процесс на основе специального механизма "доски объявлений" и наличия информации в базе данных. Для активизации гиперправил при условии поступления нескольких сообщений в [Л. 326] введено понятие мультиактивизатора. Таким образом, понятие **гиперправила с мультиактивизатором** (ГПМ) позволяет наиболее адекватно моделировать процессы логического вывода и интеллектуальной обработки данных в АССОИ, в том числе и в случае построения интеллектуальных систем или для создания системы адаптивного синтеза информационно-вычислительных конфигураций - САС ИВК. Для операционной поддержки аппарата ГПМ необходимо разработать специальный механизм вывода. Предлагается следующий подход к построению адаптивного механизма логического вывода на активной эволюционной сети ГПМ, управляемой потоком данных.

Существо предлагаемого подхода в том, что **из всех возможных ГПМ (отношений) и известных объектов (активизаторов, переменных) формируется логическая сеть ГПМ, управляемая потоком данных**. На основе этой сети ГПМ строится двудольный ориентированный граф, в некотором смысле, аналогичный графам сетей Петри [Л. 228, 326-327]. Затем, на этом графе, в зависимости от поступивших входных и требуемых выходных данных (объектов), формируется подграф логического вывода (алгоритма обработки). На этом подграфе вывода осуществляется поиск маршрута логического вывода (алгоритма обработки) путем анализа его связности. Для определения связности графа используют алгоритмы поиска минимального разреза многополюсных сетей. Если минимальный разрез равен нулю, то граф разбивается на различные компоненты, не связанные между собой. В нашем случае, если минимальный разрез равен нулю, значит,

маршрута логического вывода (алгоритма обработки) не существует. Если минимальный разрез больше либо равен единице, то маршрут (алгоритм) "успешного" логического вывода (обработки) существует.

Необходимость разработки нового подхода к построению механизма вывода определяется тем, что сначала определяют маршрут логического вывода (алгоритм обработки) на основе специальной процедуры (набора "метаправил"), а только потом осуществляют непосредственный запуск необходимых ГПМ (процедур и правил обработки данных) и получают требуемые значения. В общем случае, проблема поиска маршрута логического вывода является NP-полной, т.е. для ее решения необходим полный перебор всех возможных вариантов, что характеризуется факториальным ( $n!$  - факториал) ростом вычислительной сложности. В то же время, известно несколько алгоритмов поиска минимального разреза двухполюсной сети кубической вычислительной сложности  $O(n^3)$ , т.е. с кубической зависимостью количества операций обработки (вычислений) от количества вершин ( $n$ ) графа сети и, кроме того, известен алгоритм с вычислительной сложностью  $O(n^{2.5})$  [Л. 55-56, 146, 173, 308, 310, 416-426]. Однако, в [[Л. 41-42] предложен **квадратичной сложности алгоритм поиска минимального разреза** многополюсной сети, т.е.  $O(n^2)$ . Следовательно, предлагаемый подход к построению механизма логического вывода на основе поиска минимального разреза многополюсной сети, позволяет значительно сократить время поиска маршрута логического вывода и даже, свести ее вычислительную сложность от NP-полной к квадратичной. Подчеркнем, что в более общем смысле, подобный подход применим и к определению последовательности запуска (алгоритма обработки) процедур обработки данных, применяемых в интеллектуальных пакетах прикладных программ (ИППП). В дальнейшем, говоря о поиске маршрута логического вывода, будем подразумевать и формирование алгоритма обработки данных для ИППП и тому подобных приложений. Кроме того, так как после определения маршрута логического вывода и определения последовательности запуска гиперправил (ГПМ), на обработку не запускаются лишние и не нужные для получения вывода правила (ГПМ), то существенно сократится и общее время обработки потока входных данных. Особо подчеркнем, что в зависимости от имеющихся значений переменных и правил обработки (отношений), подграфы обработки-вывода могут динамически изменяться. При этом возможно формирование новых правил (ГПМ, процедур) обработки, существенно сокращающих процесс логического вывода.

Допустим, что все ГПМ могут быть пронумерованы единым списком. Элементами мультиактивизаторов являются переменные (объекты), которые могут быть входными или выводимыми (промежуточными) переменными (объектами). Пусть все переменные (объекты), входящие в состав мультиактивизаторов ГПМ и самих правил, также образуют некоторый единый список. Логическая сеть ГПМ может быть представлена в виде двух взаимосвязанных списков: переменных (входных и выходных) и правил (гиперправил, ГПМ).

Кроме того, для каждого правила определяются входные (т.е. входящие в мультиактивизатор) и выходные (выводимые, т.е. получаемые в результате выполнения ГПМ) переменные. Будем считать, что если у любого правила заданы все входные переменные, т.е. может быть произведена активизация правила, то после выполнения этого правила должны быть определены (означены) все его выходные переменные. Такое **представление логической сети позволяет в любой момент времени добавить, изменить или удалить любые переменные или правила, т.е. возможно эволюционное развитие этой сети.** В этих списках переменных для каждой из них определяется номер или идентификатор соответст-

вующего правила, а также ее роль: входная (активизирующая) или выходная (выводимая). Аналогично, для каждого правила соответственно определяются все входные и выходные переменные. Таким образом, получаем двудольный ориентированный граф возможных взаимосвязей всех переменных и правил, т.е. **граф логической сети ГПМ**. Отметим, что взаимосвязи между элементами одного множества (т.е. переменных с переменными или правил с правилами) в двудольном графе запрещены.

В зависимости от набора входных переменных определенные ребра этого графа из "возможных" становятся "реализуемыми", т.е. "существующими". Таким образом, на основании единого общего двудольного ориентированного графа логической сети (возможных логических взаимосвязей правил и переменных) можно построить различные **подграфы логического вывода**, в зависимости от конкретных значений входных и искомых переменных. Отметим, что именно табличная или, другими словами, "списочная" форма представления такого графа логической сети делает его эволюционным, т.е. позволяет в любое время изменить его состав и структуру (конфигурацию). Кроме того, возможно представление такого графа в двумерном пространстве или в виде двумерной таблицы представления данных.

Возможны две стратегии поиска маршрута логического вывода: от входных переменных ("сверху") и от искомых переменных ("снизу"). При первой стратегии подграф логического вывода строится исходя из перечня значений входных переменных и до достижения искомых переменных. При второй стратегии этот подграф строится в обратном порядке. В предлагаемом механизме вывода для ускорения построения подграфа логического вывода с "реализуемыми" ребрами, предлагается объединить обе эти стратегии, попеременно ("пошагово") выполняя построение требуемого подграфа для поиска маршрута вывода снизу и сверху, до их пересечения, слияния. Подчеркнем, что, таким образом, эта логическая сеть ГПМ (и ей подобные) **управляется потоком входных** данных, т.к. построение подграфа логического вывода и весь поиск маршрута логического вывода ведется только на основе потока известных значений входных переменных (перечня исходных переменных). Прежде всего, необходимо выделить (определить) следующие непересекающиеся множества переменных:

- 1) входные (исходные),
- 2) выходные (искомые),
- 3) выводимые,
- 4) активизирующие,
- 5) невыводимые.

Изначально, все переменные логической сети принадлежат множеству невыводимых переменных. Кроме того, все правила (процедуры, ГПМ) целесообразно разделить на три непересекающихся множества:

- 1) запускаемые,
- 2) требуемые и
- 3) незапускаемые.

В исходном состоянии все процедуры относятся к множеству незапускаемых правил (ГПМ). На рис. 5.4.1 показан пример построения двудольного ориентированного графа для поиска маршрута вывода. Отметим, что все множества переменных и правил, а также связи между ними и образуют граф логической сети, на котором, в зависимости от входных и искомых переменных, выделяется подграф вывода.

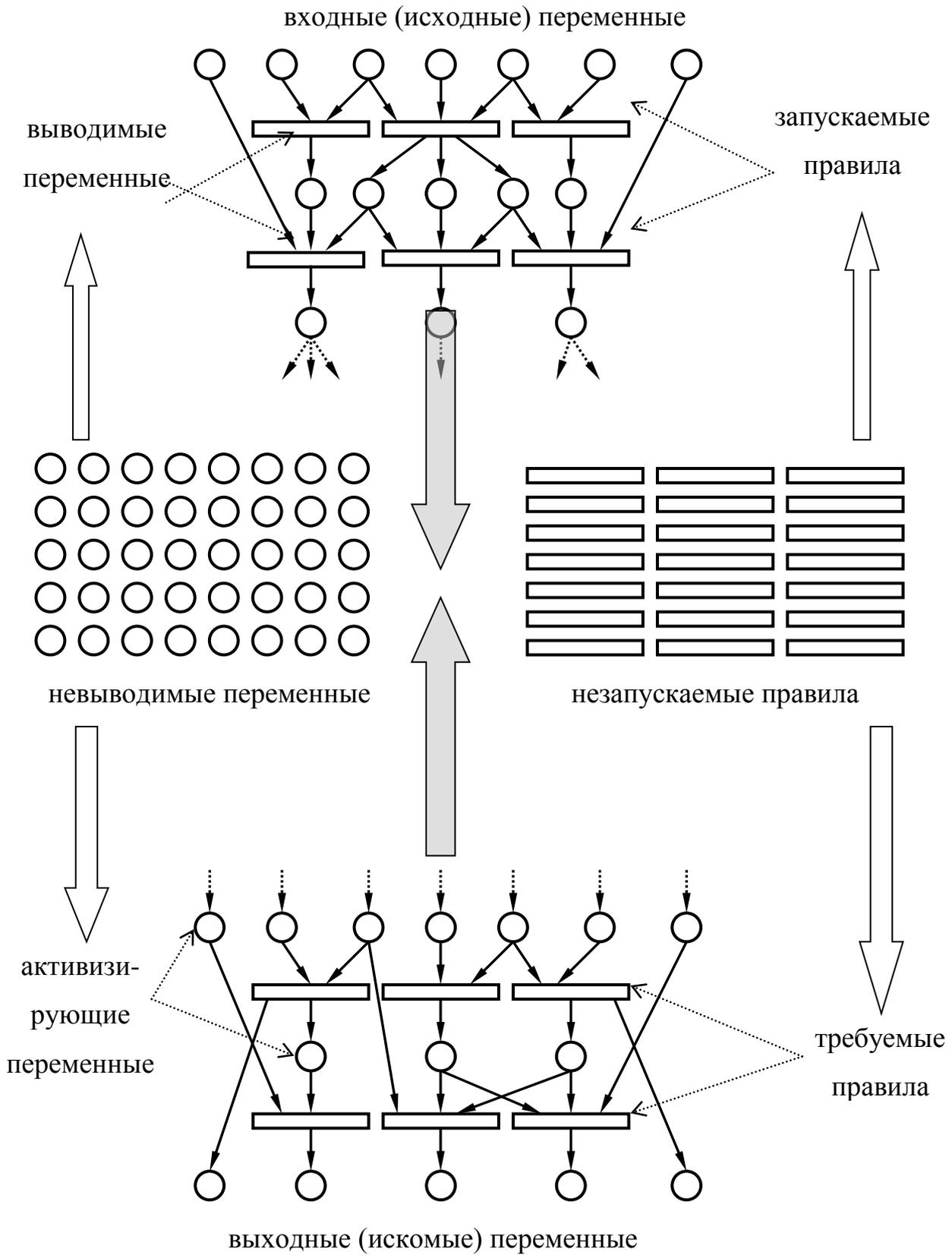


Рис. 5.4.1. Построение графа для поиска маршрута логического вывода.

Если существует маршрут "успешного" логического вывода и его подграф вывода может быть построен (выделен на графе сети ГПМ), то в его состав не должны входить:

- невыводимые переменные,
- требуемые правила и
- незапускаемые правила.

В общем случае, после построения подграфа вывода осуществляется его анализ с целью определения оптимального маршрута логического вывода. Подчеркнем, что в зависимости от исходных (входных) и искомым переменных, подграф логического вывода может быть от тривиального (один этап вывода) до полного графа всей сети ГПМ. Следовательно, во многих случаях моделирования сложных многообразных предметных областей, такой подход позволяет избежать лишних затрат вычислительных ресурсов. Итак, механизм вывода должен найти маршрут логического вывода для каждого заданного множества входных и искомым переменных.

Этот **механизм вывода функционирует следующим образом**. Пусть на вход логической сети в некоторый момент времени поступили значения нескольких входных переменных, которые пересылаются во множество входных переменных. Тогда происходит присвоение значений этим переменным ("означивание") и осуществляется проверка наличия значений у искомым переменных. В некоторых случаях маршрут логического вывода может быть определен в процессе построения подграфа вывода. Например, если всем искомым переменным присвоены значения, то вывод завершен. В противном случае, целесообразно начать поиск "снизу" и выбрать (определить) такие правила, для которых искомые значения являются выходными. Эти правила пересылаются в множество "требуемых". Затем происходит выбор таких переменных ("активизирующих"), которые являются входными для только что выбранных "требуемых" правил. Сравниваем выбранные "активизирующие" и входные переменные. Если не всем "активизирующим" переменным могут быть присвоены полученные и известные значения, то поиск маршрута логического вывода (алгоритма обработки данных для интеллектуальных пакетов прикладных программ - ИППП) продолжается. Теперь целесообразно сделать шаг "сверху" и выбрать такие правила обработки (ГПМ), для которых заданы ("означены") все входные, т.е. активизирующие, переменные. На данном шаге, такими переменными могут быть только входные (исходные) переменные. Если таких правил нет, то поиск маршрута вывода (обработки) на этом завершается и формируется сигнал о том, что не определено достаточное количество переменных и требуется дополнительная информация.

**Активность.** Существует возможность выделить те правила, для активизации или полного "означивания" входных переменных которых не хватает минимального количества переменных. Такие переменные могут быть отсортированы по количеству правил (гиперправил с мультиактивизаторами - ГПМ), в которых они встречаются, а затем в этом ранжированном порядке список переменных выдается подсистеме сбора информации АССОИ для уточнения и определения значений требуемых переменных. Однако, в некоторых случаях, когда заранее известны сложность и требуемое время обработки каждого правила, приоритетность может быть рассчитана относительно минимального времени обработки, т.е. запускать в первую очередь надо, например, самые быстро выполняемые правила. Конкретные критерии приоритетности определяются исходя из заданных условий, технических возможностей, специфики решаемой задачи и предметной области. Отметим, что формирование таких списков "потребностей" является **при-**

**знаком "активности"** системы обработки, т.е. такая автоматизированная система самостоятельно, но, конечно, по определенным правилам, запрашивает данные, определяет и ранжирует важность необходимых входных данных. Если есть такие правила (ГПМ), у которых заданы (могут быть активизированы) все входные переменные, то происходит "означивание" всех их выводимых переменных, эти правила пересылаются в множество "запускаемых" правил, а "означенные" переменные пересылаются в множество "выводимых" переменных.

**Параллелизм.** Осуществляют сравнение "искомых" и только что определенных "выводимых" переменных. Если все "искомые" переменные принадлежат множеству "выводимых" переменных, то маршрут логического вывода существует, задача успешно решена и можно переходить к оптимизации полученного маршрута вывода. Фиксируются все "запускаемые" правила и, одновременно, формируется список "запускаемых" правил, которые будут запущены в первую очередь. Подчеркнем, что если технически возможно параллельное выполнение нескольких программ, то выбранные **правила** могут быть **запущены** одновременно, т.е. **параллельно**.

Предлагается следующий **порядок определения приоритета правил**. Прежде всего, выделяют все правила, в результате обработки которых будут известны требуемые (искомые) переменные. Чем больше искомых переменных определяет правило, тем выше его приоритет обработки. Кроме того, априорно каждому правилу может быть задана его важность, которая также должна некоторым образом учитываться при определении его приоритета обработки.

Если не все "искомые" переменные принадлежат множеству выводимых переменных, то поиск маршрута вывода продолжается. В этом случае, целесообразно осуществить шаг "снизу" по построению подграфа логического вывода и выбрать (определить) из "незапускаемых" такие "требуемые" правила, для которых "активизирующие" переменные предыдущего этапа являются выводимыми или исходными. Можно считать, что значения выводимых переменных к этому этапу будут уже известны.

Затем происходит выбор таких "активизирующих" переменных, которые являются входными для только что выбранных "требуемых" правил. Сравниваем выбранные "активизирующие" переменные с множествами входных и выводимых переменных.

Если не всем "активизирующим" переменным могут быть присвоены значения, то построение подграфа логического вывода продолжается. Теперь опять целесообразно сделать шаг "сверху" и выбрать такие правила обработки (гиперправила с мультиактивизаторами), у которых все входные (активизирующие) переменные принадлежат множествам входных и выводимых переменных.

Если таких правил (ГПМ) нет, то поиск маршрута логического вывода (обработки) на этом завершается и, аналогично выше указанному, формируется специальный сигнал о том, что не определено достаточное количество значений входных переменных и требуется дополнительная информация. Если же есть такие правила (ГПМ), у которых известны (заданы или выводимы) все входные (активизирующие) переменные, то происходит "означивание" всех их выводимых переменных, сами эти правила пересылаются в множество "запускаемых" правил, а "означенные" (выводимые, вычисляемые) переменные пересылаются в множество "выводимых" переменных.

В очередной раз, осуществляют сравнение "искомых" и только что определенных "выводимых" переменных. Если все "искомые" переменные принадлежат

множеству "выводимых" переменных, то маршрут логического вывода существует, и, следовательно, задача успешно решена.

Если задача не решена, то анализируется возможность дальнейшего проведения поиска маршрута логического вывода, т.е. наличие неиспользованных ранее "незапускаемых" правил и "невыводимых" переменных.

Если поиск маршрута логического вывода (построение подграфа вывода) может быть продолжен, то аналогично указанным шагам, дальше по очереди производят поиск, чередуя шаги "снизу" и "сверху" до построения подграфа вывода, нахождения маршрута вывода или выявления невозможности этого.

Получаем, что с двух противоположных сторон на основе графа логической сети гиперправил с мультиактивизаторами, т.е. от входных (известных, исходных) и от выходных (искомых) переменных, формируется (выстраивается) подграф вывода, т.е. подграф некоторой многоэтапной структуры обработки правил, функционирования гиперправил с мультиактивизаторами.

Определение очередности обработки правил, возможности их запуска и выявление выводимых каждым правилом переменных производят без выполнения самих правил, исходя только из того, что если, как было указано выше, у правила заданы все входные (активизирующие для гиперправил с мультиактивизаторами) переменные, то обязательно могут быть некоторым образом определены и все выходные переменные. Отметим, что для многих предметных областей время и необходимые ресурсы обработки каждого правила (гиперправила с мультиактивизаторами) могут существенно различаться друг от друга. Тем не менее, будем исходить из того, что время и ресурсы построения двудольного ориентированного графа и поиска маршрута логического вывода (определения этапов) обработки и очередности запуска правил не зависят от самих правил и существенно меньше времени обработки данных и необходимых для этого вычислительных ресурсов.

В том случае, если у нескольких правил совпадают все рассчитанные параметры, то более высокий приоритет для выполнения, в целях избежания конфликтов, получает правило с меньшим первоначальным номером в списке всех правил.

В процессе построения подграфа вывода, т.е. определения порядка и очередности выполнения правил обработки, может быть выявлена такая ситуация, когда "исходных" и "выводимых" переменных не достаточно для определения "искомых" переменных, тогда обработка завершается без искомого результата. Аналогично, указанному выше, пользователям автоматизированной системы сбора и обработки информации сообщается о невозможности получения искомого результата и "активно" определяются наиболее важные для завершения обработки переменные, перечень которых выдается в подсистему сбора автоматизированной системы сбора и обработки информации - АССОИ - в качестве запроса для добавления и уточнения входных данных.

Если подграф вывода построен, и маршрут логического вывода определен, то, возможно, после некоторой оптимизации, осуществляется собственно запуск механизма логического вывода.

Таким образом, исходя только из входных данных и начального списочного формирования сети правил (гиперправил с мультиактивизаторами), "вычисляется" и формируется логическая сеть (подграф) обработки данных.

## 5.5. АДАПТИВНОСТЬ МЕХАНИЗМА ЛОГИЧЕСКОГО ВЫВОДА НА ЭВОЛЮЦИОННОЙ АКТИВНОЙ СЕТИ ГИПЕРПРАВИЛ С МУЛЬТИАКТИВИЗАТОРАМИ

**Адаптивность.** В процессе функционирования логической сети могут быть получены новые исходные данные:

- новые значения уже известных переменных или
- значения новых переменных.

В случае появления новых значений уже известных переменных, фиксируется время их появления и, при необходимости, осуществляется новый запуск всей цепочки логического вывода. При этом, одновременно (параллельно) могут выполняться, "вычисляться" разные значения одних и тех же переменных, в зависимости от времени начала их обработки, что должно корректно учитываться.

В случае появления новых "исходных" данных, а также в более общем случае добавления, изменения или удаления переменных, поступления значений новых переменных, процесс вывода, при необходимости, приостанавливается и, исходя из известных переменных, к числу которых относятся полученные ранее "исходные" и уже вычисленные переменные, добавляются (изменяются, удаляются) значения новых "исходных" переменных и осуществляется новый поиск маршрута логического вывода.

Подчеркнем, что на одном и том же наборе правил может осуществляться одновременно, но с соблюдением специальных мер поддержания корректности вывода, несколько процессов вывода, в зависимости от времени поступления значений "исходных" переменных. Кроме того, определение и вычисление маршрутов логического вывода, что является "метавычислениями" для системы, также могут одновременно осуществляться на общем наборе правил и общей вычислительной системе. Получаем, что, сформировав набор правил (ГПМ) и "метаправил" (механизм поиска вывода), в зависимости от получаемых значений входных переменных, "запущен" такой **механизм логического вывода**, который затем может **адаптивно функционировать** не только в автоматизированном режиме, но и полностью **автономно**, т.е. без вмешательства человека.

**Эволюционность.** Предлагаемый "списочный" (двумерный) способ унифицированного учета правил и переменных позволяет без перепланирования баз данных и знаний (в некоторой двумерной таблице правил/переменных) добавлять новые правила и новые переменные, а также изменять и удалять уже существующие. Это реализуется следующим образом. Фиксируется момент времени ввода новых правил и/или переменных, что необходимо для поддержания целостности вывода предшествующих значений переменных. Затем, новые правила и/или переменные вводятся в соответствующие списки (заголовки двумерной таблицы или в соответствующие оси двумерного пространства унифицированного представления данных и правил), и процессы формирования (поиска, вычисления) маршрутов вывода с этого момента, просто работают с новыми списками без всяких дополнительных усилий и указаний.

Модернизация и удаление правил и переменных также возможны, но в зависимости от особенностей предметной области уже (и все еще) вычисляемые процессы обработки (гиперправила с мультиактивизаторами) должны быть либо запущены заново, либо должны быть завершены на основе старых данных.

Таким образом, получаем реально **эволюционную** (развивающуюся) логическую сеть гиперправил с мультиактивизаторами (обработки), управляемую по-

током входных данных. Отметим, что эта сеть гиперправил с мультиактивизаторами (и аналогичные ей) является объективно двумерной и может быть наиболее адекватно представлена в перспективном разрабатываемом формализме информационного многомерного динамического (миварного) пространства унифицированного представления данных и правил.

**Самообучение.** Полученная активная эволюционная логическая сеть, кроме всего прочего, самостоятельно (по заданным алгоритмам, но автоматически) активно формирует запросы на уточнение и дополнение входных данных. Следовательно, для такой логической **сети может быть реализовано самообучение.** В некоторых случаях, после построения подсистемы сбора информации и после задания, определения или формирования правил и переменных подсистемы обработки, полученная логическая система может функционировать без вмешательства человека или других внешних воздействий, если все подсистемы будут корректно взаимодействовать друг с другом. Кроме того, могут быть заданы такие "метаправила", которые позволят создаваемой системе самостоятельно формировать новые правила ("метаправила", ГПМ) обработки данных, на основе ассоциативного анализа подграфов вывода и поиска, выявления новых закономерностей.

**Диагностика.** В отличие от известных традиционных систем логического вывода, в которых вывод (обработка данных) происходит только по запросу, создаваемая логическая сеть постоянно обрабатывает новые входные переменные, т.е. осуществляет постоянную потоковую обработку.

Если на основе сформированного адаптивного механизма вывода и эволюционной логической сети ГПМ создать некоторую информационную (логическую, концептуальную) модель предметной области и затем постоянно подавать на вход поток данных, то такая сеть самостоятельно будет вести всю обработку, определяя постоянно новые значения всех переменных. Целесообразно для некоторых предметных областей задать некоторое количество "**диагностических**" **правил обработки** результатов моделирования, которые в случае перехода заданных переменных из одной области значений в другую, **выполняли бы заданные действия.** Например, подавали бы сигнал тревоги или просто посылали сообщения пользователям о новых состояниях модели и диагностируемой предметной области.

В некотором смысле, это можно рассматривать как аналог разрабатываемых ранее диагностических экспертных систем реального времени (ЭС РВ). В таком случае, получаем логичное совмещение распознавания и диагностики в одной эволюционной АССОИ. Кроме того, на одной и той же активной эволюционной логической сети ГПМ, управляемой потоком данных, может одновременно функционировать несколько различных информационных моделей, результаты работы, которых целесообразно сравнивать, а также выбирать из них наиболее адекватные по итогам экспериментальной апостериорной проверки.

Итак, возможности по изменению, адаптации описания (моделирования) предметной области, т.е. правил и данных, придают адаптивному механизму вывода на активной эволюционной логической сети гиперправил с мультиактивизаторами способности к ее обучению, а в перспективе - самообучению.

Эволюционная логическая сеть гиперправил с мультиактивизаторами, управляемая потоком данных, объединяет и обработку известных данных на основе известных правил, и одновременно осуществляет обучение, изменение системы правил со своевременным внесением изменений в соответствующие процессы обработки, что обеспечивает эволюционное, синергетическое развитие всей АССОИ (компьютерной системы) в целом.

Таким образом, активное формирование запросов на получение требуемых входных данных, возможность эволюционного изменения состава и структуры (правил и переменных) логической сети гиперправил с мультиактивизаторами, потоковая обработка входных данных позволят построить эволюционную активную сеть гиперправил с мультиактивизаторами, управляемую потоком данных. На основе того, что из всех возможных гиперправил с мультиактивизаторами и известных переменных можно сформировать логическую сеть ГПМ, построить ее двудольный ориентированный граф, на котором, в зависимости от поступивших входных и требуемых выходных данных, адаптивно формируется подграф логического вывода и осуществляется поиск маршрута логического вывода путем поиска минимального разреза, предлагается новый подход к построению адаптивного механизма логического вывода на эволюционной активной сети гиперправил с мультиактивизаторами.

В определенном смысле, можно аналогично построить адаптивный механизм определения алгоритма обработки данных для интеллектуальных пакетов прикладных программ, на основе эволюционной активной сети процедур обработки, управляемой потоком данных.

Кроме того, практическая реализация предлагаемого адаптивного механизма логического вывода на эволюционной активной сети гиперправил с мультиактивизаторами, управляемой потоком данных, **позволит** совместить в АССОИ процессы обучения и диагностики; максимально распараллелить обработку любого потока данных; снизить вычислительную сложность поиска маршрута логического вывода с NP-полной до квадратичной; обеспечить преемственность, совместимость и взаимодействие с информационным динамическим многомерным (миварным) пространством унифицированного представления данных и правил, экспертными системами и интеллектуальными пакетами прикладных программ; гарантировать оперативность и своевременность логической обработки потока данных с некоторым фиксированным временем задержки обработки, а также повысить эффективность использования технических средств АССОИ.

Таким образом, если задача поиска маршрута логического вывода не является сложной, количество переменных и процедур мало, то нет необходимости использовать предлагаемый метод быстрого "графового" поиска логического вывода. Но для реальных задач, прежде всего - для решения сложных специальных задач, время и ресурсы, затрачиваемые на "графовые" расчеты, как правило, значительно меньше, чем время поиска и выполнения логических процедур обработки.

Кроме того, если маршрута логического вывода нет, т.е. минимальный разрез равен нулю, то никакого логического вывода вообще можно не делать, а сразу посылать уточняющие запросы в подсистему сбора информации.

## 5.6. КВАДРАТИЧНОЙ СЛОЖНОСТИ МЕТОД ПОИСКА МИНИМАЛЬНОГО РАЗРЕЗА МНОГОПОЛЮСНЫХ СЕТЕЙ

### 5.6.1. АНАЛИЗ ПОТОКОВОГО ПОДХОДА К ПОИСКУ МИНИМАЛЬНОГО РАЗРЕЗА

Адаптивную логическую сеть правил (гиперправил с мультиактивизаторами), управляемую потоком данных, можно представить в виде некоторого многоуровневого многотипового динамического графа:

$$G=(V, X),$$

в котором,  $n$ - количество вершин. Граф, в котором выделено  $S$  вершин, называют **S-полюсной сетью** или, в более общем виде: многополюсной сетью [Л. 41-42, 55-56, 146, 173, 308, 310, 416-426]. В нашем случае, под полюсами сети будут пониматься заданные и искомые логические переменные (объекты).

Под операцией **удаления вершины** из графа принято понимать такую операцию, при которой удаляется некоторая вершина вместе с инцидентными ей ребрами [Л. 416-426].

Под операцией **удаления ребра** принято понимать такую операцию, при которой удаляется только это ребро, т.е. соответствующая пара из множества  $X$ .

Полюса сети соединяются цепями. **Разрезом сети называют множество ребер, удаление которых разрушает все цепи.** Разрез называется минимальным, если он имеет минимально возможное число ребер [Л. 416-426]. Задача нахождения минимального разреза связана с задачей поиска максимального потока, традиционный подход, к которой характеризуется наличием достаточно большого количества известных различных алгоритмов.

**Основной целью разработки различных алгоритмов является снижение вычислительной сложности.** Для **разреза двухполюсной сети** предложены различные алгоритмы [Л. 55-56, 146, 173, 308, 310, 416-426], в зависимости от особенностей графа, вычислительной сложности  $O(n^3)$ , но наиболее эффективный из известных - алгоритм авторов Карив-Арбиб (Kariv-Arbib) со сложностью  $O(n^{2.5})$ .

Отметим, что в общем виде сформулированная задача поиска минимального разреза многополюсной сети по вершинам и ребрам ее графа является **NP-полной**, а эффективного алгоритма ее решения не предложено. Вместе с тем, **поточковый подход к нахождению разреза является избыточным**, т.к. нас не интересует максимальный поток, а основной вопрос - есть ли хотя бы одна цепь логического вывода, связывающая полюса, или сеть разбита на несвязанные компоненты.

### 5.6.2. КОНСТРУКТИВНОЕ ОПРЕДЕЛЕНИЕ РАЗРЕЗА МНОГОПОЛЮСНОЙ СЕТИ ПО ВЕРШИНАМ И РЕБРАМ ЕЕ ГРАФА

Приведенное выше определение разреза многополюсной сети не является конструктивным. Целесообразно предложить следующее конструктивное определение разреза многополюсной сети по вершинам и ребрам ее графа.

**Определение 5.6.1.** Под **разрезом** будем понимать такое множество удаляемых вершин и ребер, которое **разбивает граф на  $S$  компонент по числу полюсов сети, при этом различные полюса находятся в различных компонентах.**

Суть определения разреза не изменилась - полюса не взаимосвязаны. Каждому удаляемому ребру или вершине ставится в соответствие некий неотрицательный коэффициент, в зависимости от прикладной задачи.

Разрезов может быть несколько, а минимальным будет тот, для которого сумма соответствующих коэффициентов минимальна.

### 5.6.3. ПОИСК МИНИМАЛЬНОГО РАЗРЕЗА ДВУХПОЛЮСНОЙ СЕТИ ПО РЕБРАМ ЕЕ ГРАФА

Прежде всего, рассмотрим простейший случай разреза двухполюсной сети по ребрам. Предлагается следующий подход. Некоторым образом все вершины нумеруются (рис. 5.6.1):

- первый полюс - 1, а
- второй полюс -  $n$ .

Пусть значения всех коэффициентов ребер равны 1.

Строим нижнетреугольную матрицу смежности  $A$  следующим образом:

1.  $\forall i, \forall j a_{ij} = 0; i = \overline{1, n}; j = \overline{1, n}$ . (столбцы  $j$  соответствуют вершинам).
2. 1-ая вершина (первый столбец)  $j=1$ : если эта вершина связана с  $i$ -той вершиной, то  $a_{ij}=1$  для всех  $i = \overline{j+1, n}$ .
3. Для остальных столбцов определяем аналогично:  $\forall j, i = \overline{j+1, n}$  если эти вершины связаны ребром, то  $a_{ij}=1$ .

Количество ребер графа равно сумме значений всех клеток матрицы. Построив нижнетреугольную матрицу, переходим к поиску минимального разреза двухполюсной сети. **Существо предлагаемого подхода** в том, что если граф разбивается на две части (в первой  $k$  вершин, а во второй остальные -  $(n-k)$ ), то все ребра, связывающие вершины из разных частей графа принадлежат прямоугольнику, образованному клетками нижнетреугольной матрицы (рис. 5.6.1), которые расположены ниже  $k$ -той строки и левее  $(n-k)$ -того столбца.

Если граф разбит на два компонента, то практически матрица распадается на две подматрицы, а соответствующий прямоугольник (образованный клетками нижнетреугольной матрицы, которые расположены ниже  $k$ -той строки и левее  $(n-k)$ -того столбца) является нулевым (рис. 5.6.1).

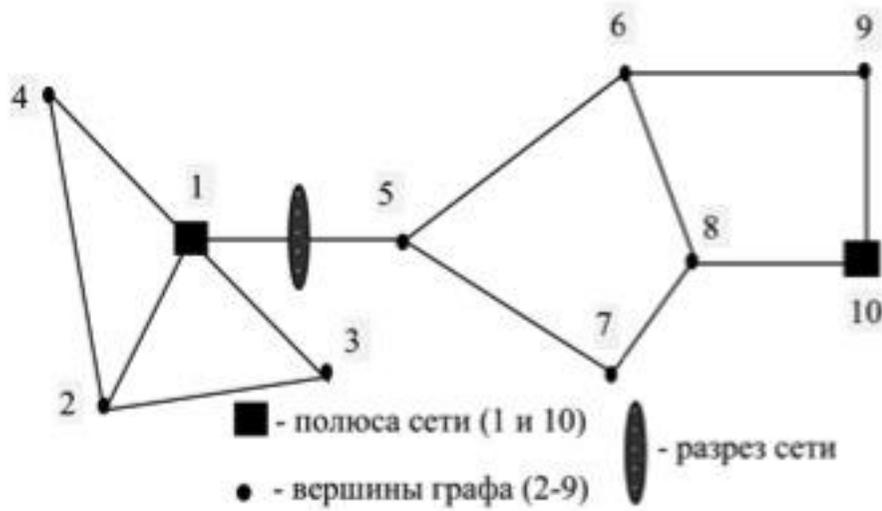
Для нахождения минимального разреза в нижнетреугольной матрице необходимо просуммировать все прямоугольники при  $k$  от 1 до  $(n-1)$ , а затем выбрать минимальный, которому и будет соответствовать искомый разрез.

Рассмотрим итеративную процедуру определения суммы.

Определяем сумму 1-го прямоугольника (1-ый столбец без элемента  $a_{11}$ ):

$$PR_1 = \sum_{i=2}^n a_{i1}, \quad (5.6.1)$$

где:  $k=1; j=1; i = \overline{j+1, n}$ .



а) граф двухполюсной сети

1	<b>p</b>									
2	1									
3	1	1								
4	1	1								
5	1									
6					1					
7					1					
8						1	1			
9						1				
10								1	1	<b>p</b>
	1	2	3	4	5	6	7	8	9	10

б) нижнетреугольная матрица смежности двухполюсной сети (min. разрез = 1)

Рис. 5.6.1. Пример поиска минимального разреза двухполюсной сети по ребрам.

Для второго прямоугольника от  $\mathbf{PR}_1$  вычитаем сумму его верхней строки (т.е. элемент  $a_{21}$ ), а затем добавляем сумму клеток по следующему столбцу.

$$\mathbf{PR}_2 = \mathbf{PR}_1 - a_{21} + \sum_{i=3}^n a_{i2}, \quad (5.6.2)$$

где:  $k=2$ .

Для  $k$ -го прямоугольника получаем следующую итеративную формулу:

$$\mathbf{PR}_k = \mathbf{PR}_{k-1} - \sum_{j=1}^{k-1} a_{kj} + \sum_{i=k+1}^n a_{ik}. \quad (5.6.3)$$

Из всех  $\mathbf{PR}_k$  выбираем минимальный, который соответствует минимальному разрезу двухполюсной сети по ребрам.

Тупиковость получаемых разрезов следует из конструктивного определения разреза, т.к. в указанные прямоугольники входят только такие ребра, которые связывают различные компоненты сети.

**Определим вычислительную сложность  $\Omega$ .** Всего определяются суммы для  $(n-1)$ -го прямоугольника, при этом для каждого из них выполняется  $(k-1)$  вычитание и  $(n-k)$  сложение, т.е. всего  $(n-1)$  действий. Получаем, что сложность определяется произведением числа прямоугольников на число действий для каждого прямоугольника:  $(n-1)^2$ .

$$\Omega = (n-1)((k-1) + (n-k)) = (n-1)(n-1) = (n-1)^2. \quad (5.6.4)$$

Следовательно, получаем **квадратичную вычислительную сложность  $\Omega$ , т.е.  $O(n^2)$** , что лучше, чем для всех других известных алгоритмов.

Отдельной проблемой является способ нумерации вершин, но для поиска пути логического вывода представляется достаточным следующее. Так как априорно известна структура графа, то нумерацию вершин начинают с известных переменных логической сети правил. Затем нумеруют наиболее близкие к ним выводимые переменные и т.д. Искомые переменные начинают нумеровать с наибольшего значения, аналогично нумеруя наиболее близкие к ним, но "непронумерованные" переменные.

Таким образом, логическая сеть правил как бы "растягивается" наподобие обычной сети, которую тянут в противоположные стороны, взяв, с одной стороны, за известные переменные, а с другой - за искомые. Каждой переменной присваивается только один номер. Все переменные, оставшиеся после нумерации взаимосвязанных с искомыми и известными переменными, а также "равноудаленные" переменные в своей области (зоне) нумеруются в некотором произвольном порядке. Такое частное решение проблемы нумерации вершин многополюсной сети представляется достаточным для исследуемой предметной области.

#### 5.6.4. ПОИСК МИНИМАЛЬНОГО РАЗРЕЗА МНОГОПОЛЮСНОЙ СЕТИ ПО РЕБРАМ ЕЕ ГРАФА

Решив задачу для двухполюсной сети, перейдем к поиску минимального разреза многополюсной сети. Особенностью данного подхода является то, что для многополюсной сети **строится фактически такая же матрица**, в которой по мере связности указаны все вершины и полюса. В полученной матрице ищем разрез 1-го полюса от всех остальных, который соответствует прямоугольникам от 1-го до 2-го полюса; затем определяем разрез 2-го полюса от всех и так далее. Из всех прямоугольников между соседними полюсами выбираем минимальные, исключая

при этом повторяющиеся клетки, и получаем минимальный разрез многополюсной сети по ребрам (рис. 5.6.2).

**Вычислительная сложность такого алгоритма меньше**, чем для двухполюсной сети за счет того, что, определив минимальный прямоугольник для предшествующей пары, в дальнейшем все операции производят только с оставшейся частью матрицы смежности (снижается размерность).

### 5.6.5. ПОИСК МИНИМАЛЬНОГО РАЗРЕЗА МНОГОПОЛЮСНОЙ СЕТИ ПО ВЕРШИНАМ И РЕБРАМ ЕЕ ГРАФА

Наиболее общим случаем, который представляет интерес для некоторых предметных областей, является поиск разреза многополюсной сети по вершинам и ребрам ее графа. **Разрез по вершинам и ребрам** отличается тем, что сначала удаляют вырезаемые вершины, вычеркивая из матрицы соответствующие строки и столбцы, а затем на оставшейся матрице определяют разрез по ребрам (рис. 5.6.3). Так как поиск разреза по вершинам связан с их перебором, то в общем случае данная задача **является NP-полной**.

Если, исходя из особенностей предметной области и прикладной задачи, представляется возможным допустить, что перечень удаляемых вершин известен заранее, то задача сводится к предыдущим. Сумма разреза по вершинам и ребрам складывается из суммы коэффициентов удаляемых вершин с вычеркиванием инцидентных ребер и из суммы удаляемых ребер при реберном разрезе оставшегося после удаления вершин графа. Оценим вычислительные ресурсы **КОР** для общего решения задачи:

$$\text{КОР} = \frac{n!}{z!(n-z)!} n^2, \quad (5.6.5)$$

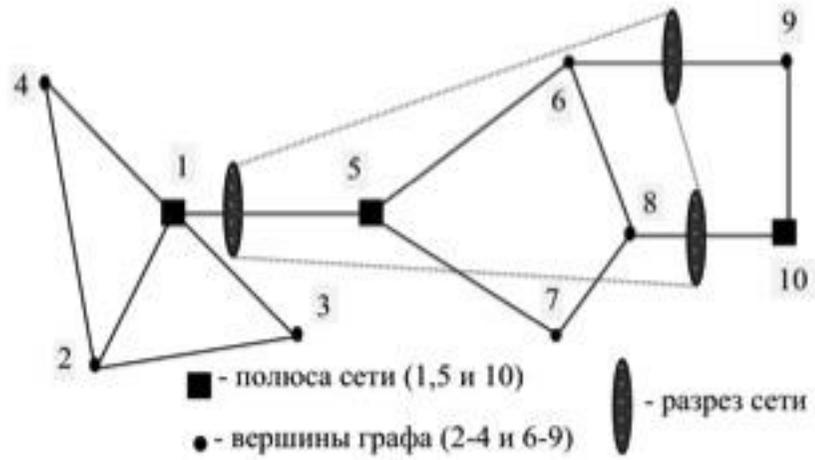
где  $z$  - количество удаляемых вершин. Если  $n \gg z$ , то

$$\text{КОР}_1 = \frac{n!}{z!(n-z)!} n^2 \leq \frac{n^{z+2}}{z!}. \quad (5.6.6)$$

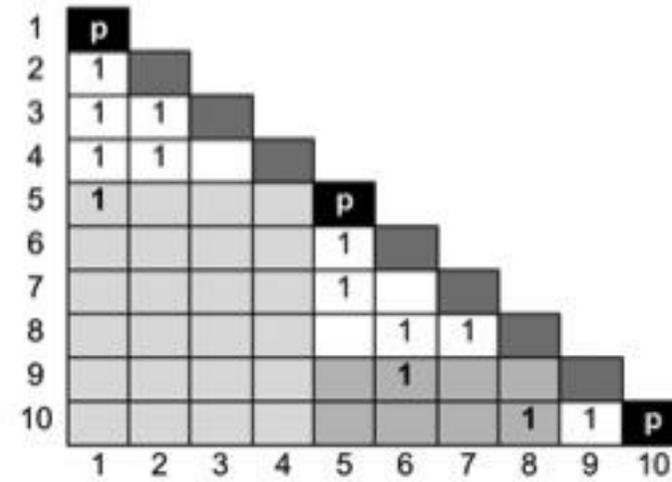
Если  $z \gg \frac{n}{2}$ , то

$$\text{КОР}_2 = \frac{n! \cdot n^2}{z! z!} \leq \frac{n^3 (n-1)(n-2) \cdots (n-z)}{z!} = \frac{n^{z+2}}{z!}. \quad (5.6.7)$$

Итак, предложено общее эффективное решение задачи нахождения минимального разреза многополюсной сети по вершинам и ребрам ее графа. Квадратичной сложности метод поиска минимального разреза многополюсных сетей может быть применен для ускорения поиска маршрута вывода практически на любых логических сетях правил, но, прежде всего, он предназначен для ускорения обработки данных на основе миварного адаптивного пространства унифицированного представления данных и правил.

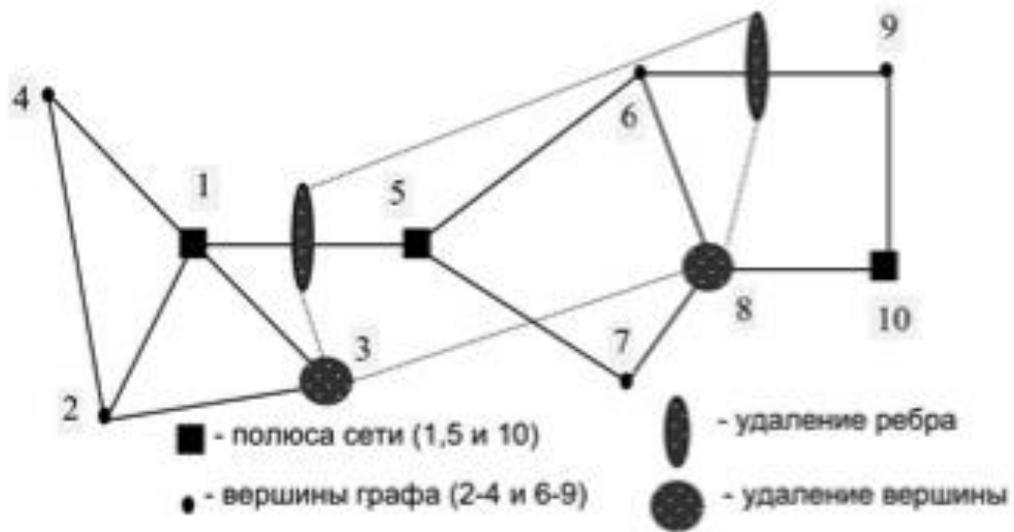


а) граф многополюсной сети



б) нижнетреугольная матрица смежности многополюсной сети (min. разрез 1+2= 3)

Рис. 5.6.2. Пример поиска минимального разреза многополюсной сети по ребрам.



а) граф многополюсной сети

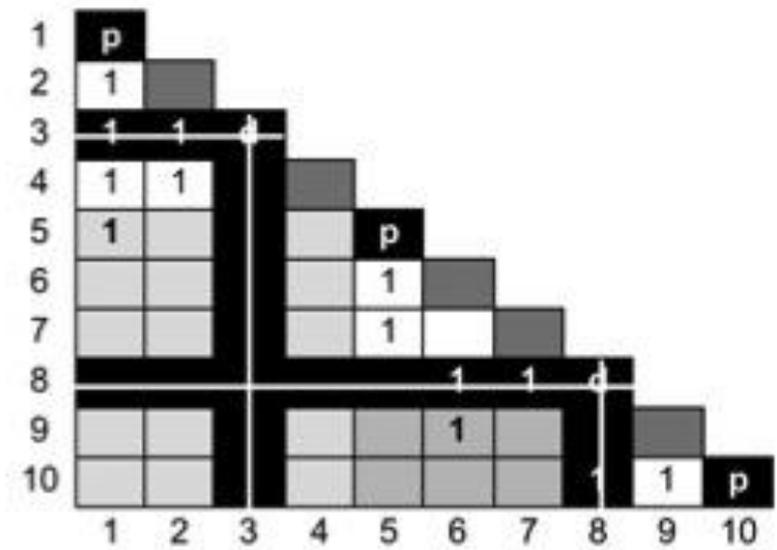
б) нижнетреугольная матрица смежности многополюсной сети ( $\min. \text{разрез} = 2v + 2p$ )

Рис. 5.6.3. Пример поиска минимального разреза многополюсной сети по вершинам и ребрам ее графа.

### 5.7. ЛИНЕЙНЫЙ МАТРИЧНЫЙ МЕТОД ОПРЕДЕЛЕНИЯ МАРШРУТА ЛОГИЧЕСКОГО ВЫВОДА НА СЕТИ ПРАВИЛ

Кроме указанных выше методов поиска маршрута логического вывода на сети правил возможны и другие методы, например, несколько подобный по идеологии, но отличающийся по реализации: матричный метод. Суть этого метода состоит в том, что для некоторой сети логических правил, представленной в списочной форме, строится матрица. Затем, на основе анализа этой матрицы определяется факт наличия успешного маршрута вывода, затем определяются возможные маршруты логического вывода, а на последнем этапе из этих маршрутов выбирают "кратчайший", наиболее оптимальный по заданным критериям оптимальности. К критериям оптимальности относят, например, скорость логического вывода, использование аппаратных средств, количество шагов логического вывода и т.д.

Итак, пусть известны  $m$  - правил и  $n$  - переменных (входящих в правила либо в качестве исходных, активизирующих, либо в качестве получаемых, т.е. выходных переменных). Тогда в матрице  $V$  ( $n \times m$ ) могут быть представлены все взаимосвязи между правилами и переменными. При этом, на каждой строке при описании каждого правила все входные переменные этого правила на соответствующих позициях матрицы помечаются символом  $x$ , все выходные -  $y$ , все переменные, которые уже получили в процессе вывода или задания исходных данных некоторое конкретное значение -  $z$ , а все искомые (выходные) переменные, т.е. те, которые необходимо "вывести" из исходных (входных) данных -  $w$ .

Кроме того, для уменьшения количества операций предлагаемого линейной сложности метода (процедуры) поиска маршрута логического вывода, целесообразно добавить в такую матрицу  $V$  одну строку и один столбец для хранения в них служебной информации.

Тогда, мы получаем матрицу  $V$  размерности  $(n+1) \times (m+1)$ , в которой отражена вся структура исходной сети правил. Сразу отметим, что, как и выше было показано, структура этой логической сети может изменяться в любое время, т.е. это сеть правил с изменяемой (эволюционной) структурой. Пример такой матрицы показан на рис. 5.7.1.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	
2			x	y	y			x	x	
...						...				
m		x		x	x		y			
m+1										

Рис. 5.7.1. Пример матрицы  $V$  размерности  $(n+1) \times (m+1)$ .

Опишем метод (алгоритм) матричного поиска маршрута вывода. Для поиска маршрута логического вывода на полученной матрице производят следующие действия.

1. В строке  $(m+1)$  помечают известные  $z$  и искомые  $w$  переменные, например, как на рис. 5.7.2 в строке на позициях:  $z$  - 1, 2, 3 и  $w$  - (n-2).

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	
2			x	y	y			x	x	
...						...				
m		x		x	x		y			
m+1	z	z	z				w			

Рис. 5.7.2. Пример обработки матрицы  $V$  размерности  $(n+1) \times (m+1)$ .

2. Осуществляют последовательно, например, сверху вниз, поиск таких правил, которые могут быть активизированы, т.е. у которых известны все входные переменные.

Если таких правил нет, то маршрута логического вывода нет и необходимо выдать запрос на уточнение (добавление) входных данных. Причем, как было сказано выше, на этом этапе возможно определение таких переменных, которые необходимы, прежде всего, для запуска правил. Не будем описывать это более подробно, так выше обсуждались аналогичные проблемы, которые в данном методе решаются точно также, как было показано выше.

Если такие правила, которые могут быть активизированы, есть, то у каждого из них в соответствующем месте служебной строки делается пометка, что правило может быть запущено. Например, можно ставить в этих клетках таблицы цифру **1**, что и показано на рис. 5.7.3 в клетке  $(n+1,1)$ .

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	1
2			x	y	y			x	x	
...						...				
m		x		x	x		y			
m+1	z	z	z				w			

Рис. 5.7.3. Пример обработки матрицы  $V$  размерности  $(n+1) \times (m+1)$ .

3. Если таких правил несколько, то осуществляется выбор по заранее определенным критериям такого или таких правил, которые должны быть активизированы в первую очередь. Данные вопросы уже подробно освещались выше, поэтому сейчас мы просто будем поступать аналогично. Отметим лишь, что при наличии достаточных ресурсов, одновременно могут запускаться сразу несколько правил.

4. Имитация запуска правила (процедуры) осуществляется путем присваивания выводимым в этом правиле переменным значений "известно", т.е. в нашем примере - **z**. Запущенное правило, для удобства дальнейшей работы помечается дополнительно, например, цифрой **2** (это не обязательно). Например, как показано на рис. 5.7.4 в клетках  $(n-1, m+1)$ ,  $(n, m+1)$  и  $(n+1, 1)$  произведены соответствующие изменения.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	2
2			x	y	y			x	x	
...						...				
m		x		x	x		y			
m+1	z	z	z				w	z	z	

Рис. 5.7.4. Пример обработки матрицы  $V$  размерности  $(n+1) \times (m+1)$ .

5. После имитации запуска правил проводят анализ достижения цели, т.е. анализируют получение требуемых значений путем сравнения служебных символов в служебной строке. Если в служебной строке  $(m+1)$  осталось хоть одно значение "искомая" (т.е.  $w$ ), то осуществляют дальнейший поиск маршрута логического вывода. В противном случае, задача считается успешно решенной, а все задействованные правила в соответствующем порядке их запуска и образуют искомый маршрут логического вывода. На следующем этапе необходимо провести оптимизацию полученного маршрута, но это мы обсудим позже. Итак, на рис. 5.7.4 запущено только одно правило, но искомые значения не определены, значит надо переходить к следующему этапу.

6. Прежде всего, определяют наличие таких правил, которые могут быть запущены после определения новых значений на предыдущем этапе. Если таких правил нет, то маршрута вывода нет и поступают аналогично пункту 2 этого метода (алгоритма).

Если такие правила есть, то продолжают поиск маршрута вывода. В нашем примере такие правила есть, что и показано на рис. 5.7.5. В клетке  $(n+1, 2)$  появилась  $1$ , как признак возможности запуска этого правила.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	2
2			x	y	y			x	x	1
...						...				
m		x		x	x		y			
m+1	z	z	z				w	z	z	

Рис. 5.7.5. Пример обработки матрицы  $V$  размерности  $(n+1) \times (m+1)$ .

7. На следующем этапе опять аналогично этапу 4 запускают правила (имитируют запуск), далее аналогично этапам 5 и 6 выполняют необходимые действия столько раз, сколько требуется для получения результата. При необходимости, повторяют все этапы со 2 по 7 до достижения результата.

При этом результат может быть как положительный - маршрут вывода существует, так и отрицательный - вывода нет из-за неопределенности входных данных.

Для наглядности продолжим пошаговое выполнение нашего примера. Итак, в нашем примере необходимо провести имитацию запуска правила 2, что и показано на рис. 5.7.6.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	2
2			x	y	y			x	x	2
...						...				
m		x		x	x		y			
m+1	z	z	z	z	z		w	z	z	

Рис. 5.7.6. Пример обработки матрицы  $V$  размерности  $(n+1) \times (m+1)$ .

В клетках  $(4, m+1)$  и  $(5, m+1)$  получаем признак выводимости переменных 4 и 5, а в клетке  $(n+1, 2)$  формируем признак того, что правило уже было запущено, т.е. ставим цифру 2. После этого, проводим анализ служебной строки и видим, что не все искомые переменные известны. Значит необходимо продолжить обработку матрицы  $V$  размерности  $(n+1) \times (m+1)$ . Анализ этой матрицы показывает возможность запуска правила  $m$ , что и отражено на рисунке 5.7.7.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	2
2			x	y	y			x	x	2
...						...				
m		x		x	x		y			1
m+1	z	z	z	z	z		w	z	z	

Рис. 5.7.7. Пример обработки матрицы  $V$  размерности  $(n+1) \times (m+1)$ .

Продолжим дальнейшую обработку примера. При запуске правила  $m$  появляются новые значения, в том числе и для искомых переменных, что и показано на рисунке 5.7.8.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	2
2			x	y	y			x	x	2
...						...				
m		x		x	x		y			2
m+1	z	z	z	z	z		z(w)	z	z	

Рис. 5.7.8. Пример обработки матрицы  $V$  размерности  $(n+1) \times (m+1)$ .

Итак, в нашем примере в служебной строке больше не осталось искомых правил, а в клетках таблицы появились новые значения: в клетке  $(n+1, m) - 2$ , а в клетке  $(n-2, m+1)$  вместо значения  $w$  появилось значение  $z$ .

Таким образом, получен положительный результат, следовательно, маршрут логического вывода при данных исходных значениях существует. Но для нас теперь важно определить количество выполненных действий по поиску маршрута логического вывода.

Для определения количества выполненных действий, проанализируем этот алгоритм.

Общее количество клеток в матрице  $V$  размерности  $(n+1) \times (m+1)$  определяется аналогичным произведением  $(n+1)$  на  $(m+1)$ .

При обработке столбцов этой матрицы мы присваивали по одному разу новые значения клеткам служебной строки  $(m+1)$ , причем на каждом следующем шаге алгоритма обработки, известные ранее клетки этой строки, как бы, "вычеркивались", т.е. больше не обрабатывались. Максимальное количество таких действий **не может превышать  $n$** .

При обработке правил, т.е. строк матрицы, мы присваивали по два раза значение в соответствующей клетке для каждого запускаемого правила. Второй раз отмеченные правила больше не обрабатывались, а соответствующие строки матрицы, как бы, "вычеркивались". Следовательно, после каждого запуска соответствующего правила, размерность анализируемой матрицы уменьшалась, что сокращало в дальнейшем количество последующих этапов обработки. Действий, описанных в данном пункте, **не может быть более чем  $2m$** . Заметим, что служебное поле  $(n+1, m+1)$  матрицы не обрабатывается, поэтому количество описанных действий пропорционально исходным числам, без добавления единицы.

Определение "запускаемых" правил (т.е. поиск таких правил, которые могут быть запущены после определения новых значений на предыдущем этапе) может быть эффективно реализовано следующим образом. Для нас важно знать только наличие каждой входной переменной правила в списке известных переменных. Полученное будем называть "текущим эталоном" или просто "эталонном". Тогда, каждую строку нашей таблицы можно представить в виде двоичного числа (бинарного), имеющего только два значения: **1** - если соответствующая переменная является входной для этого правила и **0** - в противном случае, т.е. даже когда эта переменная есть в правиле, но является выводимой из него. Если все строки такой матрицы будут представлять собой бинарные числа, то поиск запускаемых правил сводится просто к сравнению ("по маске") двух бинарных чисел. Количество таких сравнений зависит от многих параметров, но на каждом этапе обработки производится не более одного сравнения каждой оставшейся ("не вычеркнутой") строки - правила с "эталонном". Причем, на каждом последующем этапе количество таких сравнений уменьшается.

Для сокращения количества необходимых сравнений предлагается следующий подход. Все бинарные числа - "показатели" входных переменных правил необходимо отсортировать по возрастанию (либо по убыванию, но важно, чтобы соблюдался строгий порядок). Если у нескольких правил наборы входных чисел совпадают, то они просто будут иметь равные значения соответствующего бинарного числа. Напомним, что после каждого "успешного" этапа количество известных переменных возрастает, следовательно, наш "эталон" постепенно увеличивается, а уменьшаться он не может. Тогда, на каждом этапе обработки следует сравнивать текущее значение "эталона" с ближайшим значением в отсортированном списке "показателей" всех правил. Отметим, что этот список "показателей" постепенно уменьшается. При таком подходе текущее значение "эталона" может изменяться не более чем  **$n$** , т.е. количество переменных. Следовательно, и количество сравнений не должно превышать это же число, т.е. **не более  $n$** .

Здесь следует специально оговориться, что при одновременном "выводе" сразу нескольких новых переменных, для того, чтобы не "пропустить", т.е. не "перешагнуть" через нужные значения, следует изменять "эталон" каждый раз не более чем на один знак, затем производить сравнение и увеличивать эталон дальше. Впрочем, возможно, при определенном варианте сортировки переменных, будет целесообразно сразу увеличивать эталон на несколько значений, а все прави-

ла, у которых значение "маски" попадает в интервал от старого значения эталона до нового, считать "запускаемыми". Тогда количество сравнений может значительно сократиться.

Таким образом, получаем, что при некоторых предварительных не очень сложных действиях представляется возможным сократить общее количество сравнений "эталона" со значениями "масок" правил до такого числа, которое **не превышает общее количество переменных**, т.е. менее либо равно числу  $n$ .

При выполнении данного матричного метода, кроме выше указанных действий, необходимо также определять новые выводимые переменные и присваивать новые значения "эталону" известных переменных. Прежде всего, количество этапов появления новых выводимых значений не может превышать общего количества правил, т.е. оно менее либо равно  $m$ . Кроме того, само количество появления новых выводимых значений не может превышать общего количества переменных. Для этого, при каждом новом появлении выводимых значений переменных, аналогично выше указанному сортированию "масок" "по возрастанию", необходимо также отсортировать все "маски выводимых" переменных.

При каждом выявлении новых запускаемых правил, сразу после этого необходимо провести анализ и сравнение "маски выводимых" переменных, а последующие действия по присваиванию новых значений в "эталон" известных переменных производить только при необходимости, например, когда полученная маска выводимых переменных содержит значения новых, неизвестных до этого переменных.

Отметим, что можно каждый раз при появлении новой выводимой переменной просто изменять значения "эталона", но при этом, иногда уже текущему значению  $1$  будет еще раз присваиваться такое же значение. Конечно, в этом нет ничего опасного, а, возможно, в отдельных ситуациях для упрощения алгоритма обработки именно так и следует делать. В таком случае, количество таких "не оптимизированных" действий все равно не должно превысить количества клеток основной таблицы, т.е. произведения  $n \times m$ . Если же провести оптимизацию с использованием двоичных "масок", то общее количество всех присваиваний не будет превышать количества переменных, т.е. **не более числа  $n$** . Таким образом, общее количество действий при матричном методе определяется суммой действий на каждом этапе:

- 1) присваивание известных  $z$  и искомым  $w$  значений клеткам служебной строки  $(m+1)$ , общее количество таких действий **не более числа  $n$** ;
- 2) присваивание значений признака обработки правил клеткам служебного столбца  $(n+1)$ , общее количество таких действий не должно превышать числа  $2m$ , но при необходимости может быть сокращено в два раза так, чтобы оно было **не более числа  $m$** ;
- 3) присваивание известных  $z$  значений клеткам служебной строки  $(m+1)$  на основе сравнений "эталона" со значениями "масок" правил, общее количество таких действий **не более числа  $n$** ;
- 4) определение новых выводимых переменных и присваивание новых значений "эталону" известных переменных, т.е. клеткам служебной строки  $(m+1)$ , общее количество таких действий **не более числа  $n$** .

Отметим, что фактически на 1, 3 и 4 этапе производится обработка одного и того массива данных, т.е. клеток служебной строки  $(m+1)$ . При этом общее, суммарное количество действий на всех этих этапах (1, 3 и 4) не должно превышать общее количество клеток в этой строке, так как обработанные значения "вычеркиваются" и более не обрабатываются. Тогда, получаем, что **максимальное общее**

**количество всех действий (KD) при линейном матричном методе поиска маршрута логического вывода (вычислительная сложность)** не должно превышать количества клеток в служебных частях матрицы, т.е. не должно быть более, чем:

$$\mathbf{KD} \leq (\mathbf{n+m}). \quad (5.7.1)$$

В самом крайнем случае, когда нет возможности реализовать все предложенные здесь сокращения вычислений, матричный метод позволит решать задачу поиска маршрута логического вывода с вычислительной сложностью, пропорциональной произведению количества переменных на количество правил, т.е. количеству клеток этой матрицы:  $\mathbf{n} \times \mathbf{m}$ .

Рассмотрим еще несколько методов, позволяющих ускорить, т.е. повысить оперативность обработки данных в интеллектуальных многопроцессорных системах, например, построенных на основе предложенного миварного пространства представления данных и правил. Прежде всего, перейдем к анализу метода максимального распараллеливания доступа к общей базе данных, предназначенного для повышения оперативности обработки данных в АССОИ.

## 6. МЕТОД МАКСИМАЛЬНОГО РАСПАРАЛЛЕЛИВАНИЯ ДОСТУПА К ОБЩЕЙ БАЗЕ ДАННЫХ

### 6.1. АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ РАСПАРАЛЛЕЛИВАНИЯ ДОСТУПА К ОБЩЕЙ БАЗЕ ДАННЫХ

Применяемые для решения специальных задач методы логической обработки данных предоставляют возможность параллельной потоковой обработки входных данных. Для реализации этого представляется целесообразным использование, как правило, общей базы данных (БД). В силу параллельности и асинхронности функционирования системы обработки данных поток обращений к общей БД является нерегулярным и включает операции считывания информации из базы данных и ее обновления. При этом неуправляемый доступ параллельных процессов обработки к общей БД, который обеспечивает наибольшую скорость обработки информации, в общем случае не допустим по причине возможного взаимного искажения процессами обработки используемой ими информации. С другой стороны, организация строго последовательного выполнения операций над БД исключает возможность оперативности и своевременности комплексной обработки потоков данных практически при любом реальном соотношении их интенсивности и характеристик производительности ЭВМ.

Обзор литературы и анализ различных задач показал [Л. 5-7, 9-13, 16, 20-28, 33-40, 55-57, 62-66, 70-75, 86-96, 107-118, 120-132, 135, 139-152, 157-167, 183-186, 190-216, 224-228, 242-244, 248-253, 261-264, 271, 274-280, 284-287, 296, 304-311, 313-328, 331, 337-413, 419-421, 460-509], что при функционировании АССОИ общую БД многопроцессорных или интеллектуальных систем целесообразно размещать на сервере, тогда процессы обработки будут функционировать параллельно и порождать транзакции обработки БД, которые, в общем случае, не являются независимыми и могут взаимно искажать обрабатываемые данные. Некоторые последующие процессы, требующие малого времени обработки, могут быть завершены раньше, чем предшествующие процессы.

Возникает **противоречие: изменения необходимо сохранить, но вносить их непосредственно в БД нельзя**, так как последующие по времени поступления на обработку процессы не должны оказывать влияния на предшествующие им процессы обработки. Следовательно, для решения этого противоречия необходимо разработать метод максимального распараллеливания множественного доступа к общей БД, при котором гарантируется корректность ее обработки. Разработка такого метода представляет собой проблему обеспечения параллельности потоковой обработки баз данных.

В известных работах [Л. 5-7, 10-12, 16, 20-22, 25-26, 34-40, 55-57, 74-75, 109, 126-132, 147-152, 167, 195, 224, 274-275, 285-287, 326, 346-347], посвященных распараллеливанию, рассматриваются следующие возможные виды параллелизма:

- 1) одиночный поток команд, одиночный поток данных;
- 2) одиночный поток команд, множественный поток данных;
- 3) множественный поток команд, одиночный поток данных;
- 4) множественный поток команд, множественный поток данных.

В контексте решаемой задачи, относящейся к последнему виду параллелизма, наиболее адекватным представляется аппарат организации одновременного выполнения нескольких транзакций и запросов [Л. 73-74, 195, 264, 310-311, 326], порождаемых независимыми друг от друга процессами обработки данных. Для

обеспечения корректности, как правило, процессы делят на более мелкие функциональные единицы обработки: транзакции, запросы, обращения, а сам ресурс БД дробят на меньшие составные части: файлы, записи, элементы записей. В ряде случаев это позволяет несколько повысить скорость обработки, но при одновременном обращении нескольких транзакций к одному блокируемому элементу ресурса **фактически организуется последовательная обработка БД**. В этом случае ни один из известных методов обеспечения параллельности обработки БД [Л. 5-7, 10-12, 16, 20-22, 25-26, 34-40, 55-57, 74-75, 109, 126-132, 147-152, 167, 195, 224, 274-275, 285-287, 326, 346-347] не гарантирует ее корректности. Подчеркнем, что для некоторых прикладных задач характерной является обработка входного потока однотипных данных. Следовательно, обращения однотипных транзакций различных процессов обработки к одному общему элементу ресурса БД для такого класса задач являются достаточно частыми и играют ключевую роль. Таким образом, известные подходы к распараллеливанию потоковой обработки баз данных во многих случаях не позволяют вести совмещенную во времени обработку поступающих сообщений, что исключает возможность выполнения требований по оперативности обработки данных.

## 6.2. ПОДХОД К МАКСИМАЛЬНОМУ РАСПАРАЛЛЕЛИВАНИЮ ДОСТУПА К ОБЩЕЙ БАЗЕ ДАННЫХ

Рассмотрим существо предлагаемого подхода к решению этого противоречия. Каждому процессу выделим виртуальный образ ресурса БД и организуем централизованное корректное управление, как виртуальными образами, так и самим ресурсом, на основе построения **виртуальной потоковой базы данных** (ВПБД). Пусть существует множество:

$$A = \{a_1, a_2, \dots, a_i, \dots, a_I\},$$

где  $i = \overline{1, I}$ ;  $A$  - множество элементов (наборов элементов) входного потока.

Каждому элементу входного потока ставится в соответствие момент времени  $t_i$  поступления данного элемента на обработку:  $\{a_i, t_i\}$ . Кроме того, существует множество:

$$P = \{p_1, p_2, \dots, p_i, \dots, p_I\},$$

где  $i = \overline{1, I}$ ;  $P$  - является множеством процессов обработки соответствующих элементов входного потока.

Элементы входного потока поступают на обработку в различные моменты времени. Тогда, в зависимости от времени поступления, все элементы (наборы элементов) входного потока (элементы множества  $A$ ) и соответствующие им процессы обработки (элементы множества  $P$ ) могут быть разделены на:

- **старших** (поступивших на обработку ранее) и
- **младших**.

**Определение 6.2.1.** Под потоковой параллельностью обработки БД будем понимать такую обработку элементов входного потока, когда одновременно обрабатываются несколько однотипных по названию переменных, которым присвоены различные значения и при этом выполняется следующее условие: младшие процессы не могут влиять на старшие процессы, но в тоже время, старшие процессы могут вносить такие данные, которые оказывают влияние на младшие процессы.

Для каждого процесса обработки  $p_i$  требуется некоторый интервал времени  $int_i$ , необходимый для завершения данного процесса, следовательно, имеет место следующее множество пар:  $\{ \langle p_i, int_i \rangle \}$ . Одной из особенностей логической обработки входного потока является то, что в зависимости от заполнения БД, интервалы времени обработки различных процессов  $p_i$  могут различаться:

$$\exists i \neq i' : int_i \neq int_{i'} . \quad (6.2.1)$$

Если  $t_i$  - начало обработки,  $int_i$  - величина интервала времени обработки, то:  $(t_i + int_i)$  - время окончания обработки процесса  $p_i$ . Тогда:

$$\forall i \exists t_i + int_i < t_{i+1} , \quad (6.2.2)$$

т.е. время завершения обработки любого элемента входного потока меньше, чем время поступления на обработку следующего элемента входного потока. В этом случае все процессы выполняются последовательно друг за другом, а, следовательно, нет необходимости в обеспечении потоковой параллельности обработки БД.

В некоторых предметных областях, выполнение условия 6.2.2 не представляется возможным. С учетом 6.2.1 приходим к выводу, что для любого достаточно длительного интервала времени функционирования системы обработки данных всегда существует некоторое количество таких  $i$ , для которых выполняется следующее:

$$\exists i : int_i > t_{i+1} - t_i . \quad (6.2.3)$$

Таким образом, для всех АССОИ, для которых верно выражение 6.2.3 актуальной является задача обеспечения потоковой параллельности обработки БД. Учитывая 6.2.1 и при выполнении 6.2.3, обеспечение потоковой параллельности обработки БД предоставляет возможность завершения нескольких младших процессов обработки даже раньше, чем завершится обработка некоторого старшего элемента входного потока:

$$\exists i : t_{i+1} > t_i ; int_{i+1} < int_i ; t_{i+1} + int_{i+1} < t_i + int_i . \quad (6.2.4)$$

Перейдем к формальной постановке проблемы обеспечения максимальности распараллеливания доступа к общей базе данных.

### 6.3. ПОСТАНОВКА ПРОБЛЕМЫ ОБЕСПЕЧЕНИЯ МАКСИМАЛЬНОСТИ РАСПАРАЛЛЕЛИВАНИЯ ДОСТУПА К ОБЩЕЙ БАЗЕ ДАННЫХ

На основе анализа формул 6.2.1-6.2.4 представляется возможным сформулировать следующую формальную постановку проблемы.

**Формальная постановка** проблемы обеспечения максимальности распараллеливания доступа к общей базе данных (потоковой параллельной обработки БД): необходимо обеспечить такой режим функционирования АССОИ, при котором любой процесс обработки  $p_i$ , внося изменения в общую базу данных, не влиял бы на обработку старших процессов  $p_{i-1}, p_{i-2}, \dots$ , но одновременно с этим и на этой же общей БД оказывал бы влияние на обработку младших процессов  $p_{i+1}, p_{i+2}, \dots$ .

**Дано:** входной поток  $\{a_i\}$  и общая база данных **ДВ**.

**Требуется:** одновременная параллельная обработка следующих процессов  $\{ \dots, p_{i-2}, p_{i-1}, p_i, p_{i+1}, p_{i+2}, \dots \}$ , при которой любой  $p_i$  может читать и вносить изменения в общую БД **ДВ**.

**При условии, что:** изменения вносимые  $p_i$  не влияют на обработку  $\{ \dots, p_{i-2}, p_{i-1} \}$ , но влияют на обработку  $\{ p_{i+1}, p_{i+2}, \dots \}$ .

Пусть одновременно в базе данных обрабатываются несколько, например,  $k$  - элементов входного потока, при этом  $k < i$ :

$$\{a_{i-k}, a_{i-k+1}, \dots, a_{i-k+j}, \dots, a_{i-k+(k-1)}\}.$$

В таком случае, самым старшим процессом является  $a_{i-k}$ , который работает непосредственно с реальной БД. Для всех остальных действующих процессов обработки создается общая виртуальная потоковая база данных (ВПБД), в которой для каждого процесса, начиная с  $p_{i-k+1}$  и более младших  $p_{i-k+j}$  создается и ведется соответствующая персональная виртуальная потоковая база данных (ПВПБД)  $vpbd_j$ . Все изменения, которые процесс  $p_{i-k+j}$  должен внести в БД, фиксируются в его ПВПБД  $vpbd_j$  и тогда

$$\exists \{p_{i-k+j}, vpbd_j\}.$$

Процесс  $p_{i-k}$  работает непосредственно с БД, для процесса  $p_{i-k+1}$  доступ по чтению организуется непосредственно к БД, а изменения заносятся в персональную  $vpbd_1$ . Для следующего процесса  $p_{i-k+2}$  доступ по чтению производится через  $vpbd_1$ , а изменения регистрируются в  $vpbd_2$  и так далее, аналогично для всех следующих процессов.

Таким образом, младшие процессы получают доступ по чтению последовательно через все соответствующие персональные виртуальные потоковые базы данных старших процессов обработки, а все изменения вносятся только в соответствующие ПВПБД.

Получаем, что все изменения вносятся в общую БД, но младшие процессы не оказывают никакого влияния на одновременно обрабатываемые старшие процессы. Совокупность всех персональных виртуальных потоковых баз данных образует единую виртуальную потоковую базу данных, которая фактически является надстройкой над реальной БД. На рис. 6.3.1. показано формальное представление предложенного подхода в виде преобразования исходной сети Петри, реализующей обработку традиционным способом, в сеть Петри, реализующую потоковую параллельную обработку.

#### 6.4. КОНВЕЙЕРНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ФУНКЦИОНИРОВАНИЯ СЕРВЕРА БАЗЫ ДАННЫХ С МАКСИМАЛЬНЫМ РАСПАРАЛЛЕЛИВАНИЕМ

Допустим, что одновременно могут обрабатываться  $k$  процессов, а остальные накапливаются во входной очереди. При поступлении элемента  $a_i$  на обработку порождается процесс обработки  $p_i$ . Процесс  $p_i$ , поступивший на обработку раньше всех других, является самым старшим процессом, а все остальные процессы являются для него младшими.

Для каждого  $p_i$ , кроме самого старшего, создается ПВПБД  $vpbd_j$ . Как только процесс становится самым старшим, все данные из его ПВПБД переносятся в реальную БД, ресурс ПВПБД высвобождается, а вся дальнейшая обработка этого процесса ведется непосредственно с БД. Будем различать два вида ресурсов обработки:

- 1) ресурсы “процессоров” - количество одновременно обрабатываемых процессов, задач и т. п., которое зависит от количества процессоров и
- 2) ресурсы “оперативной памяти” - суммарный размер необходимой оперативной памяти для хранения и обработки значений переменных, буферов БД, ключевых файлов и т. п., которые зависят от общедоступной оперативной памяти.

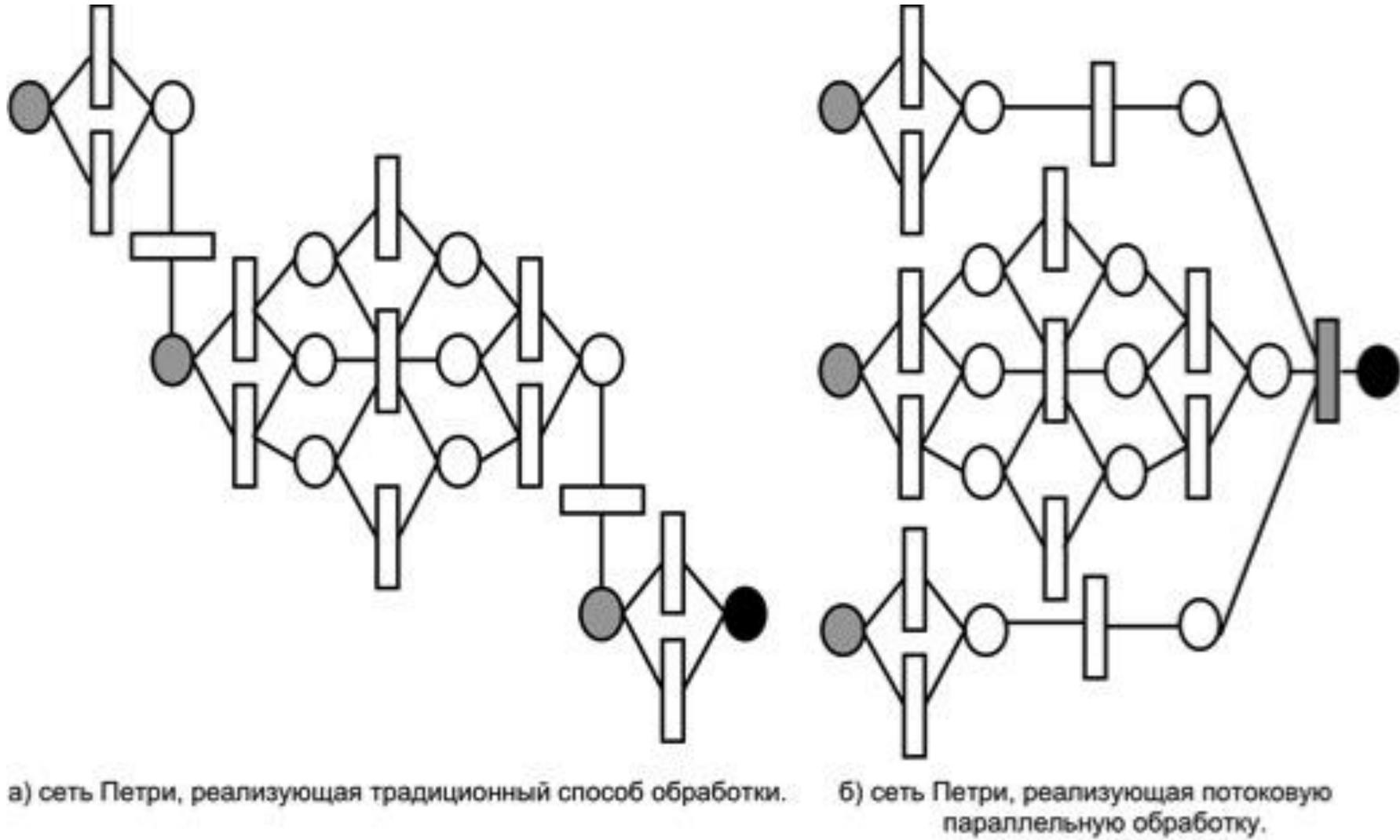


Рис. 6.3.1. Формальное представление потоковой параллельной обработки.

Отметим, что физически ВПБД не обязательно должна быть реализована только в оперативной памяти, но представляется целесообразным в дальнейшем рассматривать общий объем ВПБД и размеры конкретных ПВПБД, как ресурсы второго класса, т.е. “оперативной памяти”. Все процессы обработки порождаются последовательно во времени, т.е. одновременно не могут быть порождены два процесса и имеет место строго иерархический временной порядок порождения процессов. Следовательно, все процессы могут быть разделены относительно некоторого процесса на старших (поступивших на обработку ранее) и на младших (поступивших на обработку позже).

Все младшие процессы осуществляют доступ по чтению к БД (т.е. просматривают БД) только через соответствующие ПВПБД всех упорядоченных по старшинству более старших процессов, образуя стек типа “последним вошел, последним вышел”.

Таким образом, получаем, что, если некоторый процесс внес изменения в общую БД, то эти **изменения будут сохранены** (фактически они накапливаются только в его ПВПБД) и будут **доступны по чтению только для более младших** процессов обработки.

Если некоторый процесс, который не является самым старшим, завершает обработку, то, соответствующая ему, ПВПБД сохраняется в общей ВПБД на своем месте в стеке. Соответственно, количество хранимых ПВПБД в общей ВПБД может быть больше, чем количество функционирующих процессов обработки, причем без учета самого старшего процесса.

Подчеркнем, что порядок ПВПБД в стеке определяется только очередностью поступления на обработку. В том случае, когда самый старший процесс завершает обработку, то ПВПБД непосредственно следующего за ним функционирующего процесса, который соответственно становится самым старшим, а также все ПВПБД более старших процессов, обработка которых была завершена ранее (если таковые имеются), непосредственно загружаются в БД в порядке очередности по старшинству и освобождают ресурсы памяти для общей ВПБД.

При высвобождении необходимого ресурса, на обработку поступает очередной элемент, порождая самый младший процесс, для которого и создается соответствующая ПВПБД.

Итак, ПВПБД создается для учета вносимых изменений в БД, для хранения текущих значений переменных данного процесса, а также может использоваться в качестве буфера реальной базы данных для “подкачки” необходимых данных.

Если некоторый процесс не вносил никаких изменений в БД, то его ПВПБД, по завершению обработки и удалению текущих значений переменных, поглощается следующим за ним младшим процессом, т.е. ресурс высвобождается и такая персональная виртуальная потоковая база данных (ПВПБД) удаляется из стека общей ВПБД.

Обобщенная схема управления виртуальной потоковой базой данных ВПБД, обеспечивающая конвейеризацию обработки, показана на рис. 6.4.1.

## 6.5. ИССЛЕДОВАНИЕ ПАРАМЕТРОВ ВИРТУАЛЬНЫХ ПОТОКОВЫХ БАЗ ДАННЫХ

При исследовании параметров ВПБД будем использовать следующие обозначения:

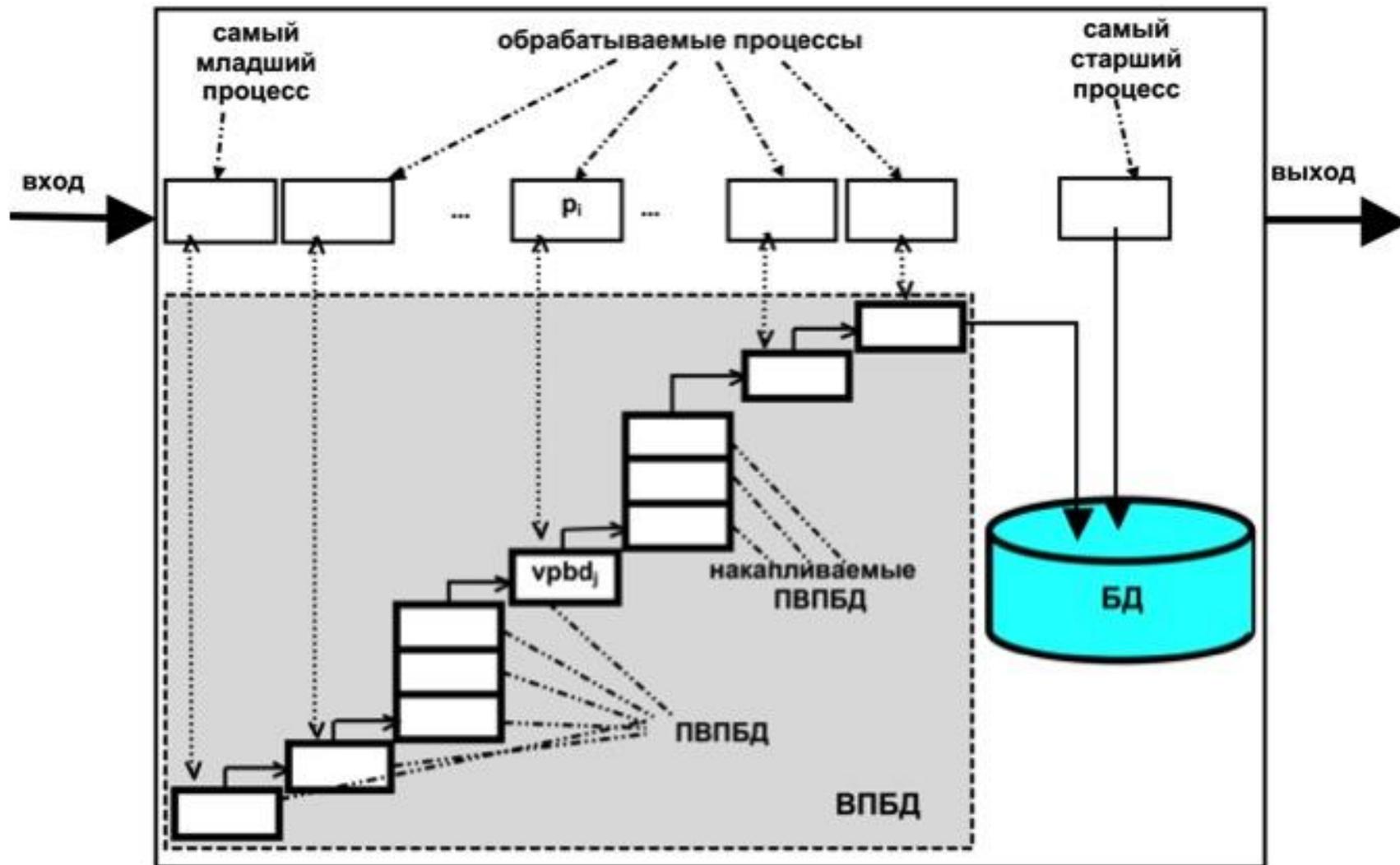


Рис. 6.4.1. Обобщенная схема управления виртуальной потоковой базой данных.

- $a_i$  - элемент входного потока;
- $p_i$  - процесс обработки элемента  $a_i$ ;
- $t_i$  - время поступления  $a_i$  на обработку;
- $d$  - размер минимального ресурса ВПБД, например страница для символьных БД;
- $B$  - максимальный размер общего объема ВПБД;
- $b$  - переменная для расчета текущего размера ресурсов ВПБД.

Будем исходить из допущения, что процессы обработки непосредственно не влияют на обработку друг друга, а логическое влияние осуществляется косвенно, посредством внесения и чтения информации из общей базы данных.

Пусть, одновременно может обрабатываться  $k$  процессов. Для элемента  $a_{i-k+j}$  создается персональная виртуальная потоковая база данных. Представляется целесообразным исходить из того, что размер любой ПВПБД должен быть кратен  $d$ . Тогда введем переменную:  $c_j$  - количество минимальных ресурсов  $d$  в соответствующей ПВПБД  $vpbd_j$ . Следовательно, имеет место формула:

$$vpbd_j = d \times c_j \quad (6.5.1)$$

и тогда:

$$b = \sum_{j=1}^k vpbd_j = \sum_{j=1}^k d \times c_j, \quad (6.5.2)$$

при условии, что:

$$b \leq B. \quad (6.5.3)$$

Отметим, что практически  $c_j$  - это опосредованная характеристика сложности конкретного процесса обработки  $p_i$  и представляется необходимым ввести следующие ограничения:

$$c_{\min} \leq c_j \leq c_{\max}. \quad (6.5.4)$$

Исходя из того, что:  $c_{\min} \rightarrow 1$ , представляется возможным записать:

$$c_{\min} = 1. \quad (6.5.5)$$

Анализ различных БД показал, что практически, для упрощения вычислений, можно разбить все процессы на две категории: простые и сложные. Для простых процессов:  $c_j \rightarrow c_{\min}$ , а для сложных процессов:  $c_j \rightarrow c_{\max}$  и при этом справедливо, что:

$$c_{\min} \ll c_{\max}. \quad (6.5.6)$$

Перейдем к рассмотрению различных видов входных потоков. Прежде всего, исследуем ресурсы необходимые только для обработки.

Рассмотрим первый вариант, при котором все элементы потока порождают только простые процессы обработки, т.е.  $c_j \rightarrow c_{\min} = 1$ . В таком случае требуемые ресурсы будут минимальными  $b = b^{\min}$  и тогда:

$$b^{\min} = \sum_{j=1}^k d \times c_j = k \times d. \quad (6.5.7)$$

Получаем линейную зависимость размера  $b$  от количества одновременно обрабатываемых процессов  $k$  с коэффициентом  $d$ .

Рассмотрим вариант, при котором весь поток состоит из таких элементов, которые порождают сложные процессы, т.е.  $c_j \rightarrow c_{\max}$ . Для такого варианта потребуются максимальные ресурсы при обработке  $b = b^{\max}$  и тогда:

$$\mathbf{b}^{\max} = \sum_{j=1}^k \mathbf{d} \times \mathbf{c}_{\max} = \mathbf{k} \times \mathbf{d} \times \mathbf{c}_{\max}. \quad (6.5.8)$$

Для данного типа потока также получаем линейную зависимость, но с большим коэффициентом:  $\mathbf{d} \times \mathbf{c}_{\max}$ .

В третьем варианте рассмотрим одновременную обработку только некоторого количества сложных процессов - обозначаемое в дальнейшем:  $\mathbf{f}$  и всех остальных простых процессов в количестве  $(\mathbf{k} - \mathbf{f})$ . Тогда имеет место следующая формула:

$$\mathbf{b} = \sum_{j=1}^k \mathbf{d} \times \mathbf{c}_j = \mathbf{f} \times \mathbf{d} \times \mathbf{c}_{\max} + (\mathbf{k} - \mathbf{f}) \times \mathbf{d} \times \mathbf{c}_{\min} = \mathbf{d}(\mathbf{f} \times \mathbf{c}_{\max} + (\mathbf{k} - \mathbf{f})). \quad (6.5.9)$$

Для области допустимых значений первых трех вариантов получили линейную зависимость с различными коэффициентами, где нижняя граница описывается формулой 6.5.7, а верхняя граница - формулой 6.5.8 и при этом всегда верно, что  $\mathbf{b}^{\min} \leq \mathbf{b} \leq \mathbf{b}^{\max}$ .

Но в этих вариантах не рассматривались такие случаи, когда сложные и простые процессы обрабатываются в общем потоке, перемешиваясь между собой и младшие простые процессы могут завершать обработку ранее, чем старшие сложные процессы. Ресурс "памяти" общей ВПБД должен расходоваться не только непосредственно на обработку, но и для накопления информации (данных) уже завершенных младших процессов (тех которые младше самого старшего обрабатываемого процесса).

Таким образом, необходимо учесть накапливание в общей ВПБД персональных  $\mathbf{vpbd}_i$ , где  $\mathbf{I} = \mathbf{1}, \mathbf{L}$ , а  $\mathbf{L}$  - максимальное количество накапливаемых ПВПБД, которое в общем случае зависит от потока входных элементов. Тогда:

$$\mathbf{V} = \mathbf{b}_{\text{ОБР}} + \mathbf{b}_{\text{НАК}}, \quad (6.5.10)$$

где  $\mathbf{b}_{\text{ОБР}}$  - объем памяти ВПБД, необходимый для обрабатываемых процессов;  $\mathbf{b}_{\text{НАК}}$  - дополнительный объем памяти ВПБД, необходимый для хранения ПВПБД тех процессов, обработка которых уже завершена, но которые должны храниться в стеке.

Принимая во внимание, что  $\mathbf{c}_{\min} = \mathbf{1}$  и с учетом формулы 6.5.9, получаем:

$$\mathbf{b}_{\text{ОБР}} = \mathbf{d}(\mathbf{k} + \mathbf{f}(\mathbf{c}_{\max} - \mathbf{1})). \quad (6.5.11)$$

Аналогично формуле 6.4.2 можно записать, что:

$$\mathbf{b}_{\text{НАК}} = \sum_{i=1}^{\mathbf{L}} \mathbf{vpbd}_i = \sum_{i=1}^{\mathbf{L}} \mathbf{d} \times \mathbf{c}_i, \quad (6.5.12)$$

где  $\mathbf{c}_i$  - аналогично  $\mathbf{c}_j$ , определяет количество минимальных ресурсов  $\mathbf{d}$  в хранимой ПВПБД  $\mathbf{vpbd}_i$ . Подчеркнем, что переменные  $\mathbf{I}$  и  $\mathbf{L}$  не зависят от  $\mathbf{j}$  и  $\mathbf{k}$ , более того вполне может иметь место и то, что  $\mathbf{L} > \mathbf{k}$ . Отметим, что  $\mathbf{b}_{\text{НАК}}$  - это дополнительные расходы памяти, которые являются платой за возможность проведения параллельной потоковой обработки баз данных. Буфер накапливаемых ПВПБД необходим только тогда, когда во входном потоке непосредственно за "сложным" элементом обработки (т.е. порождающим сложный процесс) следует некоторое количество "простых" элементов. Пусть сложному процессу, требующему  $\mathbf{c}_{\max}$  минимальных ресурсов соответствует интервал времени обработки  $\mathbf{int}_{\max}$ , а простому процессу, потребляющему  $\mathbf{c}_{\min}$  соответствует -  $\mathbf{int}_{\min}$ .

Как уже отмечалось выше, представляется возможным и целесообразным рассматривать случай, когда на обработке находятся только сложные и простые процессы, т.е. всего две степени сложности процесса: простые и сложные. Все процессы поступают на обработку последовательно, тогда получаем, что:

- старший сложный процесс не может завершить обработку ранее младшего простого процесса;
- старший сложный процесс не может завершить обработку ранее младшего сложного процесса;
- старший простой процесс не может завершить обработку ранее младшего простого процесса;
- старший простой процесс **может** завершить обработку ранее младшего сложного процесса.

Следовательно, накапливаться могут только данные младших процессов и:

$$\forall l : c_l = c_{\min} = 1. \quad (6.5.13)$$

Допустим, что сложные процессы не следуют непосредственно друг за другом, а между ними всегда есть несколько простых процессов, которые могут накапливаться в буфере ВПБД. Если за одним сложным следует достаточно большое количество простых процессов, то накапливаться будут только те простые процессы, которые завершают обработку раньше непосредственно старшего сложного процесса.

Пусть  $e_{\max}$  - максимальное количество таких простых процессов. Если интервал времени между последовательным поступлением на обработку любых двух элементов входного потока много меньше, чем минимальный интервал времени обработки процесса, т.е.  $\forall i : (t_{i+1} - t_i) \ll \text{int}_{\min}$ , то величиной  $(t_{i+1} - t_i)$  можно пренебречь и тогда:

$$e_{\max} \approx \frac{\text{int}_{\max}}{\text{int}_{\min}}. \quad (6.5.14)$$

Если на обработке находится всего  $F$  сложных процессов, то обозначим через  $e_f$  - количество накапливаемых простых процессов следующих непосредственно за некоторым сложным процессом  $f$ , где  $f = \overline{1, F}$ . Тогда получаем, что:

$$L = \sum_{f=1}^F e_f \leq F \times e_{\max}. \quad (6.5.15)$$

Анализ различных предметных областей показал, что имеет место зависимость интервала времени обработки от степени ее сложности. Чем сложнее процесс обработки, тем больше требуется для него ресурса памяти. Таким образом, переменная  $c_j$  представляет собой не только количество минимальных ресурсов, но и является опосредованной **характеристикой степени сложности** процесса обработки.

Более того, представляется обоснованным допустить, учитывая формулу 6.5.13, что

$$e_{\max} \approx \frac{\text{int}_{\max}}{\text{int}_{\min}} = \frac{c_{\max}}{c_{\min}} = c_{\max}. \quad (6.5.16)$$

С учетом формул 6.5.13, 6.5.14 и 6.5.15, формула 6.5.12 может быть преобразована:

$$b_{\text{НАК}} = \sum_{l=1}^L d \times c_l = \sum_{l=1}^L d \times c_{\min} = L \times d \times c_{\min} = d \times L = d \times \sum_{f=1}^F e_f. \quad (6.5.17)$$

Представляется необходимым получить выражение, определяющее максимальный размер требуемого накапливаемого ресурса памяти для некоторого количества сложных процессов  $f$ , обозначаемого:  $b_{\text{НАК}}^{f \max}$ . Тогда получаем:

$$b_{\text{НАК}}^{f \max} = d \times \sum_{f=1}^F e_{\max} = d \times \sum_{f=1}^F c_{\max} = d \times F \times c_{\max}. \quad (6.5.18)$$

Следовательно, получаем общее выражение для ресурса памяти ВПБД:

$$\mathbf{B} = \mathbf{b}_{\text{ОБР}} + \mathbf{b}_{\text{НАК}} = \mathbf{d} \times (\mathbf{k} + \mathbf{F}(\mathbf{c}_{\text{max}} - 1)) + \mathbf{d} \times \sum_{f=1}^{\mathbf{F}} \mathbf{e}_f. \quad (6.5.19)$$

Перейдем к оценке минимального и максимального требуемого ресурса в зависимости от потока входных элементов. Максимальное использование ресурса памяти будет при таком входном потоке, когда за каждым сложным процессом следует  $\mathbf{e}_{\text{max}}$  простых процессов. Максимальное количество одновременно обрабатываемых сложных процессов, при условии накопления между ними результатов обработки простых процессов будет стремиться к максимальному числу одновременно обрабатываемых процессов:  $\mathbf{F} \rightarrow \mathbf{k}$ , а если учесть, что

$$\forall i : (\mathbf{t}_{i+1} - \mathbf{t}_i) \ll \text{int}_{\text{min}}, \quad (6.5.20)$$

то величиной  $(\mathbf{t}_{i+1} - \mathbf{t}_i)$  можно пренебречь и тогда  $\mathbf{F} = \mathbf{k}$ . Следовательно:

$$\mathbf{b}_{\text{ОБР}}^{\text{max}} = \mathbf{k} \times \mathbf{d} \times \mathbf{c}_{\text{max}}, \quad (6.5.21)$$

$$\mathbf{b}_{\text{НАК}}^{\text{max}} = \mathbf{d} \times \mathbf{F} \times \mathbf{c}_{\text{max}} = \mathbf{d} \times \mathbf{k} \times \mathbf{c}_{\text{max}}. \quad (6.5.22)$$

Тогда получаем:

$$\mathbf{B}^{\text{max}} = \mathbf{b}_{\text{ОБР}}^{\text{max}} + \mathbf{b}_{\text{НАК}}^{\text{max}} = 2 \times \mathbf{k} \times \mathbf{d} \times \mathbf{c}_{\text{max}}, \quad (6.5.23)$$

$$\mathbf{B}^{\text{min}} = \mathbf{b}_{\text{ОБР}}^{\text{min}} + 0 = \mathbf{k} \times \mathbf{d}. \quad (6.5.24)$$

Таким образом, получены, показанные на рис. 6.5.1, нижняя и верхняя границы размеров буфера виртуальных потоковых баз данных:

$$\mathbf{k} \times \mathbf{d} \leq \mathbf{B} \leq 2 \times \mathbf{k} \times \mathbf{d} \times \mathbf{c}_{\text{max}}, \quad (6.5.25)$$

Оценим реализационные перспективы предложенной ВПБД. На практике, представляется целесообразным допустить следующие исходные значения:  $\mathbf{d} \approx 1 \text{ Кб}$ ,  $\mathbf{c}_{\text{max}} \approx 1000$ ,  $\mathbf{k} = 10$ . Тогда верхняя граница значений  $\mathbf{B}^{\text{max}} \approx 10 \text{ Мб}$ , что для современных ЭВМ представляется реализуемым. В дальнейшем, представляется возможным уменьшать верхнюю границу следующими путями: уменьшение размера минимального ресурса, за счет оптимизации структур хранения данных ВПБД и уменьшение сложности выполняемых процессов, т.е. величины  $\mathbf{c}_{\text{max}}$ , за счет совершенствования алгоритмов обработки.

## 6.6. СТРУКТУРЫ ПРЕДСТАВЛЕНИЯ ДАННЫХ ДЛЯ ВИРТУАЛЬНЫХ ПОТОКОВЫХ БАЗ ДАННЫХ

Виртуальные потоковые базы данных должны работать с любыми традиционными БД, независимо от их моделей данных. Ключевой проблемой при этом является разработка таких структур представления данных для виртуальных потоковых баз данных, которые были бы совместимы, как минимум, со всеми основными СПД традиционных моделей данных (МДн) [Л. 5-13, 17, 20-28, 31-40, 43-48, 54-57, 62-66, 71-80, 84, 88-98, 107-118, 120-132, 135-152, 155-167, 170-180, 189-216, 219, 224-228, 231-253, 255-258, 261-264, 268-271, 274-280, 284-289, 292-300, 304-328, 331, 333, 337-340, 342-413, 492-501].

$$B^{\max} = b_{\text{ОБР}}^{\max} + b_{\text{НАК}}^{\max} = 2 \times k \times d \times c_{\max},$$

$$B^{\min} = b_{\text{ОБР}}^{\min} + 0 = k \times d.$$

Нижняя и верхняя границы размера буфера ВПБД:  $k \times d \leq B \leq 2 \times k \times d \times c_{\max}$ .

Нижняя и верхняя границы размера буфера ВПБД:  $k \times d \leq B \leq 2 \times k \times d \times c_{\max}$ .

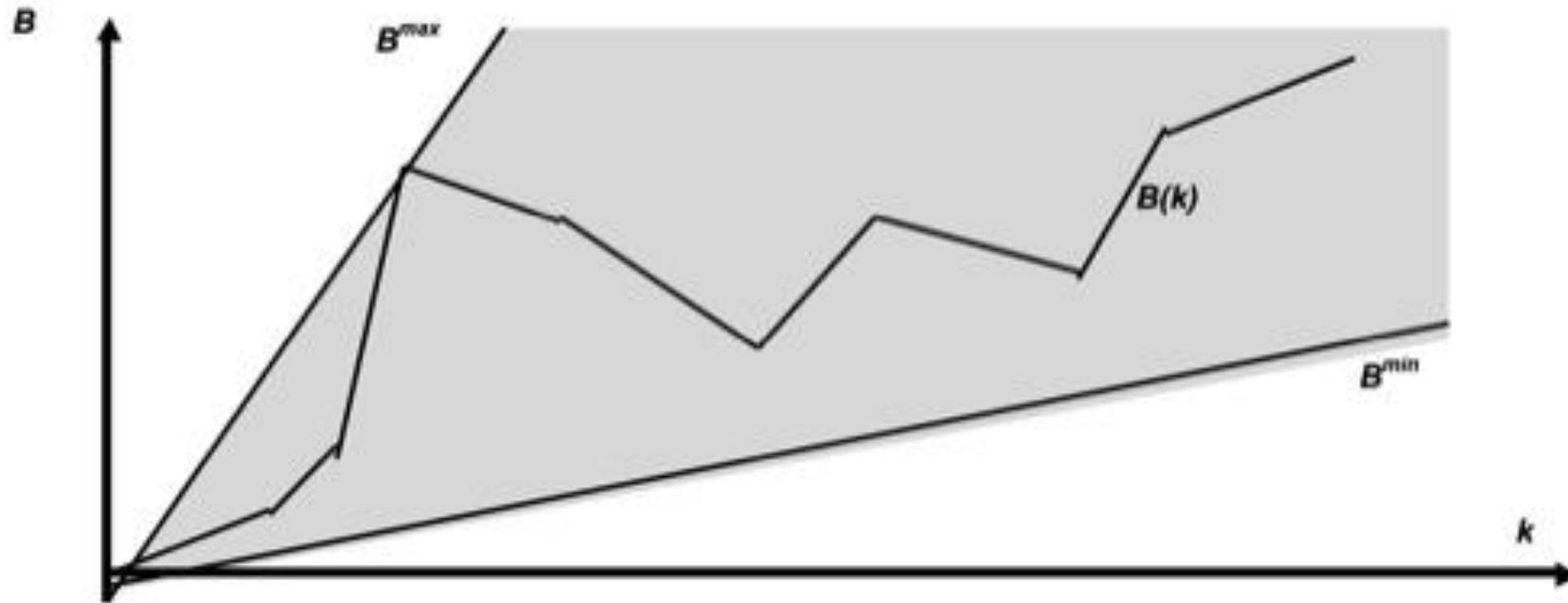


Рис. 6.5.1. Исследование параметров виртуальных потоковых баз данных.

В настоящее время проблема совместимости СПД различных моделей данных решается либо на концептуальном уровне, либо путем построения специальных преобразователей из СПД одной МДн в СПД другой, например - из реляционной МДн в сетевые СПД [Л. 19, 27-28, 87, 91, 100-104, 120-124, 183-185, 190, 205, 216, 247, 263, 293, 296, 305, 314-316, 319, 322, 326, 359, 403]. При решении задачи обеспечения параллельности потоковой обработки БД, основанных на концепции клиент-сервер, не представляется возможным гарантировать однородность БД. Более того, исходя из независимости отдельных фрагментов БД, следует, что пользователь любого автоматизированного рабочего места (АРМ) АССОИ может работать с несколькими БД, поддерживающими различные модели данных. Следовательно, ВПБД необходимо применять с гетерогенными БД на основе универсальных СПД.

Отметим, что **пятиуровневые ОТПД**, как было показано выше, позволяют описывать (даже для модели "сущность-связь") как сущности, так и их связи в едином формате. Такое представление позволяет в явном виде сохранять или определять отличия связей от объектов, а также деление по категориям типов, категориям вершин, классам сущностей, категориям дуг и классам связей. Следовательно, наиболее общей формой представления данных в виде таблиц для традиционных моделей данных является формализм одномерных таблиц пятиуровневого представления данных (ОТПД-5).

Кроме табличного описания данных представляется возможным использовать **миварный подход**, основанный на использовании многомерного дискретного информационного пространства унифицированного представления данных. Конкретные значения точек миварного пространства, размерность миварного пространства и структуры конкретных таблиц представления данных зависят от моделируемых предметных областей, а также от возможностей программного и аппаратного обеспечения. Теоретически, никаких ограничений не накладывается. Более того, размерности различных таблиц в одном пространстве могут различаться, предоставляется возможность для каждого пользователя на основе единой общей БД создавать свои индивидуальные представления и образы предметной области. В качестве служебной информации о каждом миваре можно хранить ограничения по доступу пользователей.

Итак, если обработка данных для решения сложных задач реализуется в распределенной АССОИ на основе нескольких гетерогенных (разномодельных) баз данных, то в качестве основы для создания ВПБД предлагается использовать, либо одномерные таблицы представления данных, либо миварный подход к представлению данных, в зависимости от наличия вычислительных ресурсов, требований описываемой предметной области и классов решаемых задач.

## 6.7. ОЦЕНКА БЫСТРОДЕЙСТВИЯ ОБРАБОТКИ ПОТОКА ДАННЫХ

Каждому процессу обработки  $p_i$  соответствует некоторый интервал времени обработки  $int_i$ . Процесс обработки состоит из последовательности различных операций, которые представляется целесообразным разделить на два класса:

1. **ОР** - операции без обращения к БД, т.е. только в оперативной памяти;
2. **ВД** - операции, требующие обращения к базе данных.

Введем следующие обозначения:

- $int\ op_{iop}$  - интервал времени выполнения одной операции  $iop$  первого класса, где  $iop = \overline{1, IOP}$ ; и

- $\mathbf{intbd}_{ibd}$  - интервал времени выполнения одной операции  $\mathbf{ibd}$  второго класса, где  $\mathbf{ibd} = \overline{\mathbf{1, IBD}}$ .

Тогда получаем, что в общем случае:

$$\mathbf{int}_i = \sum_{iop=1}^{IOP} \mathbf{int op}_{iop} + \sum_{ibd=1}^{IBD} \mathbf{int bd}_{ibd}. \quad (6.7.1)$$

Если обозначить суммарное время обработки процессов от  $i$  до  $i+N$  через -  $\mathbf{INT}$ , тогда для обработки в общем случае будет верно, что:

$$\mathbf{INT} = \sum_{n=0}^N \mathbf{int}_{i+n} = \sum_{n=0}^N \left( \sum_{iop_n=1}^{IOP_n} \mathbf{int op}_{iop_n} + \sum_{ibd_n=1}^{IBD_n} \mathbf{int bd}_{ibd_n} \right). \quad (6.7.2)$$

При обычной последовательной обработке интервал времени обработки, обозначаемый как  $\mathbf{INT}_1$ , будет выражаться следующим образом:

$$\mathbf{INT}_1 = \sum_{n=0}^N \sum_{iop_n=1}^{IOP_n} \mathbf{int op}_{iop_n} + \sum_{n=0}^N \sum_{ibd_n=1}^{IBD_n} \mathbf{int bd}_{ibd_n}. \quad (6.7.3)$$

При функциональной параллельной обработке в оперативной памяти могут выполняться одновременно несколько или даже все операции одного процесса  $\mathbf{iop}_n$ . Исходя из анализа особенностей логической обработки и предметной области, можно сделать вывод, что первая и последняя операции любого процесса относятся к первому классу и не могут выполняться одновременно, так как между ними обязательно существует хотя бы одна операция второго класса, т.е. обращение к БД.

Следовательно, необходимо ввести некоторый коэффициент обработки  $\mathbf{kobr}_n^2$ , который будет определяться временем выполнения алгоритмически последовательных операций каждого конкретного процесса  $\mathbf{iop}_n$ . Из всех операций, которые выполняются одновременно необходимо выбрать максимальную по времени обработки. Тогда, получаем, что  $\mathbf{int}_i^2$  - интервал времени при функциональной параллельной обработке в оперативной памяти процесса обработки будет иметь следующий вид:

$$\mathbf{int}_i^2 = \left( \mathbf{kobr}_n^2 + \mathbf{MAX}_{iop_n=2}^{IOP_n-1} \mathbf{int op}_{iop_n} \right) + \sum_{ibd_n=1}^{IBD_n} \mathbf{int bd}_{ibd_n}. \quad (6.7.4)$$

Сравнивая выражения 6.7.1 и 6.7.4 приходим к выводу, что

$$\mathbf{int}_i^2 \leq \mathbf{int}_i. \quad (6.7.5)$$

При функциональной параллельной обработке интервал времени обработки, обозначаемый как  $\mathbf{INT}_2$ , будет выражаться следующим образом:

$$\mathbf{INT}_2 = \sum_{n=0}^N \left( \mathbf{kobr}_n^2 + \mathbf{MAX}_{iop_n=2}^{IOP_n-1} \mathbf{int op}_{iop_n} \right) + \sum_{n=0}^N \sum_{ibd_n=1}^{IBD_n} \mathbf{int bd}_{ibd_n}. \quad (6.7.6)$$

Перейдем к исследованию параллельной обработки данных, при которой могут выполняться одновременно несколько или даже все операции одного процесса  $\mathbf{iop}_n$  как первого, так и второго классов. Аналогично получаем, что  $\mathbf{int}_i^3$  - интервал времени при параллельной обработке данных процесса обработки будет иметь следующий вид:

$$\mathbf{int}_i^3 = \left( \mathbf{kobr}_n^3 + \mathbf{MAX}_{iop_n=2}^{IOP_n-1} \mathbf{int op}_{iop_n} \right) + \left( \mathbf{MAX}_{ibd_n=1}^{IBD_n} \mathbf{intbd}_{ibd_n} \right). \quad (6.7.7)$$

Сравнивая выражения 6.7.1, 6.7.4 и 6.7.7 приходим к выводу, что

$$\mathit{int}_i^3 \leq \mathit{int}_i^2 \leq \mathit{int}_i. \quad (6.7.8)$$

Следовательно, при параллельной обработке данных суммарное время обработки всех процессов от  $i$  до  $i + N$ , обозначаемое как  $\mathit{INT}_3$ , будет выражаться следующим образом:

$$\mathit{INT}_3 = \sum_{n=0}^N \left\{ \left( \mathit{kobr}_n^2 + \mathit{MAX}_{iop_n=2}^{iop_n-1} \mathit{intop}_{iop_n} \right) + \mathit{MAX}_{ibd_n=1}^{IBD_n} \mathit{intbd}_{ibd_n} \right\}. \quad (6.7.9)$$

При поточковой параллельной обработке данных одновременно могут выполняться операции первого и второго классов нескольких процессов. Тогда, при потоковой параллельной обработке суммарное время обработки всех процессов от  $i$  до  $i + N$ , обозначаемое как  $\mathit{INT}_4$ , с учетом формулы 6.5.20, будет выражаться следующим образом:

$$\mathit{INT}_4 = \mathit{MAX}_{n=0}^N \left\{ \mathit{kobr}_n^4 + \mathit{MAX}_{iop_n=2}^{iop_n} \mathit{intop}_{iop_n} + \mathit{MAX}_{ibd_n=1}^{IBD_n} \mathit{intbd}_{ibd_n} \right\}. \quad (6.7.10)$$

Суть всех четырех способов организации множественного доступа параллельных процессов к общей базе данных показана на рис. 6.7.1. Если использовать краткие обозначения, то можно получить следующие символические формулы:

$$\begin{aligned} \mathit{INT}_1 &\approx \Sigma(\Sigma + \Sigma), \\ \mathit{INT}_2 &\approx \Sigma(\mathit{MAX} + \Sigma), \\ \mathit{INT}_3 &\approx \Sigma(\mathit{MAX} + \mathit{MAX}), \\ \mathit{INT}_4 &\approx \mathit{MAX}(\mathit{MAX} + \mathit{MAX}). \end{aligned}$$

Проводя анализ и сравнение формул 6.7.8, 6.7.9 и 6.7.10 получаем, что:

$$\mathit{INT}_4 \leq \mathit{INT}_3 \leq \mathit{INT}_2 \leq \mathit{INT}_1.$$

Исходя из опыта практической деятельности, всегда выделяют некоторый интервал максимально допустимого времени обработки (задержки обработки) потока входных данных. Будем обозначать такой интервал как  $\mathit{INT}_{\text{допустимый}}$ .

Качественное сравнение зависимости интегрального времени обработки потока входных данных от способа организации множественного доступа к общей базе данных в условиях существования временной зависимости между обрабатываемыми элементами входного потока показано на рис. 6.7.2. Анализ приведенных зависимостей показывает, что при , всегда наступает такой момент времени, когда

$$\mathit{INT}_1 > \mathit{INT}_2 > \mathit{INT}_3 \gg \mathit{INT}_{\text{допустимый}},$$

но только при использовании ВПБД для:

$$\forall t_i \quad \mathit{INT}_4 < \mathit{INT}_{\text{допустимый}}.$$

Итак, только метод **максимального распараллеливания доступа к общей базе данных**, основанный на потоковой параллельной обработке данных с использованием виртуальных потоковых баз данных, обеспечивает такое **быстродействие** с минимальной "не возрастающей" (фиксированной) задержкой, которое необходимо для решения различных задач АССОИ **в режиме времени, близком к реальному**. Перейдем к рассмотрению еще одного быстродействующего метода обработки данных при решении специальных задач в АССОИ, основанного на полно-переборном суммировании чисел.

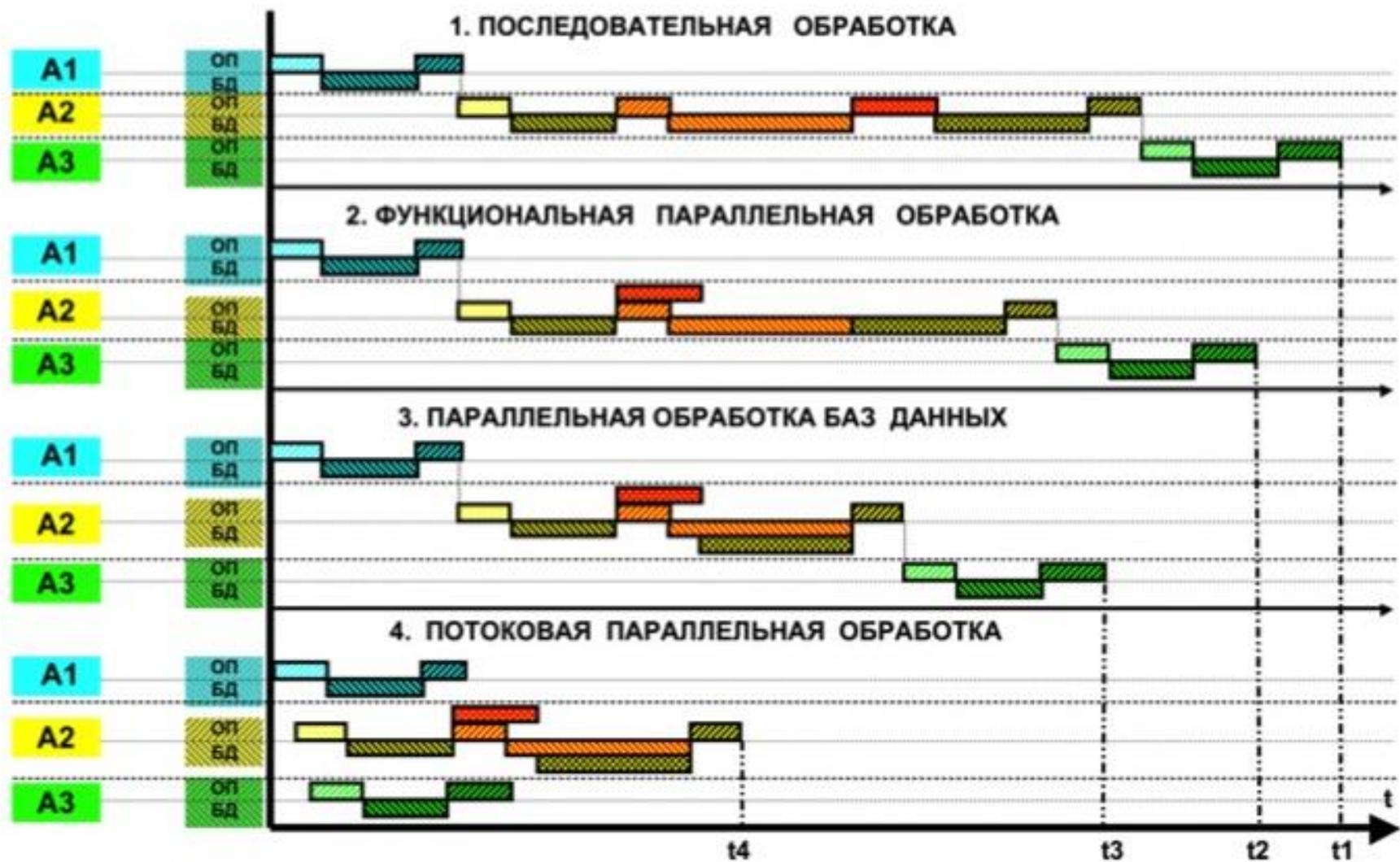


Рис. 6.7.1. Организация множественного доступа параллельных процессов к общей БД.

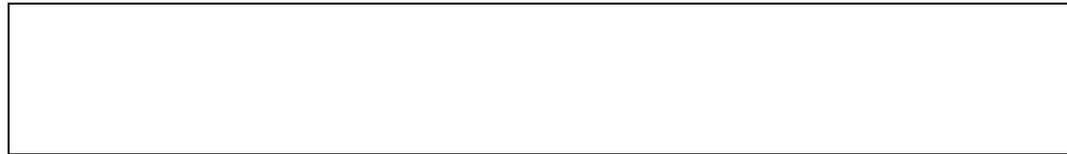
При последовательной обработке:



При функциональной параллельной обработке:

$$INT_2 = \sum_{n=0}^N \left( kobr_n^2 + \text{MAX}_{iop_n=2}^{IOP_{n-1}} \text{int op}_{iop_n} \right) + \sum_{n=0}^N \sum_{ibd_n=1}^{IBD_n} \text{int bd}_{ibd_n} .$$

При параллельной обработке данных:



При поточковой параллельной обработке данных одновременно могут выполняться все операции нескольких процессов:

$$INT_4 = \text{MAX}_{n=0}^N \left\{ kobr_n^4 + \text{MAX}_{iop_n=2}^{IOP_n} \text{int op}_{iop_n} + \text{MAX}_{ibd_n=1}^{IBD_n} \text{int bd}_{ibd_n} \right\}$$

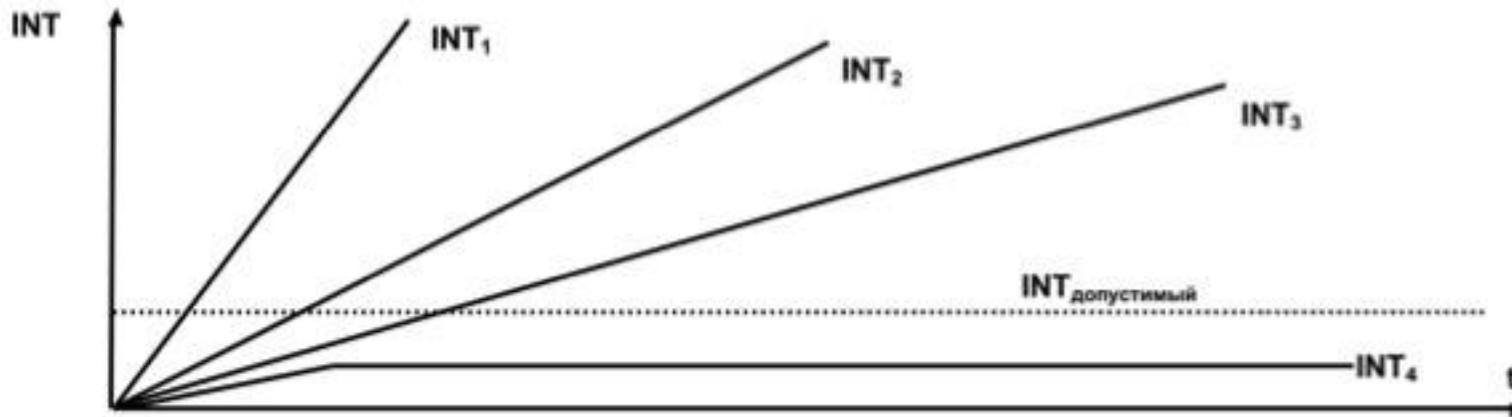


Рис. 6.7.2. Исследование зависимости интегрального времени обработки потока данных от способа организации множественного доступа параллельных процессов к общей базе данных.

## 7. ЛИНЕЙНЫЙ МЕТОД ЕДИНИЧНО-ИНКРЕМЕНТНОГО СУММИРОВАНИЯ ЧИСЕЛ

### 7.1. НЕОБХОДИМОСТЬ РАЗРАБОТКИ ЛИНЕЙНОГО МЕТОДА СУММИРОВАНИЯ ЧИСЕЛ

В настоящее время актуальной является задача повышения быстродействия ЭВМ при обработке специальной информации. Одним из направлений повышения оперативности обработки специальной информации является разработка новых методов решения целочисленных задач. Известно, что если алгоритмы шифрования идеальны и оптимальным методом их взлома является прямой перебор всех возможных ключей, то стойкость криптосистем определяется длиной ключа [Л. 18, 245]. Определив отношение количества ключей к быстродействию самого мощного компьютера, получают нижнюю оценку сложности дешифрования сообщения для идеального алгоритма. Оценка времени, необходимого самым мощным суперкомпьютерам для полного перебора ключей, приведена в [Л. 245]. Полный перебор ключей может осуществляться на основе полного переборного единично-инкрементного суммирования чисел, т.е. при **P**-ичном суммировании **n**-разрядного числа **X** с массивом **V** **n**-разрядных чисел от **0** до **P<sup>n</sup>**.

Общее количество операций сложения при этом определяют, как степенную зависимость общего количества операций сложения от разрядности чисел, а предлагаемый линейный метод единично-инкрементного суммирования чисел позволяет перейти от степенной к линейной зависимости. Линейный метод единично-инкрементного суммирования чисел [Л. 225] предназначен для решения задачи суммирования **n**-разрядного числа **X** с массивом чисел от **X<sub>A</sub>** до **X<sub>B</sub>**, где:

$$0 \leq X_A \leq X_B \leq P^n. \quad (7.1.1)$$

Путем разбиения исходного числа на блоки, нового поблочного суммирования, анализа и формирования требуемых чисел, линейный метод суммирования чисел позволяет уменьшить количество операций сложения при **P**-ичном суммировании **n**-разрядного числа **X** с массивом **V** **n**-разрядных чисел от **0** до **P<sup>n</sup>**, повысить интегральное быстродействие суммирования чисел, а также перейти от степенной к линейной зависимости общего количества операций сложения от разрядности чисел.

Актуальность и необходимость разработки линейного метода единично-инкрементного (переборного) суммирования чисел определяется тем, что его практическая реализация может привести к существенному сокращению необходимого количества сумматоров в ЭВМ, уменьшению разрядности этих сумматоров и значительному повышению оперативности обработки специальной информации [Л. 225].

Анализ методов решения подобных задач показал, что существует метод сложения двоичных чисел, в котором, с целью ускорения процесса суммирования, формируют два слова: сумму по модулю 2 исходных слагаемых и конъюнкцию исходных слагаемых со сдвигом на один разряд влево, после чего одноименные разряды полученных слов разбивают на группы специальным образом [Л. 3]. Недостатком этого метода является то, что он применим только для двоичных чисел и, ускоряя каждую отдельную операцию сложения, не позволяет уменьшить количество операций сложения, уменьшить разрядность сумматоров и повысить быстродействие суммирования чисел.

Наиболее близким по сущности к предлагаемому линейному методу является метод суммирования чисел, при котором суммирование производят параллельно во всех группах разрядов, на которые разделен сумматор, при этом в каждую группу разрядов условно прибавляют “единицу” или “нуль” переноса, а затем осуществляют выбор окончательного результата суммирования [Л. 4]. Однако, недостатком этого метода является то, что он повышает только быстродействие каждой операции сложения, но не позволяет уменьшить количество операций сложения при полном переборном  $P$ -ичном суммировании, уменьшить разрядность сумматоров, а также перейти от степенной к линейной зависимости общего количества операций сложения от разрядности чисел. Рассмотрим основы предлагаемого нового линейного метода единично-инкрементного суммирования чисел.

## 7.2. ОСНОВЫ ЛИНЕЙНОГО МЕТОДА ЕДИНИЧНО-ИНКРЕМЕНТНОГО СУММИРОВАНИЯ ЧИСЕЛ

Сущность линейного метода единично-инкрементного суммирования чисел заключается в следующем. Прежде всего, представляется возможным разбить исходное число на равные группы. При каждом добавлении единицы, как показано в таблице 7.2.1, все основные операции суммирования осуществляются только в первой группе самых младших разрядов чисел (разряды 1-4 в таблице 7.2.1). Если нет единицы переноса (разряды 5-16 в строках 0-15 таблицы 7.2.1), которая формируется только при завершении цикла суммирования в группе (5-ый разряд 16-ой строки или 6-ой разряд 32-ой строки таблицы 7.2.1), то в более старших разрядах ничего не изменяется. Следовательно, в таком случае суммирование групп старших разрядов можно не производить и сократить требуемое количество операций сложения.

Основной выигрыш предлагаемого метода единично-инкрементного суммирования чисел состоит в том, что при переборном сложении достаточно больших чисел (например, 128, 256 и более разрядов) “лишние”, т.е. фактически “невыполняемые” операции сложения старших разрядов чисел (например, разряды с 7 по 16 в таблице 7.2.1), алгоритмически блокируются и не выполняются. Рассмотрим краткий алгоритм - последовательность действий, необходимых для реализации предлагаемого нового метода единично-инкрементного суммирования чисел.

Сущность предлагаемого алгоритма реализации линейного метода единично-инкрементного суммирования чисел заключается в выполнении следующих действий:

- исходное  $n$ -разрядное  $P$ -ичное число  $X$  разбивают на  $k$  блоков  $X_1, \dots, X_i, \dots, X_k$  по  $m$  разрядов в каждом;
- производят параллельное на нескольких сумматорах или последовательное на одном сумматоре добавление единицы к каждому числу  $X_1, \dots, X_i, \dots, X_k$  с формированием в  $(m+1)$ -ом разряде “единиц” или “нулей” переноса;
- полученные числа записывают в массивы  $Z_1, \dots, Z_i, \dots, Z_k$  соответственно;
- повторяют эти действия  $P^m$  раз и получают  $(m+1)$ -разрядные массивы  $Z_1, \dots, Z_i, \dots, Z_k$ , содержащие в каждом  $P^m$   $P$ -ичных чисел;

№	Разряды															
	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
14	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
15	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
18	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
19	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
20	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
21	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1
22	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
23	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
24	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
26	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0
27	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
28	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
29	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
30	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
31	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
32	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
34	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
35	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
36	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

- формируют строки требуемого массива  $Y$ :
  - в качестве текущих значений в каждом массиве  $Z_1, \dots, Z_i, \dots, Z_k$  выбирают первые числа;
  - записывают выбранные числа в первую строку и переходят к формированию следующей строки требуемого массива  $Y$ ;
  - в массиве  $Z_1$  в качестве текущего значения выбирают следующую строку и записывают это число в первые  $m$  разрядов;
  - затем переходят к следующему массиву  $Z_i$  и проверяют наличие сигнала перехода к новому текущему значению из предыдущего массива  $Z_{i-1}$ ;
  - сигнал перехода из массива  $Z_{i-1}$  формируют в том случае, если в предшествующей строке массива  $Z_{i-1}$  в  $(m+1)$ -ом разряде был “нуль”, а в текущей строке массива  $Z_{i-1}$  в  $(m+1)$ -ом разряде “единица”;
  - если сигнал перехода есть, то в массиве  $Z_i$  в качестве текущего значения выбирают следующую строку и записывают это число в следующие  $m$  разрядов формируемой строки массива  $Y$ ;
- повторяют указанные действия  $(k-1)$  раз до формирования полной строки;
- повторяют все действия по формированию следующей строки массива  $Y$   $(P^n - 1)$  раз до полного формирования всех строк массива  $Y$ , причем если в любом массиве  $Z_i$  текущим значением выбирают последнюю строку, то в качестве следующего текущего значения массива  $Z_i$  выбирают его первую строку;
- осуществляют представление всего полученного массива  $Y$ , содержащего  $P^n$   $n$ -разрядных  $P$ -ичных чисел.

Предлагаемый линейный метод единично-инкрементного суммирования чисел может быть реализован в виде некоторого устройства суммирования. Рассмотрим такое устройство более подробно.

### 7.3. УСТРОЙСТВО ЛИНЕЙНОГО ЕДИНИЧНО-ИНКРЕМЕНТНОГО СУММИРОВАНИЯ ЧИСЕЛ

На рисунке 7.3.1 показана схема (вариант) устройства для осуществления линейного метода единично-инкрементного суммирования чисел. Устройство линейного единично-инкрементного суммирования чисел содержит следующие блоки:

- блок 1 представления исходного числа  $X$ ,
- $k$  блоков 2, 3, 4, 5, 6 выделения  $m$ -разрядных чисел,
- блок 7 сумматоров,
- $k$  блоков 8, 10, 12, 14, 16 фиксации промежуточных массивов  $Z_1, \dots, Z_i, \dots, Z_k$ ,
- $(k-1)$  блок 9, 11, 13, 15 фиксации единиц переноса,
- $k$  блоков 17, 19, 21, 23, 25 формирования строки массива  $Y$ ,
- $(k-1)$  блок 18, 20, 22, 24 анализа единиц переноса,
- блок 26 формирования текущей строки массива  $Y$  и
- блок 27 представления всего массива  $Y$ .

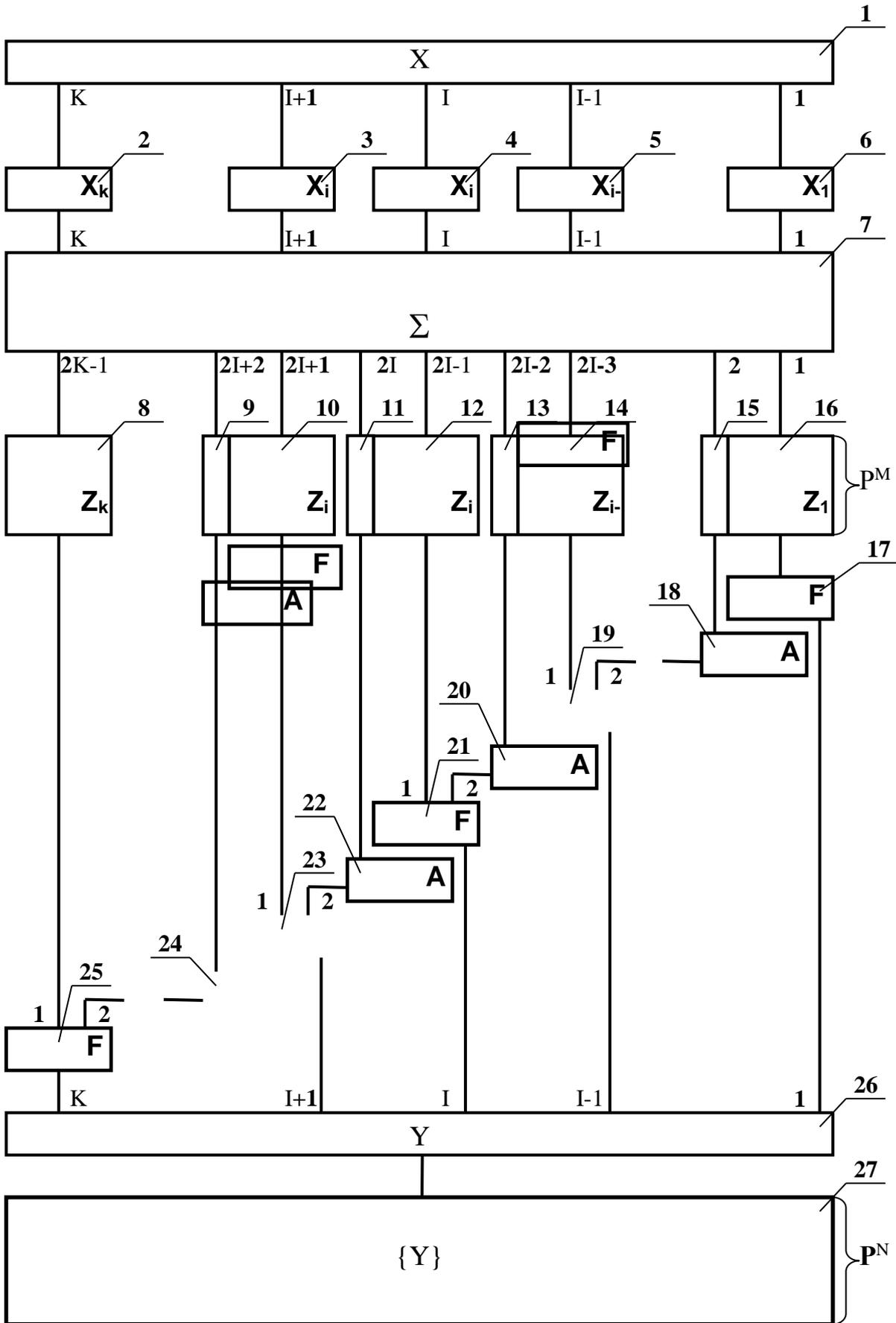


Рис. 7.3.1. Схема линейного переборного суммирования чисел.

Каждый из  $k$  блоков 8, 10, 12, 14, 16 фиксации промежуточных массивов  $Z_1, \dots, Z_i, \dots, Z_k$  и из  $(k-1)$ -го блока 9, 11, 13, 15 фиксации единиц переноса содержит  $P^m$  чисел, а блок 27 представления всего массива  $Y$  содержит  $P^n$   $n$ -разрядных  $P$ -ичных чисел.

Рассмотрим взаимодействие блоков указанного устройства реализации линейного метода единично-инкрементного суммирования чисел. Блок 1 представления исходного числа  $X$  имеет  $k$  выходов, которые соответственно своему номеру соединены со входами 2, 3, 4, 5, 6 блоков выделения  $m$ -разрядных чисел. Каждый выход блоков 2, 3, 4, 5, 6 выделения  $m$ -разрядных чисел соединен соответственно своему номеру со входами блока 7 сумматоров.

Нечетные выходы блока 7 сумматоров, начиная с 1 и до  $2k-1$ , соединены с соответствующими входами  $k$  блоков 8, 10, 12, 14, 16 фиксации промежуточных массивов  $Z_1, \dots, Z_i, \dots, Z_k$ . Четные выходы блока 7 сумматоров, начиная с 2 и до  $2k-2$ , соединены с соответствующими входами  $(k-1)$ -го блока 9, 11, 13, 15 фиксации единиц переноса. Выходы блоков 8, 10, 12, 14, 16 фиксации промежуточных массивов  $Z_1, \dots, Z_i, \dots, Z_k$  соединены с первыми входами соответствующих  $k$  блоков 17, 19, 21, 23, 25 формирования строки массива  $Y$ . Выходы блоков 9, 11, 13, 15 фиксации единиц переноса соединены соответственно со входами  $(k-1)$ -го блока 18, 20, 22, 24 анализа единиц переноса. Выходы блоков 18, 20, 22, 24 анализа единиц переноса соединены соответственно со вторыми входами следующих за ними блоков 19, 21, 23, 25 формирования строки массива  $Y$ , причем выход блока 20 анализа единиц переноса  $ANAL_{i-1}$  соединен со вторым входом блока 21 формирования строки массива  $Y$   $FORM_i$ . Выходы  $k$  блоков 17, 19, 21, 23, 25 формирования строки массива  $Y$  соединены соответственно своим номерам со входами блока 26 формирования текущей строки массива  $Y$ . Выход блока 26 формирования текущей строки массива  $Y$  соединен со входом блока 27 представления всего массива  $Y$ .

Перейдем к описанию алгоритма единично-инкрементного суммирования чисел.

#### 7.4. АЛГОРИТМ ЛИНЕЙНОГО ЕДИНИЧНО-ИНКРЕМЕНТНОГО СУММИРОВАНИЯ ЧИСЕЛ

Для реализации линейного метода единично-инкрементного суммирования чисел предлагается следующий алгоритм, который осуществляется в предлагаемом устройстве суммирования следующим образом.

Прежде всего, в блок 1 представления исходного  $n$ -разрядного  $P$ -ичного числа  $X$  записывают исходное число  $X$ . С выходов этого блока 1 представления исходного числа  $X$  сигналы поступают на входы соответствующих  $k$  блоков 2, 3, 4, 5, 6 выделения  $m$ -разрядных чисел, в которых фиксируют числа  $X_1, \dots, X_i, \dots, X_k$ . С выходов всех  $k$  блоков 2, 3, 4, 5, 6 выделения  $m$ -разрядных чисел сигналы одновременно или последовательно поступают на соответствующие входы блока 7 сумматоров.

В блоке 7 сумматоров производят параллельное на нескольких сумматорах или последовательное на одном сумматоре добавление единицы к каждому числу  $X_1, \dots, X_i, \dots, X_k$  с формированием в  $(m+1)$ -ом разряде “единиц” или “нулей” переноса. Повторяют указанные выше в предыдущем пункте действия  $P^m$  раз и получают в результате  $(m+1)$ -разрядные массивы чисел. Сигналы с нечетных выходов блока 7 сумматоров, начиная с 1 и до  $2k-1$ , поступают на входы соответствующих  $k$  блоков 8, 10, 12, 14, 16 фиксации промежуточных массивов  $Z_1, \dots, Z_i, \dots, Z_k$ .

Во всех  $k$  блоках 8,10,12,14,16 фиксации промежуточных массивов  $Z_1, \dots, Z_i, \dots, Z_k$  производят фиксацию промежуточных массивов  $Z_1, \dots, Z_i, \dots, Z_k$ . Сигналы с четных выходов блока 7 сумматоров, начиная с 1 и до  $2k-2$ , поступают на входы соответствующих  $(k-1)$ -го блоков 9, 11, 13, 15 фиксации единиц переноса. Во всех этих блоках 9, 11, 13, 15 производят фиксацию соответственно “единиц” или “нулей” переноса. Сигналы с выходов  $k$  блоков 8, 10, 12, 14, 16 фиксации промежуточных массивов  $Z_1, \dots, Z_i, \dots, Z_k$  поступают на первые входы соответствующих  $k$  блоков 17, 19, 21, 23, 25 формирования строки массива  $Y$ . Сигналы с выходов  $(k-1)$ -го блоков 9, 11, 13, 15 фиксации единиц переноса поступают на входы соответствующих  $(k-1)$ -го блока 18, 20, 22, 24 анализа единиц переноса.

Формируют строки требуемого массива  $Y$  следующим образом. Прежде всего, в качестве текущих значений в каждом массиве  $Z_1, \dots, Z_i, \dots, Z_k$  выбирают первые строки. Сформированные сигналы с выходов  $k$  блоков 8, 10, 12, 14, 16 фиксации промежуточных массивов  $Z_1, \dots, Z_i, \dots, Z_k$  поступают на первые входы соответствующих  $k$  блоков 17, 19, 21, 23, 25 формирования строки массива  $Y$ .

В блоках 17, 19, 21, 23, 25 формирования строки массива  $Y$  фиксируют текущие значения чисел, затем сигналы с выходов  $k$  блоков 17, 19, 21, 23, 25 формирования строки массива  $Y$  поступают на соответствующие входы блока 26 формирования текущей строки массива  $Y$ . В блоке 26 формирования текущей строки массива  $Y$  получают первую строку требуемого массива  $Y$  и сигнал с выхода блока 26 формирования текущей строки массива  $Y$  поступает на вход блока 27 представления всего массива  $Y$ , в котором записывают первую строку требуемого массива  $Y$ . Переходят к формированию следующей строки требуемого массива  $Y$ . В блоке 16 фиксации промежуточного массива  $Z_1$  в массиве  $Z_1$  в качестве текущего значения выбирают следующую строку. Сформированный сигнал поступает из блока 16 фиксации промежуточного массива  $Z_1$  на вход блока 17 формирования строки массива  $Y$   $FORM_1$ . В блоке 17 формирования строки массива  $Y$   $FORM_1$  формируют новое текущее значение числа. Сигнал из блока 17 формирования строки массива  $Y$   $FORM_1$  поступает на соответствующий вход блока 26 формирования текущей строки массива  $Y$ , в котором записывают это текущее число в первые  $m$  разрядов.

Затем переходят к следующему блоку 21 формирования строки массива  $Y$   $FORM_i$ , который соединен с блоком 12 фиксации промежуточного массива  $Z_i$  и проверяют в предшествующем блоке 20 анализа единиц переноса  $ANAL_{i-1}$ , который соединен с блоком 13 фиксации единиц переноса массива  $Z_{i-1}$ , наличие сигнала перехода к новому текущему значению в блоке 21 формирования строки массива  $Y$   $FORM_i$ . При этом сигнал перехода формируют в предыдущем блоке 20 анализа единиц переноса  $ANAL_{i-1}$ , только в том случае, если в предшествующей строке в блоке 13 фиксации единиц переноса массива  $Z_{i-1}$  был “нуль”, а в текущей строке в блоке 13 фиксации единиц переноса массива  $Z_{i-1}$  “единица”. Если сигнал перехода сформирован в предыдущем блоке 20 анализа единиц переноса  $ANAL_{i-1}$ , то в блоке 21 формирования строки массива  $Y$   $FORM_i$  в качестве текущего значения выбирают следующую строку из блока 12 фиксации промежуточного массива  $Z_i$ .

Формируют сигнал из блока 12 фиксации промежуточного массива  $Z_i$ , который поступает на вход блока 21 формирования строки массива  $Y$   $FORM_i$ . Фиксируют текущее значение числа из массива  $Z_i$ , затем формируют сигнал с выхода  $i$ -того блока 21 формирования строки массива  $Y$   $FORM_i$ , который поступает на соответствующий вход блока 26 формирования текущей строки массива  $Y$ , в котором записывают это текущее число в следующие  $m$  разрядов формируемой строки массива  $Y$ .

Повторяют указанные действия (**k-1**) раз до формирования полной строки в блоке 26 формирования текущей строки массива **Y**. В блоке 26 формирования текущей строки массива **Y** получают следующую строку требуемого массива **Y** и посылают сигнал с выхода блока 26 формирования текущей строки массива **Y** на вход блока 27 представления всего массива **Y**, в котором записывают следующую строку требуемого массива **Y**. После этого повторяют все действия по формированию следующей строки массива **Y** ( $P^n-1$ ) раз до полного формирования всех строк массива **Y** в блоке 27 представления всего массива **Y**, в котором записывают все  $P^n$  строк требуемого массива **Y**. Причем, если в любом массиве  $Z_i$  в блоке 12 фиксации промежуточного массива  $Z_i$ , текущим значением выбирают последнюю строку, то в качестве следующего текущего значения массива  $Z_i$  выбирают его первую строку. После рассмотрения алгоритма единично-инкрементного суммирования чисел целесообразно перейти к исследованию параметров этого суммирования.

### 7.5. ИССЛЕДОВАНИЕ ПАРАМЕТРОВ ЕДИНИЧНО-ИНКРЕМЕНТНОГО СУММИРОВАНИЯ ЧИСЕЛ

Проведем оценку временной и емкостной сложности предлагаемого метода на основе определения количества требуемых операций сложения и объема оперативной памяти. Обозначим количество операций сложения - **KS**, а ресурсы требуемой оперативной памяти - **OP**. Для традиционного простого переборного суммирования необходимо выполнить на **n**-разрядном сумматоре  $P^n$  операций сложения. В оперативной памяти необходимо хранить только **n**-разрядное исходное число **X (n)** и получаемое **n**-разрядное число **Y (n)**. Отметим, что требуемый массив **{Y}**, содержащий  $P^n$  чисел, не должен храниться в оперативной памяти. Тогда:

$$KS_0 = P^n; \quad (7.5.1)$$

$$OP_0 = 2n. \quad (7.5.2)$$

Общее количество операций сложения **KS<sub>1</sub>** в линейном методе определяют на этапе суммирования как **k** раз по  $P^m$  (**m+1**)-разрядных операций сложения, где:  $n = km$  и  $m = \text{Const}$ . В оперативной памяти необходимо хранить исходное число **X (km)**, требуемое число **Y (km)**, разбиение числа **X** на **k** чисел по **m** разрядов (**km**), **k** массивов  $Z_i$  по  $P^m$  (**m+1**)-разрядных чисел ( $k(m+1)P^m$ ) и **k** блоков формирования числа по **m** разрядов (**km**). Пусть

$$\frac{P^m}{m} = \text{Const} = a, \quad (7.5.3)$$

тогда получаем следующие зависимости:

$$KS_1 = kP^m = \frac{n}{m} P^m = \frac{P^m}{m} n = an; \quad (7.5.4)$$

$$OP_1 = 4km + k(m+1)P^m \approx km(4+P^m) = (4+P^m)n \approx P^m n. \quad (7.5.5)$$

Таким образом, линейный метод суммирования чисел позволяет уменьшить количество операций сложения при полном переборном суммировании, уменьшить разрядность сумматоров, повысить интегральное быстродействие суммирования чисел, а также перейти от степенной к линейной зависимости общего количества операций сложения от разрядности чисел. Размер требуемой оперативной памяти линейно зависит от разрядности чисел (**n**), но этот линейный коэффициент имеет степенную зависимость от выбираемого фиксированного числа **m**.

Кроме того, суммирование  $k$  раз по  $P^m$  можно провести заранее, что позволит использовать "табличные" сумматоры и вообще обойтись без традиционных операций сложения. Отметим, что предлагаемый метод суммирования может быть реализован на различной вычислительной технике, обладающей различными характеристиками. В некоторых случаях непосредственная реализация линейного метода единично-инкрементного суммирования чисел не представляется возможной из-за несоответствия аппаратных ресурсов алгоритмическим требованиям. Однако, для снятия этого противоречия представляется целесообразным предложить дополнительно две модификации линейного метода единично-инкрементного суммирования чисел. Рассмотрим "табличную" модификацию единично-инкрементного суммирования чисел на основе применения "табличного" сумматора.

## 7.6. "ТАБЛИЧНАЯ" МОДИФИКАЦИЯ ЕДИНИЧНО-ИНКРЕМЕНТНОГО СУММИРОВАНИЯ ЧИСЕЛ

Для разработки "табличной" модификации линейного метода единично-инкрементного суммирования чисел прежде всего, необходимо внимательно проанализировать таблицу 7.2.1. Анализ этой таблицы показывает, что можно использовать только одну заранее сформированную таблицу  $Z_0$ , содержащую  $P^m$   $m$ -разрядных чисел.

Величину константы  $m$  целесообразно выбирать так, чтобы вся эта таблица  $Z_0$  размещалась в оперативной памяти.

Реализовать имитацию хранения  $k$  блоков 8, 10, 12, 14, 16 фиксации промежуточных массивов  $Z_1, \dots, Z_i, \dots, Z_k$  можно путем применения **специальных ссылок** на единственную таблицу  $Z_0$ . Вместо каждого массива  $Z_1, \dots, Z_i, \dots, Z_k$  создается соответствующая специальная ссылка  $S_1, \dots, S_i, \dots, S_k$ , указывающая на номер определенной "текущей" строки в массиве  $Z_0$  (рис. 7.6.1.).

При получении сигнала переноса в соответствующей специальной ссылке  $S_i$  осуществляется корректный переход к следующей строке массива  $Z_0$ . Таким образом, единично-инкрементное переборное суммирование можно выполнять без применения специальных устройств-сумматоров.

Определим ресурсы, необходимые для реализации данной модификации линейного метода единично-инкрементного суммирования чисел.

Общее количество операций сложения  $KS_2$  определяют на этапе предварительного суммирования как  $P^m$   $m$ -разрядных операций сложения.

В оперативной памяти  $OP_2$  необходимо хранить требуемое число  $Y$  ( $km$ ), массив  $Z_0$  по  $P^m$   $m$ -разрядных чисел ( $mP^m$ ) и  $k$  специальных ссылок  $S_1, \dots, S_i, \dots, S_k$  по  $m$  разрядов ( $km$ ).

Тогда:

$$KS_2 = P^m = \text{Const}; \quad (7.6.1)$$

$$OP_2 = 2km + mP^m = m(2k + P^m) = 2n + \text{Const}. \quad (7.6.2)$$

Рассмотрим еще одну модификацию метода единично-инкрементного суммирования чисел с уменьшением необходимого объема оперативной памяти ЭВМ (рис. 7.7.1.).

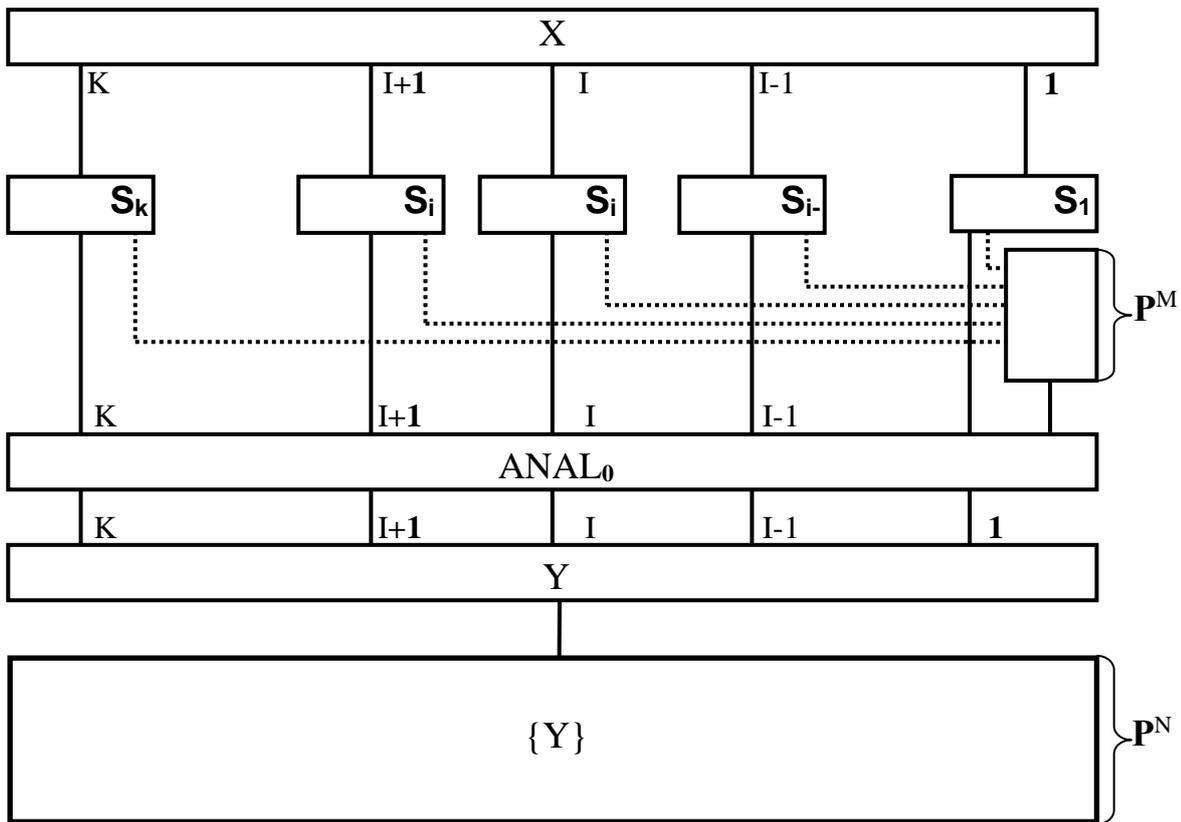


Рис. 7.6.1. Схема "табличного" линейного суммирования чисел.

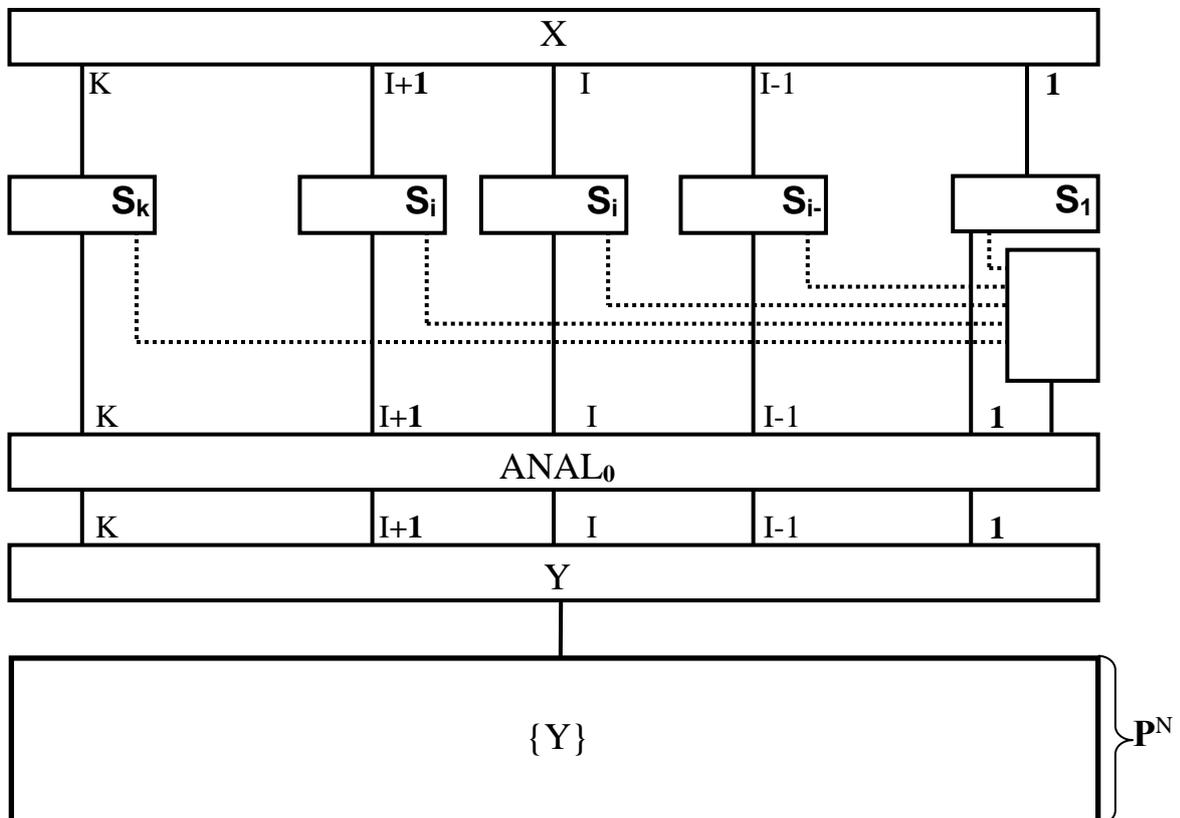


Рис. 7.7.1. Схема суммирования чисел с ограниченной памятью.

### 7.7. МОДИФИКАЦИЯ МЕТОДА СУММИРОВАНИЯ ЧИСЕЛ С УМЕНЬШЕНИЕМ НЕОБХОДИМОГО ОБЪЕМА ОПЕРАТИВНОЙ ПАМЯТИ ЭВМ

Если объем оперативной памяти ЭВМ (ВВК) не позволяет эффективно реализовать линейный метод суммирования, то целесообразно реализовать переборное единично-инкрементное суммирование следующим образом. Число  $X$  также разбивается на равные группы, которые записываются в соответствующие специальные ссылки. При этом целесообразно разбивать многоразрядное число на такие группы, чтобы число разрядов в этих группах совпадало с разрядностью используемых сумматоров. При реализации традиционного суммирования также производят разбиение исходного числа на  $k$  групп, размерность ( $m$ ) которых соответствует разрядности сумматоров, а затем осуществляют суммирование каждой группы  $P^n$  раз. Тогда общее количество операций  $KS_{0R}$  суммирования определяется, как сумма  $k$  раз по  $P^n$  операций сложения:

$$KS_{0R} = kP^n. \quad (7.7.1)$$

В предлагаемой модификации метода полное переборное суммирование происходит только с самой первой ссылкой  $S_1$ , к которой в цикле добавляют единицу (рис. 7.7.1). После каждого суммирования анализируют полученное число. Если это число не равно нулю, то все текущие значения из всех специальных ссылок пересылаются в соответствующие разряды формируемого числа  $Y$  и переходят к формированию следующего числа.

Если анализируемое число равно нулю, то формируется сигнал суммирования (переноса) для следующей специальной ссылки и осуществляется добавление единицы к этой ссылке. Полученное число также анализируется и осуществляется либо формирование очередного числа  $Y$ , либо переход к суммированию в следующей специальной ссылке. Для полного переборного суммирования, таким образом, требуется провести  $P^m$  операций сложения для самой старшей ссылки  $S_k$ , для каждой следующей ссылки количество операций сложения увеличивается и для самой младшей ссылки  $S_1$  требуется уже  $P^n$  операций сложения. Тогда общее количество операций сложения  $KS_3$  определяется суммой:

$$KS_3 = P^m + \dots + P^{im} + \dots + P^n = \sum_{i=1}^k P^{im} \approx \frac{k}{2} P^{km} < kP^n = KS_{0R}. \quad (7.7.2)$$

При таком подходе зависимость количества операций сложения от разрядности остается степенной, но использование разбиения и алгоритмической минимизации количества необходимых операций сложения все же позволяет, примерно в два раза, сократить общее количество операций суммирования с  $KS_{0R}$  до  $KS_3$ .

Кроме того, при этом в оперативной памяти  $OP_3$  необходимо хранить только требуемое число  $Y$  ( $km$ ) и текущее число  $X$  в виде  $k$  специальных ссылок  $S_1, \dots, S_i, \dots, S_k$  по  $m$  разрядов ( $km$ ). Тогда:

$$OP_3 = 2km = 2n. \quad (7.7.3)$$

Использование линейного метода на основе разбиения числа на группы, специальных ссылок  $S_1, \dots, S_i, \dots, S_k$  и "табличного" сумматора позволит осуществлять не только полное переборное единично-инкрементное суммирование чисел при  $P$ -ичном суммировании  $n$ -разрядного числа  $X$  с массивом  $V$   $n$ -разрядных чисел от  $0$  до  $P^n$ , но и переборное суммирование на некотором фиксированном интервале чисел от  $X_A$  до  $X_B$ , где:

$$0 \leq X_A \leq X_B \leq P^n. \quad (7.7.4)$$

В этом случае исходное значение будет не 0, а некоторое число  $X_A$ , которое по определенным выделенным группам фиксируется в массиве  $Z_0$ , путем указания соответствующих значений специальных ссылок  $S_1, \dots, S_i, \dots, S_k$ . После задания исходного состояния  $X_A$  выполняются все выше перечисленные соответствующие действия по суммированию этого числа вплоть до достижения предельного значения  $X_B$ . При этом, метод остается неизменным, а изменяются только граничные значения его применения по единично-инкрементному суммированию чисел. Определенные значения всех переменных целесообразно выбирать индивидуально, исходя из условий конкретной задачи, параметров и возможностей доступных вычислительных средств. Например, значение основной константы  $m$  должно соответствовать разрядности используемых сумматоров и учитывать размеры доступной оперативной памяти. Таким образом, разработанный линейный метод единично-инкрементного (полного переборного) суммирования чисел позволяет:

- уменьшить количество операций сложения при переборном суммировании,
- уменьшить разрядность сумматоров,
- повысить интегральное быстродействие суммирования чисел, а также
- перейти от степенной к линейной зависимости общего количества операций сложения от разрядности чисел.

Кроме того, предложены две модификации линейного метода суммирования:

- 1) первая модификация позволяет реализовать суммирование на основе специальной таблицы без использования сумматоров;
- 2) вторая модификация позволяет ускорить суммирование и реализовать предложенный метод на вычислительных системах с недостаточной оперативной памятью.

Итак, приходим к выводу, что линейный метод единично-инкрементного суммирования чисел целесообразно использовать при разработке специального программного обеспечения различных высокопроизводительных вычислительных комплексов АССОИ для повышения оперативности обработки специальной информации.

Кроме вышеуказанных методов, повысить оперативность обработки данных в АССОИ представляется возможным, при определенных условиях, за счет локальных корректировок вычислений и обработки данных, которые мы и рассмотрим в этом разделе.

## **7.8. ЛОКАЛЬНЫЕ КОРРЕКТИРОВКИ ВЫЧИСЛЕНИЙ И ОБРАБОТКИ ДАННЫХ**

В автоматизированных системах сбора и обработки информации - АССОИ, - для решения конкретных различных задач [Л. 326, 443-452, 460-503] осуществляется поиск и формирование маршрута обработки данных. В зависимости от значений входных данных выполняются все вычисления в соответствии с полученным маршрутом обработки данных. Как правило, количество входных переменных больше 1.

Существуют такие ситуации, когда большинство значений входных переменных остается неизменным, а изменяются только отдельные входные переменные. В настоящее время, в таком случае повторной обработки осуществляется полный пересчет ("переобработка") всех входных переменных. Однако, существуют такие задачи, которые требуют значительного времени и ресурсов для обра-

ботки. Исследуем возможность сокращения времени и ресурсов повторной обработки данных [Л. 326]. Такое сокращение, возможно, прежде всего, за счет использования ранее вычисленных значений на различных этапах обработки данных. Для этого необходимо сохранять промежуточные значения и результаты обработки на каждом этапе маршрута обработки данных, т.е. выполнять "трассировку" маршрута обработки данных.

Если изменяется только часть значений входных параметров при повторной обработке данных, то можно говорить о локальной корректировке обработки данных. Более того, в общем случае, если при повторной обработке данных добавляются, изменяются или удаляются некоторые входные переменные, а также в случае изменения маршрута обработки, целесообразно ввести термин: **адаптивная обработка данных**. Подчеркнем, что в данном случае этот термин относится только к вторичной, повторной, многократной обработке данных.

Итак, постановка задачи состоит в следующем. По определенному маршруту произведена обработка данных, получены требуемые результаты и зафиксированы промежуточные результаты обработки (вычислений). После изменения значения нескольких входных переменных необходимо произвести локальную корректировку обработки и, с учетом полученных ранее результатов, получить новые требуемые результаты обработки.

Для накопления и хранения полученных промежуточных и искомых (выходных) результатов целесообразно создать базу данных (БД) маршрутов обработки и полученных значений переменных или, другими словами, базу данных трассировки (БДТ).

В общем случае при обработке данных могут изменяться: структура обработки, т.е. маршрут, и значения входных данных. Отметим, что изменения маршрута обработки также могут быть рассмотрены в рамках введенного понятия адаптивной обработки информации, т.е. локальных корректировок обработки. При этом, степень "локальности" внесенных изменений может быть различной. Тогда получаем следующие варианты локальных корректировок:

1. Маршрут не изменяется, входные данные не изменяются.
2. Маршрут не изменяется, входные данные изменяются.
3. Маршрут изменяется, входные данные не изменяются.
4. Маршрут изменяется, входные данные изменяются.

Первый вариант является начальным, "вырожденным". Во всех остальных случаях необходимо провести дополнительную обработку и получить новые искомые значения.

Прежде всего, необходимо на основе перечней (списков, множеств) переменных (объектов) и правил (процедур) построить двудольный ориентированный граф обработки. В этом графе каждая входная переменная "порождает" некоторый подграф обработки.

Объединением всех таких подграфов является первоначальный, исходный граф обработки.

Если маршрут обработки не изменяется (вариант 2), то структура графа обработки и всех его подграфов остается постоянной. При изменении значения только одной входной переменной, выделяется подграф, порождаемый этой переменной. В этом выделенном подграфе производится обработка данных с использованием, полученных ранее значений из базы данных маршрутов.

Если маршрут обработки изменяется, а входные данные не изменяются (вариант 3), то и выходные данные не должны измениться. Отметим, что такой вариант возможен при введении новых правил, ускоряющих процесс вывода. Впро-

чем, если маршрут вывода изменен по другим причинам, то необходимо найти "точку" начала изменения маршрута и, используя базу данных трассировки, с этого момента произвести пересчет полученных результатов.

Если маршрут обработки изменяется и входные данные изменяются (вариант 4), то необходимо, аналогично вышеуказанным вариантам, определить, где произведены изменения и с этого места провести новый расчет требуемых значений.

Все предлагаемые способы адаптивных вычислений и локальных корректировок вычислений потребляют дополнительные ресурсы, поэтому, априорно, целесообразно определить границы и возможности применения этих способов, которые зависят от специфики предметной области, от количества произведенных изменений и т.п.

В целом для определенного класса задач и при определенных изменениях маршрутов и данных, предлагаемые методы могут позволить значительно сократить ресурсы повторной обработки потоков данных.

В дальнейшем, целесообразно продолжить исследования возможностей сокращения времени и ресурсов повторной обработки данных на основе предложенных здесь адаптивных вычислений.

В связи с ограниченным объемом книги, представляется целесообразным на данном этапе ограничиться уже приведенным материалом и перейти к исследованию взаимосвязи, проанализированных выше, эволюционных неоднородных компьютерных систем и синтезом интеллектуальных систем.

Итак, перейдем к последнему разделу монографии, где проведем анализ проблемы синтеза интеллектуальных систем с использованием эволюционных неоднородных компьютерных систем и других, полученных нами выше новых научных результатов.

## 8. ЭВОЛЮЦИОННЫЕ НЕОДНОРОДНЫЕ КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СИНТЕЗ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

В завершающем разделе данной работы, после решения проблемы создания эволюционных баз данных и правил, нам осталось рассмотреть только проблему синтеза интеллектуальных систем. Эта проблема сформулирована и подробно описана в первом разделе данной книги.

Отметим, что начать исследование проблемы синтеза интеллектуальных систем представляется целесообразным с обзора существующих научных взглядов, проблем, достижений и перспектив в области построения многопроцессорных вычислительных систем. Ведь, как мы уже определили выше, интеллектуальная система, как правило, должна быть многопроцессорной. Кроме того, любая однопроцессорная ЭВМ, в общем случае, может рассматриваться как частный случай многопроцессорных систем.

Следовательно, исследование принципов построения многопроцессорных вычислительных систем позволит нам перейти к исследованию синтеза интеллектуальных систем.

### 8.1. ПРИНЦИПЫ ПОСТРОЕНИЯ ОДНОРОДНЫХ МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

В настоящее время, практически все наиболее производительные многопроцессорные вычислительные системы - МВС - являются однородными. Следовательно, прежде всего, целесообразно рассмотреть основные архитектуры и принципы построения традиционных однородных многопроцессорных вычислительных систем (параллельных ЭВМ) или однородных высокопроизводительных вычислительных комплексов - ВВК (т.е. информационно-вычислительных конфигураций).

Понятие "**архитектура параллельной ЭВМ**" включает совокупность свойств, определяющих состав и связи оборудования (структуру ЭВМ), типы используемых параллельных вычислительных алгоритмов (алгоритмику) и средства программирования (языки, трансляторы, операционные системы). Евреинов Э.В. в [Л. 94-96] отмечает, что в основе однородных вычислительных систем, структур и сред (ОВС) лежит модель коллектива вычислителей, которая является обобщением общепринятой модели вычислителя. Прежде всего, в этой модели коллектива вычислителей все элементы и связи однородны.

Многопроцессорные вычислительные системы и ВВК, как и все параллельные компьютеры состоят из трех основных компонент:

- 1) процессоры,
- 2) модули памяти, и
- 3) коммутирующая сеть.

Именно эти компоненты и их взаимосвязи лучше всего отличают один параллельный компьютер от другого [Л. 86, 94-96, 115-116, 146, 152, 197-198, 204-206, 212, 224, 236, 264, 279, 285-286, 310-311, 318, 328, 331, 333, 384-386].

Основным **принципом построения** однородных ВВК является то, что коммутирующая сеть соединяет однородные процессоры друг с другом и иногда, также с модулями памяти. Процессоры, используемые в параллельных компьютерах, как правило, точно такие же, что и процессоры однопроцессорных (иногда - двухпроцессорных) систем, хотя современная технология позволяет разместить на микросхеме более одного процессора (до четырех).

Кроме того, на микросхеме вместе с процессором могут быть расположены те компоненты (модули) или их составляющие, которые дают наибольший эффект при обработке соответствующего класса задач и параллельных вычислениях.

Архитектуры параллельных компьютеров могут значительно отличаться друг от друга. Одним из свойств, различающих параллельные компьютеры, является **число возможных потоков команд**.

По этому признаку различают следующие архитектуры:

- **MIMD** (*Multiple Instruction Multiple Data* - множество потоков команд и множество потоков данных). MIMD компьютер имеет  $N$  процессоров,  $N$  потоков команд и  $N$  потоков данных. Каждый процессор функционирует под управлением собственного потока команд.
- **SIMD** (*Single Instruction Multiple Data* - единственный поток команд и множество потоков данных). SIMD компьютер имеет  $N$  идентичных синхронно работающих процессоров,  $N$  потоков данных и один поток команд. Каждый процессор обладает собственной локальной памятью. Сеть, соединяющая процессоры, обычно имеет регулярную топологию.

Другим **принципом построения** однородных многопроцессорных вычислительных систем (ВВК) является **способ доступа к модулям памяти**, то есть, имеет ли каждый процессор локальную память и обращается к другим блокам памяти, используя коммутирующую сеть, или коммутирующая сеть соединяет все процессоры с общей памятью.

Исходя из способа доступа к памяти, различают следующие (довольно условные) типы параллельных (MIMD) архитектур.

**Компьютеры с распределенной памятью** (*Distributed memory*). Каждый процессор имеет доступ только к локальной собственной памяти. Процессоры объединены в сеть. Доступ к удаленной памяти возможен только с помощью системы обмена сообщениями.

**Компьютеры с общей (разделяемой) памятью** (*True shared memory*). Каждый процессор компьютера обладает возможностью прямого доступа к общей памяти, используя общую шину (возможно, реализованную на основе высокоскоростной сети). В таких компьютерах нельзя существенно увеличить число процессоров, поскольку при этом происходит резкое увеличение числа конфликтов доступа к шине. В некоторых архитектурах каждый процессор имеет и прямой доступ к общей памяти, и собственную локальную память.

**Компьютеры с виртуальной общей (разделяемой) памятью** (*Virtual shared memory*) В таких системах общая память как таковая отсутствует. Каждый процессор имеет собственную локальную память. Он может обращаться к локальной памяти других процессоров, используя "глобальный адрес". Если "глобальный адрес" указывает не на локальную память, то доступ к памяти реализуется с помощью сообщений с малой задержкой, пересылаемых по сети, соединяющей процессоры.

Отметим два класса компьютерных систем, которые иногда используются как параллельные компьютеры:

- **локальные вычислительные сети (LAN)**, в которых компьютеры находятся физически близко и соединены "быстрой" сетью,
- **глобальные вычислительные сети (WAN)**, которые соединяют географически распределенные компьютеры.

Хотя распределенные вычислительные системы, т.е. вычислительные сети, вводят дополнительные свойства, такие как *надежность* и *защита*, во многих случаях они могут рассматриваться как MIMD компьютеры, хотя и с высокой стоимостью удаленного доступа.

Известно, что в векторных суперЭВМ обеспечена предельная производительность для процессов скалярной и векторной обработки, которая присутствует в большинстве задач. Задачи, содержащие высокую степень внутреннего параллелизма, могут быть хорошо адаптированы к системам массового параллелизма. Реальные задачи и, тем более, пакеты задач содержат целый ряд алгоритмов, имеющих различные уровни параллелизма. Все это говорит о том, что вместо попыток приспособить все типы алгоритмов к одной архитектуре, что отражается на конфигурации архитектур и сопровождается не всегда корректными сравнениями пиковой производительности, более продуктивным является **взаимное дополнение архитектур в единой системе**, с переходом к построению **неоднородных многопроцессорных вычислительных систем (ВВК)**. Одним из первых примеров такой системы является объединение векторной системы Cray Y-XM с системой Cray T3D [Л. 115-116, 369]. Однако это объединение с помощью высокоскоростного канала приводит к необходимости разбиения задач на крупные блоки и к потерям времени и памяти на обмен информацией.

В то же время, дифференциация функций и специализация отдельных подсистем начала развиваться с появления отдельных подсистем и процессоров для обслуживания ввода/вывода, коммуникационных сетей, внешней памяти. Для реализации этого, в суперЭВМ кроме основного процессора включались внешние машины.

В различных компьютерных системах можно наблюдать элементы специализации в направлениях автономного выполнения функций операционной системы, системы программирования и подготовки заданий.

Во-первых, эти вспомогательные функции могут выполняться параллельно с основными вычислениями.

Во-вторых, для их реализации не требуются многие из тех средств, которые обеспечивают высокую производительность основного процессора, например, возможность выполнения операций с плавающей запятой и векторных операций.

В дальнейшем, при интеграции скалярной, векторной и параллельной обработки в рамках единой вычислительной подсистемы состав этих вспомогательных функций должен быть дополнен функциями анализа программ с целью обеспечения требуемого уровня параллелизма и распределения отдельных частей программы по различным ветвям вычислительной подсистемы.

Таким образом, в настоящее время многопроцессорные вычислительные системы и ВВК построенные как симметричные мультипроцессорные системы с общей памятью являются наиболее развитыми с точки зрения накопленного опыта их использования [Л. 35, 73, 115-116, 148-149, 151-152, 154, 156]. Обычно они включают не более 32-64 скалярных процессоров. Для них характерны модульность и масштабируемость. Пример таких систем - отечественные многопроцессорные вычислительные комплексы "Эльбрус-1" и "Эльбрус-2".

Широко распространены и векторные мультипроцессорные системы. Они являются основным, причем, высокоэффективным инструментом при решении многих задач, для которых накоплен большой объем программ. Число процессоров в таких суперсистемах также не превышает 64. Характерные представители данного семейства – компьютеры серий Cray X-MP, Cray C90, Cray T90; системы SX-1, SX-2, SX-3 фирмы NEC; двухпроцессорный вариант системы "Электроника СС БИС-1" [Л. 115-116, 369].

Системы с массовым параллелизмом обеспечивают наивысшую пиковую производительность. Обычно они содержат от десятков до нескольких тысяч высокопроизводительных микропроцессоров, связанных посредством коммутатора с высокой пропускной способностью. Однако, при масштабировании таких систем трудно

достичь соответствующего роста производительности. До сих пор в полной мере не решены проблемы распараллеливания вычислений, за исключением задач, разделяемых на большое число локальных процессов. Было разработано немало систем с массовым параллелизмом, различающихся средствами коммутации, методами доступа к памяти и обмена между процессорами, например: SP2 (IBM), Intel Paragon, nCube, Cray T3E, отечественная система МВС-1000 [Л. 115, 116, 148, 412].

Масштабируемые векторные системы являются развитием серии векторных мультипроцессорных систем и представляют собой системы с массовым параллелизмом на специализированных микропроцессорах. Они представлены суперкомпьютерами Cray SV1 и Cray SV2 (проект) фирмы SGI, а также SX-4 и SX-5 компании NEC.

После описания принципов построения однородных многопроцессорных вычислительных систем и ВВК представляется целесообразным перейти к анализу возможных путей создания эволюционных неоднородных компьютерных систем.

## **8.2. АНАЛИЗ ПУТЕЙ СОЗДАНИЯ ЭВОЛЮЦИОННЫХ НЕОДНОРОДНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ**

Отметим, что кроме изучения традиционных вопросов создания однородных многопроцессорных вычислительных систем и высокопроизводительных вычислительных комплексов (Бабаян Б.А., Евреинов Э.В., Корнеев В.В. и др. [Л. 21, 73-75, 94-96, 112-116, 146, 148-149, 197, 307, 404, 412]), в настоящее время во всем мире проводятся активные исследования по изысканию возможных путей создания эволюционных и неоднородных компьютерных систем (Амамия М., Танака Ю., Бурцев В.С., Вальковский В.С., Валях Е., Воеводин В.В., Каляев А.В. и др. [Л. 11, 16, 34, 38-40, 54-57, 89-90, 112-116, 126-131, 147, 165-167, 384-386]). Так как все однородные системы могут рассматриваться как частный случай неоднородной системы, то данные исследования обобщают весь накопленный опыт и открывают новые перспективы развития многопроцессорных вычислительных систем и ЭВМ в целом.

Неоднородные многопроцессорные вычислительные системы предназначены для оптимизации выполнения больших задач, включающих программы с разными формами параллелизма. Известно, что из-за функциональной специализации неоднородны многие мультипроцессорные системы. Так, в отечественной системе АС-6 имеются периферийные машины и телеметрический процессор, а во всех векторных суперЭВМ есть внешние машины для управления периферийным и коммутационным оборудованием [Л. 115].

Как было показано выше, актуальность создания систем адаптивного синтеза информационно-вычислительных конфигураций - САС ИВК - и эволюционных неоднородных компьютерных систем обусловлена необходимостью постоянной модернизации ЭВМ в условиях непрерывности их функционирования. Создание эволюционных компьютерных систем позволит изменить процесс проектирования и создания ВВК (многопроцессорных вычислительных систем), а также обеспечит возможность проведения поэтапной модернизации модулей ВВК в условиях эксплуатации. Из этого следует, что **эволюционные компьютерные системы** в общем случае должны быть **неоднородными**.

Существует противоречие между универсальностью ЭВМ и требованиями эффективности, производительности при решении конкретных специальных задач. Известно, что для различных типов задач наиболее эффективными являются разные вычислительные устройства, а все универсальные ЭВМ решают такие за-

дачи с различной производительностью, как правило, не превышающей 10-30 % от пиковой. Неоднородные компьютерные системы позволят повысить универсальность ЭВМ за счет подключения к основной системе специализированных устройств, предназначенных для эффективного решения отдельных задач.

Известные научно-технические результаты не позволяют полностью решить данную проблему, что определяет важность и необходимость проведения дополнительных исследований. Подчеркнем, что решение проблемы создания эволюционных неоднородных компьютерных систем позволит повысить интеллектуальность, универсальность и эффективность компьютерных систем, увеличить срок их эксплуатации, а также существенно снизить финансовые затраты на автоматизацию деятельности и повысить эффективность решения различных сложных и специальных задач.

В настоящее время применяются различные подходы к классификации вычислительных систем. Исходя из критерия однородности систем и, следуя модифицированному подходу, изложенному в [Л. 95], можно выделить:

- однородные системы;
- частично неоднородные распределенные системы обработки данных
- полностью неоднородные конфигурации.

Полностью неоднородные системы характеризуются тем, что в них объединены устройства, построенные на основе различных архитектур и функционирующие под управлением разных программных средств. Прежде всего, необходимо уточнить значение термина "эволюционные неоднородные компьютерные системы", показать актуальность и возможность их создания.

Основные возможные пути создания эволюционных и неоднородных ВВК (многопроцессорных вычислительных систем) разрабатывались и исследовались в ряде научно-исследовательских работ [Л. 11, 43, 89-90, 112-116, 126-131, 135, 146-152, 165-167, 183, 195, 197-198, 204-206, 212, 224, 236, 245, 264, 279, 285, 286, 294, 307, 309, 313, 333, 343-347, 357, 369, 371, 384-386, 391, 393-394, 404-406, 410, 412]. Эти исследования показали, что наиболее перспективными возможными **путями создания неоднородных эволюционных компьютерных систем**, в том числе и ВВК, являются:

- 1) создание функциональных частично неоднородных ВВК,
- 2) создание ВВК на основе неоднородных основных вычислительных средств,
- 3) создание ВВК с изменяемой (перепрограммируемой) архитектурой.

Неоднородные ВВК предназначены для оптимизации выполнения больших задач, включающих программы с разными формами параллелизма. Вообще говоря, из-за функциональной специализации неоднородны многие мультипроцессорные системы [Л. 48, 115, 146]. Так, в отечественной системе АС-6 имеются периферийные машины и телеметрический процессор, в компьютерах CDC 6600 и CDC 7600 – периферийные процессоры. Во всех векторных суперкомпьютерах есть внешние машины для управления периферийным и коммутационным оборудованием. Система Cray-2 кроме четырех вычислительных векторных процессоров оснащена мониторным процессором, реализующим функции операционной системы [Л. 115].

Однако, в случае существующих неоднородных суперсистем и ВВК речь идет о неоднородности основных вычислительных средств, что не исключает неоднородность и за счет функциональной специализации. Исследования архитектуры таких систем показали, что повышение эффективности возможно за счет сочетания достоинств конвейерной обработки в мультиконвейерном процессоре и параллельной обработки в процессоре с массовым параллелизмом, тесно связанных между собой и с подсистемой автоматического распределения ресурсов. Этот подход хо-

рошо согласуется с требованиями по масштабированию и оптимизации алгоритмов в соответствии с архитектурой. Так, в новом проекте Cray SV2 ставится задача построения системы, которая должна заменить суперкомпьютер с массовым параллелизмом Cray T3E и выпускаемые фирмой SGI векторные системы.

Отметим, что распределенные вычислительные системы основаны на разделении, распределении задач между вычислительными узлами, в которых могут находиться самые различные машины и системы, объединенные в локальные или глобальные сети. Для эффективного решения больших задач в таких системах необходимы новые подходы к методам вычислений.

Таким образом, можно сделать вывод, что наибольшие перспективы имеют неоднородные и распределенные ВВК. При этом, в общем случае, распределенные многопроцессорные вычислительные системы и ВВК также являются неоднородными.

Кроме того, при разработке новых процессоров необходима их специализация и оптимальное сочетание свойств мультиконвейерных, векторных процессоров и процессоров с длинным командным словом. В то же время, процессоры, управляемые потоком данных, наиболее перспективны для применения в подсистемах моделирования и анализа сложных задач с целью выявления параллелизма.

**Построение эволюционных неоднородных компьютерных систем основано** на том, что компьютерная система состоит из подсистем, блоков, модулей, процессоров и памяти, которые взаимодействуют между собой посредством коммуникационной среды. Для обеспечения эволюционности (синергизма, развития) и неоднородности компьютерных систем, необходимо разработать специальные интерфейсы и управляющее программное обеспечение. Интерфейсы должны использовать достижения современных сетевых технологий и обеспечивать универсальность подключения любых модулей процессоров или памяти, а также взаимодействие блоков и подсистем эволюционных неоднородных компьютерных систем. Время взаимодействия различных модулей внутри одного блока должно быть меньше межблокового времени взаимодействия. Аналогичные требования предъявляются и к подсистемам эволюционных неоднородных компьютерных систем [Л. 46] по отношению к компьютерной системе в целом.

Программное обеспечение эволюционных неоднородных компьютерных систем должно учитывать функциональное типовое предназначение каждого блока и модуля, накапливать сведения о загрузке и эффективности функционирования вычислительных блоков и модулей, выдавать запросы и рекомендации по повышению эффективности компьютерной системы при решении различных типовых задач.

Для этого необходимо разработать на основе существующих, новые модели обработки информации для каждого типа задач и модель функционирования компьютерной системы при решении таких задач, в которых для каждого типа задач будут определены наиболее эффективные типы вычислительных устройств. Кроме того, программное обеспечение должно обеспечивать возможность замены отдельных блоков и модулей без прекращения функционирования самой компьютерной системы в целом.

Для решения задачи создания **эволюционных неоднородных компьютерных систем** необходимо использовать предложенные выше концепцию создания системы адаптивного синтеза информационно-вычислительных конфигураций - **САС ИВК**, и миварную концепцию унифицированного представления данных и правил на основе методологии синергетического накопления и оперативной обработки данных.

Для создания системы адаптивного синтеза информационно-вычислительных конфигураций необходимо использовать теоретические основы создания **эволюционных, адаптивных баз данных и знаний**, т.е. новую систему представления и обработки данных, так как существующие традиционные системы не удовлетворяют возросшим требованиям. Кроме того, целесообразно использовать предложенные выше методы обработки данных, которые использует все преимущества нового миварного представления данных.

Как уже было показано, миварная концепция информационного пространства позволит создать более интеллектуальные информационные системы. В то же время, миварный подход целесообразно применять и для создания эволюционных неоднородных компьютерных систем. Таким образом, понятно, что интеллектуальные информационные системы некоторым образом взаимосвязаны с проблемой создания эволюционных неоднородных компьютерных систем. Рассмотрим более подробно взаимосвязь проблемы создания эволюционных неоднородных компьютерных систем с проблемой построения интеллектуальных систем.

### **8.3. ВЗАИМОСВЯЗЬ ИНТЕЛЛЕКТУАЛЬНЫХ И ЭВОЛЮЦИОННЫХ НЕОДНОРОДНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ**

Проблема создания эволюционных неоднородных компьютерных систем тесно связана с проблемой построения интеллектуальных автоматизированных систем обработки информации (АСОИ), а также с научным направлением, изучающим возможности создания систем искусственного интеллекта. Так, для создания действительно интеллектуальной системы необходимо выполнить ряд условий. Например:

1. Для обеспечения живучести и надежности АСОИ такая система должна быть многопроцессорной.
2. Система обработки и накопления информации должна быть универсальной и единой для любых предметных областей, так как интеллектуальная АСОИ (ИАСОИ) должна иметь возможность обучения и работы в любых предметных областях.
3. Для своевременного реагирования на изменения предметной области и окружающей среды, процессы обработки информации должны управляться потоком входных данных.
4. Кроме того, для выживания в агрессивной окружающей среде, ИАСОИ должна активно управлять процессом сбора необходимой исходной информации, формируя, таким образом, систему обработки информации типа "активная логическая сеть, управляемая потоком данных".
5. Принципиально важно, что функционирование ИАСОИ должно быть непрерывным, следовательно, и новые данные, и новые правила должны вводиться и встраиваться в систему обработки без перезагрузки и перепланирования баз данных и знаний.
6. Исходя из необходимости обучения различным предметным задачам в условиях непрерывности функционирования, следует, что система накопления информации должна иметь изменяемую эволюционную (адаптивную) структуру представления данных и правил.
7. Кроме того, такая система накопления информации должна иметь возможность быть глобальной, т.е. обобщать и содержать всю имеющуюся-

ся доступную информацию, теоретически, "в пределе", без ограничений по количеству информации.

Так как любая однородная система является частным случаем неоднородной и, исходя из требований непрерывности функционирования и живучести, следует, что: **интеллектуальная система обработки информации должна быть многопроцессорной эволюционной неоднородной компьютерной системой**, в которой можно наращивать, заменять и модернизировать любую аппаратную или программную подсистему, блок, элемент.

До настоящего времени не предложено комплексного решения указанных проблем и противоречий.

Основой любой современной автоматизированной системы обработки информации является **система обработки данных (СОД)**, которая собственно обрабатывает преобразованную и введенную пользователем информацию в виде данных на ЭВМ. Так как практически любая современная система обработки информации является автоматизированной, то и само слово "автоматизированная" может больше не употребляться. Развитие компьютерных сетевых технологий создало технические предпосылки для создания на основе эволюционных неоднородных компьютерных систем глобальных систем обработки данных. Базы данных, являющиеся основным элементом любой СОД, в настоящее время используют морально устаревшие модели данных, которые не могут обеспечить адекватность моделирования сложных, многоаспектных и взаимосвязанных объектов реального мира. В то же время, системы логического вывода, используемые в системах поддержки принятия решений, плохо взаимодействуют с БД. В результате, вместо единой глобальной СОД, мы имеем в настоящее время множество взаимодействующих, но все же различных СОД.

Под эволюционными СОД будем понимать такие СОД, в которых обеспечена возможность единого системного представления и накопления данных, изменения структур хранения данных, наращивания и/или замены технических и программных средств СОД без прекращения ее функционирования. Эволюционные СОД могут стать основой создания познающих, гносеологических АСОИ и систем искусственного интеллекта.

Отметим, что частные вопросы создания и обеспечения функционирования СОД, которые хорошо проработаны и исследованы [Л. 10-12, 20-22, 25, 34-40, 48, 52, 54-57, 62-66, 70-80, 84, 86-98, 100-105, 107-118, 120-132, 135-152, 154-167, 170-176, 178-180, 183-185, 189-216, 219-222, 224-245, 247-253, 255-258, 261-264, 266, 268-271, 274-280, 284-300, 304-319, 321-331, 338-426], в этой работе рассматриваться не будут.

Разработка методологии развития эволюционных систем обработки данных (ЭСОД) предполагает выделение двух основных классов систем обработки данных:

- **познающих,**
- **диагностических.**

**Познающие системы обработки данных** исследуют новые предметные области, активно накапливают новые данные, что приводит к быстрому наращиванию объема данных и частому изменению структуры их представления.

**Диагностические системы обработки данных** при практически неизменных объемах данных должны за минимальное время выявлять изменения в предметной области и выполнять необходимые при этом действия.

Все реально работающие системы обработки данных сочетают в себе черты двух этих классов в той или иной мере, но к наиболее сложным и наукоемким проблемам относится построение **познающе-диагностической системы (ПДС)**.

Эволюционность, неоднородность и многопроцессорность эволюционных компьютерных систем позволяет совместить эти противоречивые требования (познание и диагностика), например, следующим образом.

В компьютерной системе выделяется некоторая подсистема диагностирования изученных фрагментов и объектов предметной области, в которой на основе миварного подхода, который описан выше, осуществляется представление и обработка данных за минимальное время. На остальной части (в других подсистемах) системы обработки данных осуществляется исследование и уточнение сведений об объектах и предметной области в целом. Подчеркнем, что только миварный подход позволит реализовать это одновременно и в рамках одной системы.

Если некоторый объект достаточно хорошо изучен и необходим для диагностики, то сведения о нем передаются в подсистему диагностики. При диагностике постоянно ведется учет правильности диагностирования. В случае сильных отклонений результатов диагностики осуществляется поиск причины ошибки, в результате чего некоторые объекты могут быть выведены в подсистему познания для дальнейших исследований. В подсистеме познания одновременно может существовать несколько различных представлений одной и той же предметной области, которые сравниваются с результатами системы диагностирования, с целью оценки адекватности моделей реальной предметной области. Отметим, что при разработке различных представлений предметной области представляется возможным использовать подходы традиционной теории биологической эволюции [Л. 29, 32, 52, 58-61, 67, 76-77, 83, 85, 99, 133, 138, 168, 177, 187-188, 212, 217, 221, 229, 239, 256, 260, 281-283, 288, 295, 299, 301, 312, 320, 330, 332, 334, 336], которые кратко были рассмотрены в первом разделе. Для решения задач прогнозирования возможно параллельное моделирование на нескольких представлениях предметной области с последующим оцениванием полученных результатов. Такой подход позволит сочетать достоинства отдельных систем обучения и распознавания в рамках единой системы.

Любой пользователь эволюционной неоднородной компьютерной системы (включая САС ИВК), реализованной на миварной концепции накопления и обработки данных, получает возможность своевременного полного и достоверного получения необходимых сведений в наиболее удобной форме, что сделает реальным, доступным и наглядным понятие "**единого информационного пространства**" любой предметной области.

При этом необходимо учесть возможность применения разрабатываемых под руководством академика РАН Каляева А.В. **многопроцессорных систем с программируемой архитектурой** и структурно-процедурной организацией вычислений [Л. 44, 46, 112-115, 118, 126-131, 165-167, 384-386]. Кроме того, в США также проводятся исследования по данной тематике. Например, журнал "Компьютерра" в номере 356 от 15.08.2000г. на с.9 сообщает, что молодая американская компания QuickSilver Technology надеется произвести революцию на рынке микроэлектроники, выпустив микропроцессор, способный быстро изменять собственную структуру [Л. 144]. Авторы называют разработку адаптивным вычислителем и обращают внимание на его отличие: новая однокристалльная система сможет изменять алгоритмы обработки информации, а также реконфигурировать собственные аппаратные ресурсы. Отметим, что такой подход в некоторой степени аналогичен подходу к созданию САС ИВК, но, кроме того, программное обеспечение САС ИВК должно обеспечивать возможность замены отдельных блоков и модулей без прекращения функционирования самой компьютерной системы в целом.

Кроме того, отдельно выделим эволюционные глобальные системы обработки данных (ЭГСОД), которые представляют собой такие СОД, которые явля-

ются эволюционными без теоретических ограничений по структуре, составу и количеству накапливаемых и обрабатываемых данных. Теоретически, единая универсальная ЭГСОД может обеспечить системное представление, накопление и обработку всех известных человечеству данных. В настоящее время, компьютерная сеть Интернет предоставляет теоретические возможности по накоплению любых объемов данных, но не позволяет осуществить единое обобщенное системно-структурное представление данных и тем более их совместную обработку на основе общей системы различных правил.

Научные исследования в направлении создания перспективных автоматизированных систем активно ведутся во всем мире и в России. Так, например, в настоящее время в Институте высокопроизводительных вычислительных систем РАН (ИВВС РАН) проводятся фундаментальные исследования, направленные на создание архитектуры перспективных вычислительных суперсистем [Л. 115].

Создание проектов перспективных систем включает разработку архитектуры, ее программной модели для анализа эффективности решения различных задач, анализ различных подходов к аппаратной реализации, анализ алгоритмов и программ и методов их оптимизации.

Высокая производительность суперсистем может быть достигнута за счет высокой степени параллелизма, а высокая эффективность за счет оптимизации распределения ресурсов. По существу, при создании проекта суперсистемы необходимо комплексно решить как проблему создания параллельной архитектуры, так и проблему распределения ресурсов.

Проект по разработке архитектуры неоднородной вычислительной системы основан на тесном объединении подсистем с различной архитектурой, соответствующей различным моделям вычислений. Для статического и динамического распределения ресурсов при решении задач в системе предусматриваются аппаратно-программные средства, обеспечивающие анализ задачи, выделение фрагментов с той или иной моделью вычислений и распределение их по вычислительным подсистемам.

Проект суперЭВМ с нетрадиционной архитектурой основан на использовании **принципа потока данных**. Этот принцип предполагает, что последовательность вычислений определяется готовностью операндов и наличием свободного функционального устройства для выполнения операции. При этом, можно значительно увеличить реальный параллелизм работы и реализовать распределение ресурсов в машине с дискретностью до одной операции. Однако, для реализации этой нетрадиционной архитектуры необходима ассоциативная память, сохраняющая готовые операнды и формирующая их пары для выдачи на вход функциональных устройств.

Таким образом, создание эволюционных неоднородных компьютерных систем позволит: повысить универсальность и эффективность компьютерных систем, увеличить срок их эксплуатации, а также существенно снизить финансовые затраты на автоматизацию деятельности многих организаций.

Проведенный нами анализ показал, что наиболее рациональным путем построения эволюционных неоднородных компьютерных систем следует считать концепцию создания систем адаптивного синтеза информационно-вычислительных конфигураций - САС ИВК.

В свою очередь, такая система адаптивного синтеза информационно-вычислительных конфигураций представляет собой саморазвивающийся, синергетический программно-аппаратный комплекс на основе адаптивного синтеза своей собственной информационно-вычислительной конфигурации, управляемого единой миварной системой унифицированной обработки данных и правил.

Рассмотрим более подробно основные пути создания систем адаптивного синтеза информационно-вычислительных конфигураций.

#### **8.4. ОСНОВНЫЕ ПУТИ СОЗДАНИЯ СИСТЕМ АДАПТИВНОГО СИНТЕЗА ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ КОНФИГУРАЦИЙ**

Задача синтеза вычислительной структуры, реализующей некоторый заданный алгоритм, представляет собой важное направление в информатике, которое получило название "Отображение проблем вычислительной математики на архитектуру вычислительных систем". Известны исследования Каляева А.В., Каляева И.А., Левина И.И., Станишевского О.Б., Коробкина В.В., Пономарева И.М. [Л. 115-116, 126-131, 165-167, 384-386], в которых на основании изучения матрицы инцидентий ярусно-параллельной формы графа алгоритма предложены процедуры синтеза вычислительных структур, позволяющие проектировать специализированные вычислительные устройства с учетом наложенных ограничений. При этом, как правило, свойства алгоритма удобно исследовать с помощью графов.

В тоже время, как показали Корженевич Ю.В. и Кобайло Ю.С. [Л. 146], вычислительная структура также может быть описана ориентированным графом, в котором вершины отождествлены с функциональными устройствами, а дуги с линиями связи. При этом реализация алгоритма предполагает выполнение в определенном порядке его операций.

Тогда, в качестве вычислительной структуры можно рассмотреть и такую, граф которой совпадает с вычислительным графом алгоритма. Однако, в этом случае каждое функциональное устройство срабатывает всего один раз [Л. 146]. С помощью гомоморфной свертки, т.е. слияния нескольких вершин с соответствующим применением множества дуг, можно формировать вычислительные структуры, состоящие из меньшего числа функциональных устройств и характеризующиеся более полной загруженностью. Например, в [Л. 146] рассмотрены методы отображения параллельной формы алгоритма на вычислительную структуру, реализующую данный алгоритм при наложенных ограничениях.

Таким образом, известно достаточно много систем синтеза вычислительных конфигураций, но до сих пор не рассматривались более подробно вопросы адаптивного синтеза конфигураций компьютерных систем.

Проведенные исследования показали, что разработка многопроцессорных вычислительных систем и ВВК на традиционных принципах и известных технологических решениях является не достаточно эффективной. Новые принципы, архитектурные и технологические решения требуют разработки новых подходов и теоретических основ разработки многопроцессорных вычислительных систем. Адаптивный синтез конфигураций компьютерных систем является одним из наиболее перспективных направлений исследований в данной области.

Как было показано в первом разделе, разработка принципов построения и применения систем адаптивного синтеза (САС) информационно-вычислительных конфигураций (ИВК) для оперативного анализа и решения сложных, специальных и/или уникальных диагностических задач (УДЗ) является сложной наукоемкой проблемой.

Подчеркнем, что научная проблема создания САС ИВК, кратко описанная выше, относится к управлению в технических системах, в частности, к области самооценки, саморазвития и адаптации компьютерных систем. Кроме того, как было указано выше, проблема может быть сформулирована следующим образом: создание адаптивного синергетического (саморазвивающегося) программно-

аппаратного комплекса активной обработки потоков входных данных на основе пространства унифицированного представления данных и правил, управляемых потоком данных, в условиях дефицита времени одновременного анализа внезапно возникающих сложных задач обучения и распознавания.

Для достижения цели создания САС ИВК, подробно сформулированной в первом разделе, могут быть использованы полученные выше следующие научные результаты:

- 1) разработаны принципы построения **эволюционных адаптивных баз данных и правил**, т.е. объектно-структурного динамически многомерного информационного пространства представления данных и правил;
- 2) разработан метод обработки данных на основе применения "активной" логической сети правил, управляемой потоком данных;
- 3) разработан метод быстрого поиска маршрута логического вывода на основе построения многополюсной сети и поиска ее минимального разреза;
- 4) разработан метод максимального распараллеливания потоковой обработки взаимозависимых данных путем построения виртуальных потоковых баз данных;
- 5) разработан метод алгоритмической минимизации необходимого количества вычислительных процедур и устройств сложения для решения отдельных сложных диагностических задач.

Эти результаты взаимосвязаны следующим образом. В результате разработки эволюционных, адаптивных баз данных, т.е. новых принципов построения систем представления данных, были предложены принципы построения нового объектно-структурного динамически многомерного пространства унифицированного представления данных и правил (миварное пространство), которое полностью удовлетворяет требованиям САС ИВК.

Кроме того, миварная концепция представления данных может быть использована для создания теоретических основ адаптивного синтеза конфигураций многопроцессорных вычислительных систем.

Так как новый миварный подход к представлению и обработке данных существенно расширяет возможности по универсальности и оперативности обработки информации в САС ИВК, то был разработан новый метод оперативной обработки данных на основе применения миварного пространства и построения адаптивной активной логической сети, управляемой потоком данных.

Для повышения быстродействия и оперативности обработки данных в САС ИВК путем использования новых возможностей миварного пространства представления данных и правил был разработан новый метод быстрого поиска маршрута вывода на основе использования логической сети правил. Суть этого метода в преобразовании логической сети правил миварного пространства в многополюсную сеть теории графов и поиска ее минимального разреза. Анализ известных алгоритмов поиска минимального разреза сетей и дополнительные исследования позволили значительно повысить оперативность обработки данных и разработать квадратичной сложности метод поиска минимального разреза многополюсных сетей.

Оценка возможностей известных методов распараллеливания множественного доступа к общей базе данных выявила необходимость решения проблемы обеспечения реального параллельного доступа к базе данных для обеспечения оперативности потоковой обработки данных в САС ИВК. В результате проведенных исследований был разработан метод максимального распараллеливания множественного доступа к

общей базе данных в условиях недопущения взаимного искажения используемых данных на основе построения виртуальных потоковых баз данных.

Анализ дополнительных возможностей повышения оперативности обработки информации в САС ИВК выявил возможность существенного ускорения обработки при решении некоторых классов переборных задач. В результате был разработан новый метод (способ) алгоритмической минимизации количества вычислительных процедур и устройств сложения, позволяющий перейти от степенной к линейной зависимости общего количества операций сложения от разрядности чисел.

Таким образом, для завершения разработки теоретических основ адаптивного синтеза конфигураций компьютерных систем, т.е. для построения САС ИВК, осталось предложить только научно-технические принципы построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций для решения сложных, специальных и уникальных диагностических задач.

Рассмотрим более подробно принципы построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций - САС ИВК.

### 8.5. ПРИНЦИПЫ ПОСТРОЕНИЯ И ПРИМЕНЕНИЯ СИСТЕМ АДАПТИВНОГО СИНТЕЗА ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ КОНФИГУРАЦИЙ

Система адаптивного синтеза информационно-вычислительных конфигураций предназначена для формирования оптимально возможных проблемно-ориентированных конфигураций многопроцессорных вычислительных систем, на основе:

- как классификации решаемых сложных специальных задач,
- так и анализа существующих и доступных компонент многопроцессорных вычислительных систем с учетом их совместимости.

Следовательно, в системе адаптивного синтеза информационно-вычислительных конфигураций должны храниться данные и модели, описывающие как специальные задачи, так и возможные конфигурации элементов, блоков и средств многопроцессорных вычислительных систем и ВВК.

Таким образом, система адаптивного синтеза информационно-вычислительных конфигураций **может быть отнесена к классу автоматизированных систем сбора и обработки информации (АССОИ).**

В области исследования АССОИ работали многие специалисты (Калиниченко Л.А., Крылов Г.О., Рывкин В.М., Замулин А.В., Попов Э.В., Пospelов Д.А., Шеремет И.А., Цаленко М.Ш., Атре Ш., Дейт К., Мартин Дж., Смит Дж., Тиори Т., Фрай Дж., Ульман Дж., Цикритзис Д.С., Лоховски Ф.Х. и др.) [Л. 1-13, 15-28, 30-48, 54-57, 62-66, 70-80, 86-98, 100-118, 120-132, 135-152, 154-167, 170-176, 178, 183-185, 189-216, 224-228, 231-245, 247-253, 255-258, 261-264, 268-271, 274-280, 284-300, 304-311, 313-319, 321-328, 342-426]. Однако, результаты этих исследований не являются достаточными для решения проблемы создания САС ИВК.

Прежде всего, выделим следующие основные принципы построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций: возможность перестройки архитектуры системы в целом, распределенность обработки и памяти, модульность, наращиваемость, непрерывность функционирования, живучесть, многопроцессорность, универсальность, возможность неограниченного накопления и обработки в едином унифицированном формате любых данных и правил, максимальная оперативность обработки пото-

ков информации (возможно после некоторого периода настройки системы), синергетическая адаптивность, самонастраиваемость по определенным критериям оптимизации функционирования, самоанализ и активное формирование требований по изменению своих ресурсов и конфигурации.

Все изменения и дополнения в любой подсистеме не должны прерывать функционирование системы в целом, независимость от конкретных технических средств, открытость системы. Отметим, что органичное сочетание требований универсальности и максимальной оперативности возможно на основе применения компьютерных систем с массовым параллелизмом, программируемой архитектурой и структурно-процедурной обработкой информации.

Рассмотрим более подробно следующие **основные принципы построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций.**

1. **Адаптивность архитектуры** - исходя из общих требований адаптивности синтеза компьютерных конфигураций следует, что система адаптивного синтеза информационно-вычислительных конфигураций сама по себе должна быть адаптивной автоматизированной системой сбора и обработки информации, т.е. АССОИ.

2. **Непрерывность функционирования** - исходя из требований, предъявляемых к непрерывности процесса решения сложных специальных задач, принципиально важно, что функционирование систем адаптивного синтеза информационно-вычислительных конфигураций должно быть непрерывным.

3. **Живучесть** - система адаптивного синтеза информационно-вычислительных конфигураций должна быть устойчива к отказам отдельных технических устройств, блоков, средств и т.п., обеспечивая автоматическую переконфигурацию системы и непрерывное решение задач на исправных, доступных технических средствах.

4. **Наращиваемость вычислительной мощности** - так как финансирование закупок технических средств ведется постоянно, непрерывно, но периодически, т.е. через определенные (иногда, различные по длительности) интервалы времени, то целесообразно постепенно добавлять в систему адаптивного синтеза информационно-вычислительных конфигураций новые технические блоки и т.п.

5. **Модульность** - наращиваемость вычислительной мощности, т.е. добавление новых блоков, представляется целесообразным осуществлять в виде функциональных модулей.

6. **Многопроцессорность** - для обеспечения производительности, адаптивности, живучести, непрерывности и надежности система адаптивного синтеза информационно-вычислительных конфигураций должна быть многопроцессорной.

7. **Неоднородность** - для реализации адаптивности систем адаптивного синтеза информационно-вычислительных конфигураций в условиях непрерывности функционирования и с учетом обязательной модернизации технических средств, САС ИВК должна быть неоднородной многопроцессорной системой.

8. **Активное формирование требований по изменению своей конфигурации** - так как система адаптивного синтеза информационно-вычислительных конфигураций предназначена для синтеза конфигураций компьютерных систем (т.е. ИВК), следовательно, задача синтеза (самосинтеза) структуры самой САС так же может быть ею решена, если САС ИВК будет иметь возможность активно, т.е. самостоятельно, формировать требования к Заказчику (метасистеме) по изменению своей конфигурации.

9. **Независимость от конкретных технических средств** - система адаптивного синтеза информационно-вычислительных конфигураций должна адапти-

роваться и функционировать непрерывно и постоянно, следовательно всегда должно быть несколько вариантов по изменению конфигурации и модернизации конкретных технических средств.

10. **Открытость** - в общем случае такая сложная техническая многофункциональная система, как система адаптивного синтеза информационно-вычислительных конфигураций, должна иметь возможность взаимодействовать с различными техническими системами и средствами, т.е. быть "открытой" технической системой.

11. **Универсальность представления данных** - подсистема обработки и накопления информации системы адаптивного синтеза информационно-вычислительных конфигураций, основанная на технологиях баз данных, должна быть универсальной для любых предметных областей, так как теоретически САС ИВК должна иметь возможность обучения и работы в любых предметных областях.

12. **Общность и единство представления данных** - подсистема обработки и накопления информации системы адаптивного синтеза информационно-вычислительных конфигураций должна быть общей и единой, унифицированной для всех предметных областей.

13. **Адаптивность обработки данных** - для своевременного реагирования на изменения предметной области и окружающей среды, процессы обработки информации должны адаптивно управляться потоком входных данных.

14. **Распределенность обработки** - так как система адаптивного синтеза информационно-вычислительных конфигураций является многопроцессорной вычислительной системой, то, в общем случае, одновременная обработка данных должна быть организована на территориально распределенных вычислительных средствах.

15. **Активное управление сбором данных** - при решении сложных специальных задач, система адаптивного синтеза информационно-вычислительных конфигураций должна активно управлять процессом сбора необходимой исходной информации, формируя, таким образом, единую систему обработки информации типа "активная логическая сеть правил, управляемая потоком данных".

16. **Эволюционность представления данных и правил** - так как система адаптивного синтеза информационно-вычислительных конфигураций должна функционировать непрерывно, следовательно, и новые данные, и новые правила должны вводиться и встраиваться в подсистему обработки и накопления информации эволюционно, без перезагрузки и перепланирования баз данных и знаний.

17. **Обучаемость** - исходя из необходимости адаптивности и обучения различным предметным задачам в условиях непрерывности функционирования, следует, что подсистема накопления информации системы адаптивного синтеза информационно-вычислительных конфигураций должна иметь изменяемую синергетическую эволюционную структуру представления данных и правил.

18. **Глобальность**, т.е. возможность неограниченного накопления любых данных и правил - так как система адаптивного синтеза информационно-вычислительных конфигураций должна быть обучаемая различным задачам, без теоретического ограничения объема хранимых данных, значит, в подсистеме накопления и хранения данных системы адаптивного синтеза информационно-вычислительных конфигураций необходимо предусмотреть возможность неограниченного эволюционного накопления любых данных и правил.

19. **Максимальное быстродействие обработки потоков данных** - этот принцип является желательным и следует из необходимости оперативного реше-

ния сложных и специальных задач на основе применения систем адаптивного синтеза информационно-вычислительных конфигураций.

Целесообразно продолжить научные исследования в направлении разработки принципов построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций. В процессе дальнейших исследований возможно выявление новых принципов построения и применения систем адаптивного синтеза компьютерных конфигураций. Однако, указанных выше принципов достаточно для реализации и внедрения систем адаптивного синтеза информационно-вычислительных конфигураций.

Главное заключается в том, что **система адаптивного синтеза информационно-вычислительных конфигураций должна быть интеллектуальной неоднородной многопроцессорной универсальной развивающейся синергетической автоматизированной системой сбора и обработки информации, в которой можно наращивать, заменять и модернизировать любые подсистемы, модули, блоки, элементы и средства, а также данные и правила их обработки.**

На данном этапе исследований представляется возможным вернуться к обсуждению проблем создания систем искусственного интеллекта, с учетом полученных нами выше научных результатов. Рассмотрим роль ЭВМ при обработке информации и представлении теории искусственного интеллекта в качестве новой теории активного отражения.

## 8.6. РОЛЬ ЭВМ В ПРОЦЕССАХ ОБРАБОТКИ ИНФОРМАЦИИ

В настоящее время ученые стремятся, как можно шире автоматизировать процессы обработки информации в различных областях. Проанализируем возможности и пределы автоматизации мыслительной деятельности. Прежде всего, рассмотрим вопрос о характере взаимодействия между человеком и ЭВМ в АССОИ и интеллектуальных системах. В качестве элементов в такие системы могут быть включены такие объекты, как человек, вычислительные средства, семиотические и математические модели. Отметим, что использование ЭВМ или моделей предполагает владение определенными навыками, которые частично можно зафиксировать с помощью правил. Однако, действуя по правилам, человек в первую очередь взаимодействует не с машиной, не с объектом своего действия, а с другим человеком, с другими людьми, он вступает в социальное взаимодействие коммуникации. Конечно, он при этом вступает в контакт и с объектом, но это два совершенно различных типа взаимодействия [Л. 415].

Приведем обзор высказываний ученых по интересующей нас теме. Итак, А. Г. Спиркин считает, что выполнение машиной сложных логических операций не есть мышление в строгом смысле этого слова, так как живой мозг решает задачи совсем по другому [Л. 267]. Г. Клаус пишет [Л. 414], что мышление есть функция высшей формы движения, качественно отличной от всех форм движения низших ступеней, следовательно, ее сущность никогда нельзя повторить на моделях низшей ступени. Качественное различие между ЭВМ и мозгом нельзя устранить с помощью, какой бы то ни было системы слов или определений. Еще более ярко и понятно им написано следующее: кибернетические машины, хотя и могут состоять из меди (и т.д.), но их структура и способ функционирования не принадлежат к ступени неорганического, а являются структурой и функцией самого мышления, спроецированного человеком в природу [Л. 414, с. 51].

Таким образом, на данном этапе получаем, что машины могут выполнять отдельные функции и даже достигать результатов, превосходящих результаты нашего мозга, но все, что они в состоянии сделать, не является мышлением. Следовательно, полностью исключить человека из АССОИ или интеллектуальных систем не получится, как минимум - на данном этапе развития науки и техники. Впрочем, этого и не требуется, если учитывать знаменитое положение Н. Винера: машине - машинное, человеку - человеческое [Л. 52].

Попробуем проанализировать, какие же функции в АССОИ являются принципиально "человеческими", а какие можно отдать машине. Для ответа на этот вопрос необходимо выяснить, что же такое "информация" в широком (интеллектуальном) смысле этого слова.

Начнем с анализа философских основ понятия информации. Информацию нельзя просто причислить к сознанию, так как хотя семантика (значение, смысл) информации субъективна, но ее субъективность имеет объективную основу. С другой стороны, информацию нельзя отнести к сфере материи (к сфере объективного) потому, что материальная, объективная сторона информации явно является вторичной. Первоначально, "информацией было то, что имело какую-то связь с человеческим сознанием" [Л. 414, с. 77]. Безусловно, сознание причастно к созданию, порождению информации. Но, будучи когда-то созданной, информация получает до некоторой степени самостоятельную жизнь. Таким образом, информация - это единство субъективного и объективного, которое состоит из объективного (материального, физического) носителя и "наложенного" на него субъективного содержания, семантики. С философской точки зрения в этом нет ничего удивительного. Более того, исходя из основных законов, прежде всего - единства и борьбы противоположностей, именно этого и следовало ожидать.

Таким образом, по своей природе информация - есть противоречивое и неразрывное единство объективного и субъективного. На материальной основе носителя "семантика информации обладает способностью вызывать психический эффект, и только поэтому мы говорим о семантике... Чисто физические действия, появляющиеся в замкнутой системе управления, не имеют никакой семантики" [Л. 414, с. 89-90]. В качестве примера можно привести регулятор центробежной силы в паровой машине Уатта, не передающий никакой семантики.

Для анализа взаимодействия человека и ЭВМ в процессе переработки информации большую роль играет следующее положение Г. Клауса: "техническая конструкция ЭВМ и моделированное повторение определенных логических операций - это еще не счетная операция и не логическая операция. Мысль остается мыслью, а ее физическая модель остается физической моделью. При построении физической модели, мысль не превращается в нечто такое, что не является мыслью" [Л. 414, с. 91].

Отсюда следует принципиальный вывод о том, что семантика информации, созданная сознанием, не исчезает в области объективных (материальных) сущностей бесследно, так же как она и не возникает там. Если **информация (И)** представляет собою единство (субъективного) **семантики (СЕМ)** и (объективного) **носителя (Н)**, то есть

$$\mathbf{И} = (\mathbf{СЕМ}, \mathbf{Н}),$$

то машина только физически оперирует с носителями. При этом здесь нет никакой информации в широком (интеллектуальном) смысле, а просто осуществляется физический результат [Л. 414].

То же самое можно выразить и другими словами. **Знак**, в нашем представлении (данное положение основывается на материалах В. Г. Атрощенко), состоит из двух частей:

- 1) объективный, материальный носитель и
- 2) субъективный, идеальный образ, "навешиваемый" на материальный носитель.

Причем, знак - это неразрывное единство двух компонент, частей. Таким образом, мы работаем в рамках знакового моделирования, где под знаком (Зн), с помощью которого выражается информация, будем понимать следующее единство:

$$\text{Зн}=(\text{СЕМ}, \text{Н}),$$

то есть субъективной семантики "навешенной" на конкретный объективный носитель. Таким образом, получаем, что **ЭВМ оперирует лишь с носителями знаков, но ни коим образом не с самими знаками.**

Это принципиально важное положение позволяет сделать вывод о том, какие функции в АССОИ и интеллектуальных системах являются принципиально человеческими и не могут быть переданы машине.

Как только возникает необходимость порождения знака в процессе моделирования или интерпретации результатов, полученных при оперировании ЭВМ с материальными носителями знаков, то можно утверждать, что пока такие функции может выполнять только человек. От человека зависит и выбор материальных объективных носителей знаков (информации), будь то чернила на бумаге или последовательность электрических сигналов. Отметим, что совершенно обоснованно при обработке информации в АССОИ отдельно и специально выделяют **данные** и базы данных, в качестве носителя обрабатываемой в ЭВМ информации. От выбора материальных носителей зависит возможность проведения операций с ними, обеспечение хранения информации, а также повышение скорости обработки данных.

В принципе, человек может навешивать (назначать, определять и т.п.) любое значение на любой материальный носитель, образуя таким образом знаки. Кроме того, отдельной темой является выбор количества навешиваемой информации и формирование сложных или составных знаков. Конечно, при этом необходимо учитывать то, что человек работает не один и его должны понимать другие люди. В противном случае информация теряет свою коммуникативную функцию. Впрочем, в некоторых случаях люди, наоборот, стараются спрятать, зашифровать информацию, что бы она была доступна ограниченному кругу лиц, но это другая тема исследований.

В этом случае, например, Г. Клаус специально подчеркивал, что "информация не есть нечто абсолютное; то, что для одного какого-либо сознания играет роль информации, для другого сознания ... не является информацией" [Л. 414, с. 102-103]. Однако, существует закономерная связь между семантикой и носителем информации. Семантика информации субъективна, но ее субъективность имеет объективную основу. Эта объективная основа делает так же понятным, почему вообще могут быть расшифрованы системы информации, семантика которых сначала абсолютно неизвестна. Во всех этих случаях мы в начале знаем лишь физический носитель информации и его структуру. Если информация была бы чисто субъективной, то при исчезновении (смерти) субъективного сознания, создавшего эту информацию, должна была бы навсегда погибнуть и семантика этой информации. К счастью, как доказала история развития науки [Л. 414], этого не бывает.

Таким образом, приходим к выводу, что информация не является чем-то особенным, независимым, третьим компонентом, в которой элементы субъективного (сознания) и объективного (материи) слиты друг с другом особым образом.

В настоящее время, в автоматизированных системах сбора и обработки информации на самом деле с информацией работает только человек, а в ЭВМ обрабатывается только лишь специальным образом структурированный объективный (материальный) носитель информации, который принято называть - данные. Взаимодействуя с ЭВМ человек, на самом деле человек вступает в отношении коммуникации с другими людьми, написавшими программное обеспечение для этой ЭВМ. Роль ЭВМ в процессах обработки информации в АССОИ сводится только к обработке объективных (материальных) носителей информации, т.е. только к обработке, передаче и хранению данных. Функции ввода/вывода (интерпретации) информации в АССОИ, в настоящее время, принципиально может исполнять только носитель субъективного сознания, т.е. человек.

В следующем разделе данной работы представляется целесообразным, с учетом полученных нами выше новых научных результатов, вернуться к обсуждению проблем создания систем искусственного интеллекта.

Рассмотрим представление теории искусственного интеллекта в качестве новой теории активного отражения.

## **8.7. СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ТЕОРИЯ АКТИВНОГО ОТРАЖЕНИЯ**

### **8.7.1 ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ - КАК НАУЧНОЕ НАПРАВЛЕНИЕ**

Определить искусственный интеллект как научное направление - это значит, прежде всего, определить предмет и метод искусственного интеллекта.

Уточним, что такое предмет науки. "Предмет познания - зафиксированные в опыте и включенные в процесс практической деятельности человека стороны, свойства и отношения объектов, исследуемые с определенной целью в данных условиях и обстоятельствах" [Л. 303, с. 379]. Проанализируем это понятие. Прежде чем что-то познавать, изучать, исследовать, необходимо выяснить, показать, доказать, что это "нечто" действительно есть, существует, т.е., что мы можем каким-либо образом "пощупать его руками" непосредственно или достоверно убедиться, отразить его существование посредством либо наших ощущений, либо специальных приспособлений, устройств, приборов. Итак, это самое "нечто" мы обнаружили, следовательно "оно" должно быть чувственно - отражаемой, ощущаемой объективной реальностью. В этом выражается объективная сторона понятия "предмет познания".

Все свои ощущения субъект постепенно анализирует, обобщает и выделяет в них основное, повторяющееся, главное, а различные случайности и помехи отбрасываются. В результате этого, у познающего создается некое представление об изучаемом "нечто", хотя оно еще довольно смутно и не совсем понятно. Познающий старается уяснить и точно определить, что же это самое "нечто" собой представляет. Этот процесс опирается уже на все ранее известное, познание, на знания познающего субъекта. Субъект пробует, пытается выразить в терминах естественного языка, знаками описать то, что он ощутил, понял, узнал. Он старается определить, как можно точнее, свое полученное представление. В результате, у познающего субъекта получается языковое выражение, символическое представление изучаемого "нечто". В это формальное, символическое выражение он вкладывает (или "навешивает на ...", соотносит с ...) вполне определенное содержание, отвечающее и соотносящееся со всеми его знаниями.

Поскольку это выражение зависит от представлений субъекта, его знаний и от его способности ощущать, воспринимать, отражать реальность, постольку это представление вносит свое субъективное влияние. Следовательно, выражение - представление - понятие об изучаемом "нечто" является единством объективного и субъективного, при доминирующей роли объективного. Оно объективно потому, что отражает объективную реальность, но оно и субъективно, так как эту самую отражаемую реальность субъект воспринимает через свое представление, основываясь и исходя из всего ранее познанного им и из своих отражательных способностей.

Приведем пример. Древние ученые изучали мир. Предметом их исследований была окружающая среда. Возникает вопрос: являлись ли такие явления, как электричество, ядерные реакции и т.п. предметом их изучения? Все эти явления существовали и тогда. Ответ будет таким: эти явления не являлись предметом познания, так как древние ученые только могли догадываться, подозревать об их существовании. Следовательно, предмет познания зависит и от возможностей познающего субъекта.

Остается добавить, что познающий субъект изучает "нечто" с какой-то целью, которая, может быть, даже и не совсем ясна и понятна ему самому. Но цель у него обязательно есть! Итак, обобщим все вышеизложенное в данном пункте.

Предмет изучения - это чувственно отражаемые стороны, части, свойства и отношения объектов, изучаемые с определенной целью, обобщенно выделенные в знаковом представлении, которое постоянно наполняется содержанием, соответствующим развивающемуся знанию как о непосредственно изучаемом, так и обо всей объективной реальности в целом.

Предмета изучения не существует вне процесса познания. Предмета изучения нет и без познающего субъекта. Предмет отражает представление познающего, а его содержание постоянно обновляется, дополняется, хотя форма выражения, конкретное символическое представление такой формы может быть неизменным.

Итак, предмет познания объективен, как часть объективной реальности, но он и субъективен, поскольку эта объективная часть должна быть еще и выделена познающим субъектом. Следовательно, предмет познания существует лишь в единстве познающего и познаваемого. Символическое выражение, представление познаваемого предмета должно быть кратким и отражать самую суть.

### 8.7.2. ПРЕДМЕТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Рассматривая различные **"интеллектуальные"** процессы такие, как мышление человека, его психическая деятельность, а также процессы генерации знаний, переработки информации не "человекоподобными" методами, приходим к выводу о том, что все эти процессы **являются процессами отражения окружающей действительности**. Но, у любого материального объекта есть свойство отражения. Под отражением мы понимаем некую реакцию на взаимодействие с другими объектами. Главным, принципиальным отличием "интеллектуального" отражения является то, что "интеллектуальный" объект активно реагирует, перерабатывает, отражает воздействие других объектов. Такие **процессы "интеллектуального отражения действительности" будем, в дальнейшем, называть процессами активного отражения**.

Введя понятие "активного отражения", мы можем рассматривать "интеллектуальные" процессы без учета человекоподобия, антропоморфности известных нам интеллектуальных систем, т.е. человеческого мышления. Таким образом, пока нам известны, только системы естественного интеллекта, но исходя из предположения о том, что могут существовать и другие интеллектуальные (в том числе и

искусственные) системы, мы вводим понятие "активного отражения". **Под интеллектуальными системами будем понимать такие системы, у которых есть способность к активному отражению.** Отметим сразу, что, как существуют различные стадии, ступени, степени естественного интеллекта, так, по всей видимости, должны **существовать и степени активности отражательной способности** различных систем.

Подробнее опишем, что мы понимаем под активным отражением. Система, способная к активному отражению действительности, самостоятельно формирует некий комплекс целей, самостоятельно выбирает объект изучения и целенаправленно отражает, познает, изучает его посредством своих органов чувств, датчиков информации. Важно отметить, что при активном отражении у субъекта появляется цель. Причем, цель может быть порождена внутри этого субъекта, например, в случае мышления человека или поведения животных. Также, цель может быть заложена в отражающую систему внешним субъектом, например, при обработке информации в ЭВМ, программа для которой написана человеком. Возникает проблема того, что различные системы обладают и различной степенью активности. Например, всякая интеллектуальная система - активна, но далеко не каждую активную систему можно считать интеллектуальной, в прямом смысле этого слова. В дальнейшем, можно ввести некоторую единицу измерения степени активности различных систем, что-то, образно выражаясь, наподобие коэффициента интеллектуальности. Данный вопрос еще не достаточно проработан и требует дальнейшего изучения.

Кратко поясним, что мы понимаем под "**активностью**". В работе Ю. В. Орфеева и В. С. Тюхтина [Л. 222] написано о том, что активность как состояние некоторой системы - это состояние нужды, потребности в тех или иных полезных агентах среды или же в реакциях, спасающих от внешней угрозы, отрицательного воздействия". Суть ориентировочной активности в устремленности субъекта во вне, в мир вещей и процессов, откуда могут исходить как полезные агенты, удовлетворяющие ту или иную потребность, так и вредные, угрожающие ... . Состояние ориентировочной активности представляет собой основу, ключ к пониманию мыслительной, интеллектуальной деятельности, психического поведения животных и человека" [Л. 222, с. 45 - 50]. При активном отражении порождаются те существенные явления, феномены, присущие процессам мышления и творчества, которые не формализуются, не моделируются на уровне информационно-логических программ. Как сказано в работе [Л. 222], состояние органических и ориентировочных потребностей есть высшее выражение целостности организма в его активном отношении к внешней среде. Потребности порождают цели и задачи, которые субъект решает, и благодаря этому его поведение носит целенаправленный и целесообразный характер. Современные машины, не имеющие потребностей, не могут самостоятельно вырабатывать цели, ставить новые задачи. Цели и критерии поведения вкладываются в них человеком из вне, в виде программ, следовательно, они не обладают активностью.

Итак, понятие "**процесс активного отражения**" **вводится как более широкое, более емкое понятие, чем "интеллектуальные процессы" или "мышление"** и, в то же время, оно содержит основное, главное, что характеризует эти процессы, причем без всякого влияния антропоморфизма понятий "интеллект" и "мышление". Теперь мы можем сделать вывод о том, что **предметом искусственного интеллекта как научного направления - являются процессы активного отражения.**

### 8.7.3. МЕТОД ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Прежде всего, метод (греч. *methodos* - буквально "путь к чему-либо") - это "в самом общем значении - способ достижения цели ... Метод как средство познания есть способ воспроизведения в мышлении изучаемого предмета" [Л. 303, с. 278]. Таким образом, имея перед собой предмет познания, мы задаемся целью изучить его, понять, что он собой представляет. Для этого нам необходимо определить, выбрать путь к цели, а также способ и средства преодоления этого пути. Познание, изучение, понимание предмета - это процесс описания и объяснения, хотя бы даже и себе самому, этого предмета, т.е. воспроизведение в мышлении познаваемого предмета. Следовательно, под методом понимается как способ, так и сам процесс достижения цели. Таким образом, метод представляет собой единство способа, средств и процесса достижения цели. Учитывая это, предлагается следующее определение.

**Метод - это способ организации деятельности для достижения цели научного объяснения предмета исследования, посредством воспроизведения этого предмета в мышлении, в виде определенным образом организованного, символьного, знакового описания.**

Рассматривая ИИ - как научное направление, приходим к выводу, что **основным методом является моделирование**, причем в самых различных формах, от формализованного знакового моделирования до построения и создания различных физических, кибернетических, биологических и т.д. моделей. Важную роль играет и системно-структурный подход, как общенаучный методологический принцип.

По нашему мнению, основным методом теории искусственного интеллекта является формализованное знаковое моделирование, которое имеет следующие два аспекта:

- 1) формализация;
- 2) знаковое представление модели.

Модель, в данном случае, рассматривается как отражение, обобщение субъектом предмета познания, т.е. мышления или процессов активного отражения. При этом, мышление определяется, представляется логикой, по крайней мере, в области осознанного мышления, т.е. сознания.

Рассмотрение мышления, как предмета теории ИИ, предполагает две стороны. С одной стороны, это определенная логическая система - система знаний или, просто, знания, т.е. логические формы и отношения между ними. С другой стороны, нельзя забывать и о процессе движения, изменения логической системы, что приводит к изменению системы знаний, в том числе и к генерации, порождению новых знаний.

Формализация предполагает построение некоторой формально-логической системы, которую в свою очередь, можно определить как формальную систему знаний. Знаковое представление необходимо для материализации, фиксации, определенности полученной системы знаний (данный пункт основан на различных работах и выступлениях В. Г. Атрощенко). Не исключено, что в области ИИ будут применяться и другие научные методы исследований.

#### 8.7.4. ТЕОРИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА - КАК ТЕОРИЯ АКТИВНОГО ОТРАЖЕНИЯ

Итак, предметом теории ИИ являются процессы мышления или, в более широком смысле, процессы активного отражения. По нашему мнению, с учетом истории и неоднозначности понятия "искусственный интеллект", можно заменить, в контексте данной работы, понятие "теория ИИ" на новое понятие "теория активного отражения". Такая замена позволит уйти от столь неоднозначного, двусмысленного термина "ИИ" и освободиться от необходимости постоянно напоминать, что эти два слова используются в таком-то смысле. К тому же, такая замена позволит не употреблять столь антропоморфный, непосредственно связанный, ассоциирующийся с человеком, термина "интеллект". Да и слово "искусственный", на наш взгляд, не самое лучшее, хорошее для названия научного направления. **Термин "Искусственный Интеллект" был хорош на ранних стадиях развития данного научного направления, но ... сейчас, этот термин вносит лишь сумятицу, неоднозначность, играет, скорее, отрицательную роль.**

Будем писать "искусственный интеллект", там где это соответствует общепринятому, в узком смысле, понятию, т.е. как техническая информационная модель естественного интеллекта, а, когда будем говорить о научном направлении, тогда будем использовать термин "теория активного отражения".

Предметом теории активного отражения (ТАО) являются процессы активного отражения, в том числе и мышление, и сознание. Основным методом ТАО является формализованное знаковое моделирование. Мышление является предметом многих наук, например таких, как биология, логика, гносеология, физиология высшей нервной деятельности и другие. Каждая из этих наук изучает мышление своими методами и для своих целей, т.е. со своей точки зрения.

**Целью теории активного отражения** является изучение процессов мышления, как процессов порождения, накопления, обработки, обмена, хранения информации и, если это возможно, то создание искусственных систем, сравнимых по своим возможностям с человеческим мышлением. ТАО должна дать обоснованный ответ на известный вопрос: "может ли машина мыслить?". ТАО должна вобрать в себя достижения и результаты самых разнообразных научных дисциплин. Масштабы области интересов и исследований ТАО характеризуются самым широчайшим диапазоном. В этом и заключается основная проблема ТАО: собрать все, что известно о мышлении, переработать, проанализировать, обобщить эти материалы, выделить все существенное, с точки зрения ТАО, а затем, наметить пути и способы решения поставленных задач.

Дело в том, что говорить о теории активного отражения, о создании систем искусственного интеллекта, не изучая при этом естественный интеллект, мышление и сознание человека, было бы не верно. С точки зрения ТАО, **человеческий интеллект - это лишь одна из существующих форм мышления, активного отражения.** Поэтому ТАО не должна замыкаться только на изучении человеческого интеллекта, который является, всего лишь, хорошим готовым образцом системы активного отражения (и ничем более того). В этом и заключается **основное отличие ТАО от других наук**, изучающих мышление. Кроме того, существует целый ряд проблем, вопросов, которые не рассматривались в других науках. Вероятно, многие проблемы, которые даже затрагивались другими науками, нам придется решать, прорабатывать заново, исходя из целей ТАО.

Например, рассмотрим такую проблему. Существует реальная объективная действительность и есть мышление человека, нас же интересует процесс отражения. Необходимо выяснить, узнать, во-первых, **что отображается** из действительности, во-вторых, **во что это отражается** в мышлении и, в-третьих, **как происходит** процесс отражения. Причем, нас интересует кибернетическая, информационная сторона всего этого. Кроме того, важно понять и процессы обработки данных, информации в ЭВМ.

Рассматривая проблему "мыслящих машин", мы неизбежно столкнемся с такой философской проблемой: как, каким образом объективная (материальная) система порождает или обрабатывает субъективное (идеальное), генерирует новые знания. В таком аспекте вопрос взаимодействия материального и идеального подробно не разрабатывался, хотя, вероятно, именно от этого и зависит ответ на вопрос о возможности создания искусственных мыслящих машин, искусственного интеллекта.

**Проблему порождения материальным объектом субъективных, идеальных знаний** никаким образом не обойти, остается только познать, изучить и решить эту, казалось бы, чисто философскую, проблему на техническом уровне. Отметим тот факт, что до сих пор исследования в области искусственного интеллекта велись разрозненно, без единых целей и задач, да и к тому же, в ограниченных масштабах. Теория активного отражения должна устранить недостатки предыдущих исследований и стать тем стержнем, вокруг которого и объединятся интересы, цели, задачи всех направлений и научных дисциплин, которые занимаются исследованием возможности создания "разумных, думающих машин" - искусственного интеллекта.

#### **8.7.5. ОСНОВНЫЕ НАПРАВЛЕНИЯ РАЗВИТИЯ ТЕОРИИ АКТИВНОГО ОТРАЖЕНИЯ**

Анализируя уже известные подходы в области ИИ и используя новые положения, можно прийти к выводу о том, что существует три основных направления.

**1. Прикладное направление.** Основная цель - разработка и создание технических систем, которые могут решать отдельные задачи высокого уровня сложности и, таким образом, эти системы должны являться дополнением естественного интеллекта, которое позволяло бы **усиливать интеллектуальные способности человека**. Основное отличие данного направления в том, что не ставится задача создания автономных интеллектуальных систем, а решаются задачи моделирования отдельных интеллектуальных функций таких, как представление знаний, планирование целесообразного поведения, распознавание образов и обучение, общение человека и ЭВМ. В это направление входят все работы, которые рассматривает Д. А. Поспелов в предисловии к работе [Л. 340, с. 10 - 12], а также это близко к первому направлению классификации, предложенной С. М. Шалютиным в работе [Л. 324].

**2. Кибернетическое (информационное) моделирование мышления.** Основная цель - разработка и создание технических, кибернетических, **математических моделей мыслительных процессов**. Важен не результат, которого достигали бы такие модели, а сам процесс получения этого результата. Данное направление близко ко второму направлению классификации, обсуждаемой выше и предложенной С.М. Шалютиным [Л. 324], а также работам ДонГИИИ [Л. 47].

**3. Общетеоретическое направление.** В него выделяются работы и исследования по созданию общей кибернетической **теории мышления, искусственного интеллекта или активного отражения**. Это направление является наиболее абстрактным и наименее проработанным. Например, одной из задач данного направления является создание модели (или моделей) человеческого интеллекта, мышления, а, в перспективе, и решение задачи создания мыслящей, разумной машины. Это направление близко третьему направлению классификации, приведенной выше и предложенной С. М. Шалютиным [Л. 324, с. 10 - 11].

### **8.7.6. ИСТОКИ, ЦЕЛИ И ЗАДАЧИ ТЕОРИИ АКТИВНОГО ОТРАЖЕНИЯ**

Для начала определим истоки, отправные точки теории активного отражения. Общий подход, направление исследований нам дает философия. Предметная область, которую мы исследуем и моделируем, описывается естественными науками, хотя, вполне вероятно, что должны привлекаться гуманитарные науки и, даже, должен учитываться весь накопленный мировым искусством опыт. Это объясняется тем, что человеческий интеллект, мышление проявляет себя в самых различных формах, и мы не можем заранее точно установить границы области мыслительной деятельности человека.

**Критерием адекватности**, проверки правильности и полноты наших моделей, является практика, прикладные исследования, проводимые в первом или во втором, а может быть и совместно, направлении нашей классификации. Необходимо отметить, что только комплексное, органичное сочетание проводимых работ во всех трех направлениях позволит нам добиться успеха.

Рассмотрим цели и задачи, которые ставятся перед теорией активного отражения.

Во-первых, стратегической, фундаментальной целью ТАО является научное объяснение мыслительного процесса и природы возникновения активности отражения, оценка возможности передачи мыслительных функций техническим системам, машинам.

Во-вторых, теоретическая, естественно - научная цель - это познание механизма различных отдельных функций мозга и переработки информации, разработка и создание моделей этих отдельных функций.

В-третьих, практическая, техническая цель - это решение насущных, неотложных задач высокой степени сложности, с которыми естественный интеллект не может справиться без помощи технических средств, ЭВМ.

Отметим, что сущностью этих задач является автоматизация деятельности (в том числе и умственной) человека, в широком смысле этого слова, которая приведет к расширению возможностей и усилению способностей человеческого мышления. Еще раз подчеркнем, что проблема ИИ (ТАО) имеет глубокую связь со многими науками.

### **8.7.7. ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ СОЗДАНИЯ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА С ТОЧКИ ЗРЕНИЯ ТЕОРИИ АКТИВНОГО ОТРАЖЕНИЯ**

С точки зрения теории активного отражения любая система, претендующая на название "система искусственного интеллекта", обязательно должна содержать следующие подсистемы:

- 1) приема и выделения необходимой, важной информации, т.е. подсистема ввода и распознавания;
- 2) обработки, получения новой информации внутри системы, т.е. подсистема обучения;
- 3) накопления и хранения необходимой информации, т.е. подсистема представления знаний;
- 4) выработки целей и принятия решений, т.е. подсистема целеположения;
- 5) обмена информацией с другими системами, т.е. подсистема общения;
- 6) поддержания целостности системы, т.е. подсистема жизнеобеспечения;
- 7) воздействия на окружающую среду, т.е. подсистема реализации своих принятых решений.

Основная проблема, затруднение при разработке систем искусственного интеллекта будет заключаться в области знакового моделирования. Одной из основных отличительных черт интеллектуальных систем является способность к генерации, порождению знака. Более того, мы убеждены, что системы, не обладающие такой способностью не могут называться "интеллектуальными".

**Знак**, в нашем представлении (как было указано выше, этот тезис основывается на материалах В. Г. Атрощенко), состоит из двух частей:

- объективный, материальный носитель и
- субъективный, идеальный образ, "навешиваемый" на материальный носитель.

Причем, знак - это неразрывное единство двух компонент, частей. Следовательно, **для генезиса, порождения знака некоторая система должна быть способна выделить этот знак материально, а затем и сразу же, наделить его, "навесить" на него идеальный образ, некую интерпретацию данного знака, содержание.** Без интеллектуального, мыслящего субъекта нет знака, равно, как и нет знака без материального носителя, т.е. чисто в идеальном образе. Кроме того, в некотором смысле знак - это процесс, процесс взаимодействия двух или более субъектов, либо на расстоянии, либо во времени, либо на расстоянии и во времени.

Данный вопрос требует особого рассмотрения и изучения. Подчеркнем, что без генезиса знака внутри самой системы, т.е. именно без внутреннего активного генезиса знака, система не сможет работать со знаками, а, следовательно, система не сможет мыслить и перерабатывать знаковую информацию.

Как было показано выше, **современные ЭВМ не работают с информацией** (в широком - интеллектуальном - смысле слова). ЭВМ, всего лишь обрабатывает, преобразует, трансформирует, по определенным правилам - алгоритмам, материальный носитель знака, но не сам знак. Этот материальный носитель становится знаком (в настоящее время), только после того, как человек прочитает, проинтерпретирует его, навесит на него соответствующий субъективный, идеальный образ и, таким образом, породит, создаст знак.

**Технические средства обработки информации**, в том числе и ЭВМ, и средства передачи данных, и т.п., всего лишь, **осуществляют связь между людьми** (например, между программистом и конечным, непрограммирующим пользователем), всего лишь преобразуют или передают материальную основу информации. Следовательно, информации в современных ЭВМ нет, а есть, лишь переработка данных, преобразование электрических сигналов, которые, в свою очередь, являются носителем информации. Информация же существует только до загрузки в машину - в голове программиста, а появляется информация вновь лишь после интерпретации, прочтения - в голове у пользователя.

Поэтому, предлагается следующий критерий определения степени интеллектуальности различных систем: если некоторая система способна к порождению, генезису знака и, следовательно, способна обрабатывать непосредственно информацию, значит она - интеллектуальна. Если же не способна, то система не является интеллектуальной, т.е., например, обладает "отрицательной или нулевой степенью активности" или "интеллектуальности". Отметим, что проблема введения и определения степени "активности" или "интеллектуальности" требует особого рассмотрения, не входящего в рамки данной работы.

При таком подходе, системами искусственного интеллекта (СИИ) имеют основания называться такие искусственные технические системы, которые способны к порождению знака и к обработке информации.

В настоящее время, таких систем нет, а есть только ССИИ - **системы стремящиеся к искусственному интеллекту**. Существующие же, так называемые "**СИИ**", становятся интеллектуальными системами только в том случае, если в их состав, явно или не явно, **включают человека**, оператора, хотя бы даже на начальной или (и) на заключительной стадии обработки информации.

Таким образом, можно заменить вопрос: "могут ли машины мыслить?" на вопрос: "могут ли машины порождать знак?". **Если машина сможет работать со знаками, то она научится и мыслить!** При этом возникает вопрос - проблема измерения, сравнения, определения степени интеллектуальности, разумности таких машин. Ведь даже люди, казалось бы, способные мыслить, имеют различную степень, ступень, уровень интеллектуальности. Существует и такая проблема: останутся ли мыслящие машины всего лишь машинами или мы вынуждены будем считать их членами нашего человеческого общества со всеми правами и обязанностями. Нужны ли вообще мыслящие машины или это - чисто научная, абстрактная цель, которую лучше не воплощать в реальную практику. Большинство этих проблем видится нам в таком отдаленном и фантастическом будущем времени ... , но на все эти вопросы ответы надо искать уже сейчас, уже сегодня, когда только формируются, прорисовываются цели теории активного отражения.

В заключение приведем небольшое отступление, "сказку - аналогию" для "старших" и "младших" научных сотрудников.

### 8.7.8. ПАРОВОЗ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И СТАРИК ХОТТАБЫЧ

Попытки создания систем искусственного интеллекта напоминают нам известную детскую сказку о волшебнике старике Хоттабыче. Представим себе, что старик Хоттабыч увидел обыкновенный паровоз, который тянет за собой огромные тяжелогруженные вагоны. Он видит форму, размеры паровоза, но не знает ни его устройства, ни принципа работы, ни того, что человек-машинист управляет сложнейшей машиной. И вот захотел Хоттабыч сделать, разработать, создать такой паровоз. Ну, а что в этом такого сложного? Вот же, перед нами паровоз, все видно, понятно и просто: дым из трубы идет, значит едет паровоз, а когда дыма нет, то паровоз стоит на месте.

Дернул старик Хоттабыч волос из своей бороды, "дзинь" и ... стоит новенький, блестящий "паровоз", из чистого золота сделанный, дорогими коврами увешанный. Из трубы "паровоза" дым валит, но ... не движется "паровоз", только на одном месте стоит и "сверкает в лучах заходящего солнца". Хорошо, если Хоттабыч, хотя бы, знаком с устройством колес и смог воспроизвести их. Тогда этот

"паровоз" можно будет, по крайней мере, хотя бы, подталкивать и катать. А если удастся поднять, затащить этот "паровоз" на горку, то тогда он даже сам, сам покатится. Но ... не далеко, а только пока спуск не кончится. Потом снова можно поднять "паровоз" на горку, и снова, и снова...

Так же и человек - увидел интеллект, вроде бы выяснил, как он выглядит и решил сделать искусственный "интеллект". Склепал, собрал, сделал. И, вроде бы, внешне все есть у этого "интеллекта". Разрабатывая различные алгоритмы и создавая программы, человек поднимает свое творение - "интеллект" на "горку". Чем больше и чем лучше алгоритмы, тем выше такая "горка". Тем дальше может "самостоятельно" "скатываться", т.е. решать задачи, "паровоз" - "интеллект", причем без помощи человека. Рукоплещут зрители, кричат мальчишки, взволнованно дышат барышни: "Смотрите, смотрите! Машина сама, сама решает задачи!! Браво!!! Победа!?!? Теперь еще можно добавить: "компьютер обыгрывает человека в шахматы!" и т.п. Но, кончилась "горка" и ..., оказывается, что не машина "сама", а человек с помощью машины решал те самые задачи, разрабатывал алгоритмы, писал программы. Какой конфуз! Какая неудача!! Какой провал!!!

Сейчас, наука пошла дальше в изучении "паровоза" - интеллекта. Известно уже, зачем ему колеса, почему дым идет из трубы. Вроде бы теоретически известно, как работает двигатель. Известно, что **любая строго ограниченная область мыслительной деятельности человека, в принципе, может быть формализована, описана алгоритмически и, следовательно, может быть воспроизведена, повторена, проделана машиной, ЭВМ.** Лихо и шустро, с бравым видом, "с блеском" скатываются "паровозы" с высоких, крутых Алгоритмических и Программных "гор". Естественно, скатываются они по рельсам, которые проложил человек. Но, задумался уже человек: "Кто же управляет паровозом?". Вот стоит человек и рассуждает: "Кто (или что) управляет мышлением? Как оно возникло?". А ответ, пока, только такой. Раз сам сделал паровоз - "интеллект", то сам и управляй им, пока ничего, более лучшего не придумано.

Вот так и стал человек - "машинистом" искусственного интеллекта-паровоза. Сам дорогу выбирает, сам уголь в топку подбрасывает. Паровоз же пыхтит, дым в небо пускает и тянет, тянет тяжелый груз "науки", который стал уже непосилен для не оснащенного, невооруженного никакой техникой человека. Когда же перестанет удовлетворять паровоз - слишком медленно, да и груз все тяжелее и тяжелее, не справляется паровоз - тогда, человек сядет, задумается и создаст нечто, более мощное, скоростное - электровоз. Полежай человек опять, в кабину и - в путь, по нелегкому пути Познания. Не будет справляться электровоз? Тогда, человек, создай что-нибудь более пригодное. Вот такой трудный путь. Все в твоих руках! Человек!!!

### **8.7.9. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ - КАК ЦЕЛЬ И ГОРИЗОНТ НАУЧНОГО ПОЗНАНИЯ**

Понятие "искусственный интеллект", в некотором роде, можно сравнить со "средством передвижения". Чем дальше движется человек, тем выше и жестче требования к "средству передвижения", к его возможностям и способностям. Обратим внимание на такой факт. Чем дальше мы развиваемся, тем более совершенными и разнообразными становятся средства передвижения: ноги, телега, карета, автомобиль, паровоз, электровоз, самолет, ракета ... Но и предыдущие, прошедшие этапы забывать нельзя. С появлением автомобиля человек не перестал ходить

пешком, так и, с появлением ЭВМ человек не должен разучиться считать, творить, рисовать, писать тексты и логически рассуждать.

Приведем еще несколько аналогий между паровозом и искусственным интеллектом. Так много в настоящее время существует разнообразнейших средств передвижения. Так различны их назначения, размеры и возможности. И ведь все они нужны! Следовательно, и в области ИИ должно быть разработано и создано много различных по назначению и по возможностям систем. В принципе, сейчас можно создать некое универсальное средство передвижения, хотя это достаточно сложно (но некоторые прообразы таких средств уже есть). Это средство будет ездить, летать и плавать. Мы затратим, "угробим" на его создание массу сил, средств и ресурсов. Но можно уже сейчас: ездить на автомобиле, летать на самолете и плавать на пароходе. Все это уже существует. Зачем же тогда изобретать универсальное транспортное средство.

Универсальный искусственный интеллект, по нашему мнению, подобен такому универсальному средству передвижения, которое возможно, но практически, не очень нужно. Зачем же тогда создавать универсальный ИИ? Только, если в качестве некоего "идеала", цели, "горизонта". Некоторая система ИИ может состоять из множества специализированных, более простых и доступных "искусственных интеллектов" - каждый из которых может выполнять что-то одно, какую-либо конкретную функцию. А в целом, в совокупности, такая обобщенная система, уже сейчас, может доставить нас и на Луну, и в ближайшую булочную, и переправить через океан. Она может практически все, что угодно, да к тому же, это и реальнее, и проще, чем "универсальный транспорт".

Наконец, вспомним о том, для чего все это разрабатывается, конструируется, создается? Все это делается для расширения возможностей человека, для повышения "выживаемости" человеческого рода, ведь мышление - это то, что позволяет выжить человеку, в глобальном, широком значении этого слова. Рассматривая проблему ИИ в таком аспекте, приходим к тому, что **машина, в широком смысле, всегда была, есть и будет только средством достижения цели человека и ничем более того.** А в противном случае, может "думающая" машина сама станет человеком?

Затронем, еще один аспект проблемы ИИ, с точки зрения теории активного отражения. Опять воспользуемся аналогией паровоз - ИИ. Проблема создания паровоза гораздо проще, чем проблема создания, моделирования "искусственного машиниста", а тем более, "искусственного создателя паровоза". Всему свое время! Может быть, когда-нибудь и потребуются создать "искусственного машиниста". Скорее всего, это потребуются для некоторых специальных, чисто исследовательских целей. А иначе, если будет создан "машинист", который будет управлять паровозом без вмешательства человека, то, что останется на долю самого человека? Возможны варианты. Например, осознать свою ненужность, непричастность к дальнейшему продвижению по пути познания Истины и ... выброситься на полном ходу из поезда. Или, можно сломать "искусственного машиниста" и опять самому управлять паровозом. Или заняться разработкой и усовершенствованием своего "транспортного средства", "искусственного машиниста". Проблем-то остается еще очень и очень много.

Даже, если искусственный интеллект, как цель работ, подобен "вечному двигателю" или "философскому камню", то, все равно надо к нему идти, стремиться, двигаться. Ибо, по пути, мы узнаем много нового, полезного и интересного. Не беда, если при этом мы так и не создадим идеальный, универсальный, абсолютный ИИ. Тем более, выясняется, что он не очень то и нужен, гораздо

важнее, сейчас, создание узкоспециализированных средств, искусственных интеллектов. На данном этапе развития этой области исследователи уже прошли ту стадию, когда считалось, что можно быстро создать ИИ и осталось уже не много, еще чуть-чуть и ... тайна будет раскрыта.

И хотя все еще не все отказались от этой мысли, как от первостепенной, первоочередной, но ... мы еще пока сами не сели в кабину паровоза. А продолжаем "катать паровоз с Алгоритмических гор". Проблема "Универсального Мыслителя" откладывается до тех пор, пока не наберемся сил, пока не поймем того, что же мы все-таки от него хотим и как он должен выглядеть, работать, мыслить. Сейчас необходимо решать неотложные практические задачи и теоретически разобраться с проблемой активности, тайной порождения, возникновения знака, информации.

Необходимо создать, разработать теорию активного отражения, что позволит нам осознать все сложности и трудности проблемы моделирования мышления, а также позволит наметить "вехи" на пути создания интеллектуальных систем, в самом изначальном смысле этого слова.

Не надо пытаться уже сейчас создавать искусственного мыслителя, искусственного человека (клона). Людей и так достаточно на Земле. Зачем создавать еще одного? Тем более, что в настоящее время, к сожалению, мы просто не в состоянии сделать это, более того, мы не знаем как, в какой последовательности, это делать. Мы просто не готовы к решению этой проблемы, мы не в силах сделать этого. Мы даже не осознаем всех проблем, которые возникают при такой глобальной постановке вопроса. Мы в самом начале долгого и трудного пути, а "ношу", груз надо выбирать себе по силам.

Впрочем, биологи уже совершенно серьезно говорят о **клонировании человека**, т.е. дословно - о создании искусственного человека, искусственного разума. Ведь биологи утверждают, что теоретически все проблемы уже решены. **Клонирование человека - это решение проблемы искусственного интеллекта?**

Надеемся, что осознание ИИ как теории активного отражения, приблизит нас к цели, позволит взяться за "ношу знаний" потяжелее и идти к цели, хоть чуть-чуть, быстрее. А пока Искусственный Интеллект - это Горизонт и никто не может обоснованно сказать - достигнем ли мы его или нет. Надеемся, что полученные нами научные результаты и дальнейшая разработка теории активного отражения поможет нам ответить на интересующие нас вопросы.

## **8.8. ПСИХОЛОГИЯ И ЛОГИКО-ИНФОРМАЦИОННЫЙ ПОДХОД К МОДЕЛИРОВАНИЮ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ**

Прежде всего, рассмотрим взаимосвязь психологии и естественных наук (математики, кибернетики, информатики и т.д.) в области создания интеллектуальных систем. В информатике основные вопросы, связанные с изучением мышления человека, относят, как было написано выше, к проблеме создания систем искусственного интеллекта - ИИ. Актуальность этой проблемы обусловлена "бурным" развитием техники, внедрением новых информационных технологий, стремительным ростом технических возможностей компьютерных систем. Многие ученые в различных областях науки проводят исследования по математическому, кибернетическому описанию процессов мышления, т.е. обработки информации в голове человека. Это не удивительно, так как в настоящее время мы имеем только один вариант мышления - интеллект, т.е. мышление живого человека.

Роль психологии в исследованиях проблемы создания искусственного интеллекта трудно переоценить. Математики, кибернетики, "информационщики" с огромным интересом и вниманием изучают научные результаты, полученные психологами. Прежде всего, интерес для информатики представляют результаты исследования мышления человека с целью их моделирования и воспроизведения на компьютерных системах. Кроме того, активно исследуются проблемы автоматизации деятельности человека и психологические особенности общения человека с компьютером, хотя многие специалисты считают, что реально общается только пользователь с программистом "через компьютер".

Отметим, что некоторые результаты информатики и кибернетики могут быть использованы и при проведении психологических исследований. Особую роль играет в психологии математика. Кроме непосредственного использования математических методов для обработки результатов экспериментов, некоторые идеи математики могут быть использованы, например, непосредственно для развития новых концепций психологии.

Например, если говорить о динамике самоотношения (более подробно рассмотренного выше в первом разделе) в условиях успеха и неуспеха, то могут быть использованы математические градации понятия успех. Причем, эти градации могут быть как дискретными, так и непрерывными. Можно ввести некоторые числовые характеристики "успеха", а вместо "неуспеха" - ввести понятие "отрицательного успеха". Таких примеров можно привести достаточно много, но представляется возможным сделать обоснованный вывод о том, что математика (информатика) является не только "служанкой" психологии при проведении экспериментов, но и в некоторых случаях "равнозначным партнером", с которым требуется считаться.

С другой стороны, термины психологии играют все более важную роль при исследовании проблемы создания систем искусственного интеллекта. Например, как было указано выше, профессором Шевченко А.И. предложено следующее определение ИИ: искусственный интеллект - это алгоритм решения задач, сформированных искусственным сознанием. При этом, под искусственным сознанием понимается высшая управляющая система машины, владеющая знаниями о себе и своем окружении [Л. 47]. Такой подход разделяют не все математики, но показательно, что роль и влияние психологии в области исследования искусственного интеллекта возрастает.

Таким образом, в области создания интеллектуальных систем психология и естественные науки, прежде всего - информатика, взаимосвязаны и могут конструктивно взаимодействовать.

Далее, представляется интересным кратко ("конспективно") рассмотреть логико-информационные аспекты моделирования интеллектуальных систем, прежде всего - интеллекта. Сначала проанализируем аспекты логической обработки информации, т.е. логико-информационный подход к моделированию интеллектуальных систем. Как известно, в настоящее время, в психологии существуют различные взгляды на моделирование интеллекта. В то же время, как было сказано выше, моделированием интеллектуальных процессов активно занимаются математики ("кибернетики и информатики").

Напомним, что основными свойствами интеллекта являются: отражение объективной действительности, преобразование опыта, выделение существенного и управление действиями самой системы. При этом, как известно, интеллект есть высшая стадия развития психики животных [Л. 106, 153, 223, 259, 265, 272, 273, 291, 427-459].

Следовательно, целесообразно ввести некоторую **шкалу измерения интеллекта**. Отметим, что определенно есть некоторая взаимосвязь между интеллектом и рассмотренной выше теорией активного отражения. Возможно, что активное отражение это нечто более общее (в смысле обобщения, описания и универсальности), чем интеллект. С другой стороны, одной из высших стадий активного отражения, в изложенной выше трактовке, определенно является интеллект и мышление человека. Подчеркнем, что возможно еще более высшей стадией развития интеллекта будет некий человеко-машинный интеллект, т.е. человеческие способности будут многократно усилены за счет автоматизации некоторых функций обработки информации путем создания интеллектуальных систем (автоматизированных систем сбора и обработки информации).

При таком подходе, интересной проблемой являются отрицательные и нулевое значение степени интеллектуальности. Возможно, живое отличается от неживого, именно наличием некоторой "интеллектуальности" (или в более общем смысле - активностью). Вообще, соотношение и взаимосвязь активности и интеллектуальности требует более подробного изучения и является одним из важнейших направлений дальнейших исследований [Л. 106,153,223,259,265, 272, 273, 291, 427-459].

Другой интересной проблемой является специализация интеллекта, особенности мышления. Помимо традиционных тестов необходимо по аналогии с компьютерными системами ввести различные области - специализацию интеллекта. Заметим, что, например, интеллекты футболиста, физика, музыканта и т.д. различны, следовательно, для их корректного сравнения (ведь гении есть везде, в каждой области) необходимо ввести специализацию интеллектуальной деятельности. Кроме того, возможно различные способности, специализация мышления необходима человечеству для дальнейшей эволюции, поддержания "на всякий случай" интеллектуального генофонда человечества. Ведь на данном этапе (когда информационные технологии и возможности развиваются столь стремительно) никто не может предсказать, с чем человечеству придется столкнуться даже в ближайшем будущем, а тем более как люди будут взаимодействовать с автоматизированными и интеллектуальными системами.

Кроме того, при моделировании интеллектуальной системы, необходимо понимать, что модели любых подсистем (зрение, движение и т.п.) не могут выявить свойств самой системы, что было показано выше. Следовательно, целесообразно использовать структурно-системный подход к моделированию интеллекта.

Одним из основных свойств интеллекта является обработка информации. При этом, сама информация, как было показано выше, характеризуется следующим. Единицей измерения информации является "знак", для существования которого необходим материальный, объективный носитель. Но процесс понимания и порождения информации происходит только у некоторого субъекта, следовательно, каждому знаку присущи и субъективные, идеальные свойства. В общем случае, обработка информации, понимание знака - это некоторый процесс взаимодействия объективного носителя и субъекта, способного воспринять субъективные характеристики этой информации. В таком случае, процессы обработки информации могут быть проинтерпретированы как знаковое моделирование некоторой предметной области или окружающей среды.

Возможно, что в психологии целесообразно использовать следующий подход при анализе логических функций человека. Под логической обработкой в математике принято понимать выделение некоторых переменных и правил их взаимодействия. Затем происходит построение некоторой системы логического вывода от известных переменных через правила к требуемым (неизвестным) пе-

ременным. Как было доказано выше, эта система правил может быть представлена в виде миварного многомерного пространства обработки информации. В принципе, если задано достаточное количество входных переменных, то всегда должно быть получено правильное решение. Основная сложность заключается в том, что для нахождения правильного ответа необходимо фактически перебрать (затратив много времени) и выполнить все возможные правила и найти некоторый, возможно не единственный, маршрут логического вывода.

Введение многомерного пространства - это известный способ радикального ускорения обработки информации. Предположим, что в интеллектуальной системе (или модели интеллекта, мышления) можно построить некоторый многомерный граф взаимосвязей переменных и правил. Тогда, при необходимости логической обработки выделяют все известные и требуемые переменные, а затем осуществляют построение некоторого графа взаимосвязи известных и требуемых переменных.

Приведем следующую аналогию: пусть логическая система представлена как некоторые фрагменты "рыболовных сетей", сваленных в одну кучу. Изначально нам даны только несколько известных и требуемых "узлов" этой сети. Для определения наличия взаимосвязи будем как бы вытягивать из этой кучи в одну сторону известные "узлы", а в другую - требуемые. Если все эти узлы принадлежат одному целому фрагменту сети, то путем растягивания мы можем определить не только наличие факта взаимосвязи, но и выявить конкретные кратчайшие пути от известных к требуемым переменным. Выявив эти пути, в дальнейшем выполняем только те правила, которые требуются для логического вывода. Если при "растягивании сети" мы вытаскиваем разные фрагменты, значит, маршрута логического вывода нет и необходимо ввести новые данные, которые позволят "завязать" новые, дополнительные взаимосвязи между фрагментами сети. Эта достаточно "грубая" аналогия позволяет промоделировать процесс поиска маршрута (пути) логического вывода у человека и в других интеллектуальных системах.

Данный подход к моделированию позволяет объяснить различные способности интеллектуальных систем и приводит к некоторым "парадоксальным" выводам. Например, если рассматривать всю интеллектуальную систему как единую систему обработки и хранения информации, то получаем: чем больше информации хранится в системе, тем меньше ресурсов отводится для обработки. Конечно, существуют некоторые начальные условия по объему хранимой информации, но получается, что от обычной "студенческой зубрежки" в некотором смысле больше вреда, чем пользы. Более перспективным является путь выделения некоторого оптимального (критерии оптимальности - отдельная тема) количества необходимой информации и построение более мощной системы логического вывода в интеллектуальной системе с использованием "внешних запоминающих устройств" - книг, справочников, баз данных и т.п. Возможно, что у гениальных ученых просто более многомерное системное мышление, которое позволяет им на длительных интервалах времени проводить "в голове" построение сложной логической системы и поиска на ней маршрута логического вывода.

Кроме того, наш мозг одновременно управляет всем организмом, обрабатывает "входную" информацию и, следовательно, все лишние "раздражения" также отнимают "ресурсы обработки информации", не дают "сосредоточиться". Возможно, в этом и есть секрет "озарения", "инсайта" - когда человек долго думает, выстраивает сеть и внезапно понимает, что решение существует и "видит" путь этого решения. Внешние воздействия (шум) мешают "построению сети" и "сбивают нас с мысли". Эти предположения, возможно, не являются однозначными, но имеют право на дальнейшее исследование и обсуждение.

Итак, приходим к следующим выводам. Логико-информационный подход позволяет исследовать некоторые новые аспекты моделирования интеллектуальных систем. Построенный на основе этого подхода строго формализованный активный адаптивный механизм логического вывода на эволюционной сети правил, управляемой потоком данных, открывает новые возможности в различных науках по исследованию интеллектуальных систем и моделированию интеллекта.

## 8.9. ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ ЭВОЛЮЦИОННЫХ БАЗ ДАННЫХ И ПРАВИЛ ДЛЯ АДАПТИВНОГО СИНТЕЗА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Таким образом, для достижения цели создания эволюционных баз данных и правил в интересах синтеза интеллектуальных систем в данной монографии были поставлены и решены пять следующих **научных задач**.

1. Разработаны теоретические основы адаптивного синтеза конфигураций компьютерных систем. Прежде всего, отметим, что разработаны принципы построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций, а также созданы теоретические основы эволюционных, адаптивных баз данных и правил путем построения миварного эволюционного пространства унифицированного представления данных и правил. К основным принципам построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций могут быть отнесены: адаптивность, модульность, непрерывность, возможность глобального накопления и обработки в едином унифицированном формате любых данных и правил, максимальная оперативность обработки, активное формирование требований по изменению своих ресурсов и конфигурации, независимость от конкретных технических средств и открытость системы.

Для создания эволюционных, адаптивных баз данных и правил было осуществлено обобщение всех традиционных структур представления данных в виде пятиуровневой одномерной таблицы представления данных (ОТПД-5). Затем, были разработаны эволюционные динамические многомерные структуры представления данных, в которых могут изменяться не только значения переменных, но и количество осей пространства, т.е. сама структура представления данных. Таким образом, предложено рассмотреть новый класс моделей данных с адаптивной, эволюционной структурой. Напомним, что эволюционные динамические многомерные структуры унифицированного представления данных и правил для краткости и однозначности обозначения названы "миварными" структурами. Суть **миварного представления** данных и правил в следующем. Формируется минимальное пространство представления данных путем определения основных осей и фиксации основных объектов и их отношений. Названия объектов и отношений в некотором порядке фиксируются на осях, а на соответствующих пересечениям точках этого пространства, которые называются миварами, записываются конкретные значения свойств и отношений объектов. В дальнейшем, при появлении новых данных, они накапливаются при неизменной структуре, или происходит изменение структуры путем добавления, изменения или удаления любого отдельного мивара, любой точки на любой оси пространства или даже самой оси.

Важной особенностью миварного подхода является то, что отношения объектов хранятся в том же едином миварном пространстве и могут представлять собой правила или процедуры обработки. Отметим, что любое отношение также

может быть добавлено, удалено или изменено в любое время. Таким образом, именно миварное пространство является фундаментом теоретических основ создания систем адаптивного синтеза информационно-вычислительных конфигураций и обеспечивает реальную эволюционность синтеза конфигураций многопроцессорных вычислительных систем.

2. Разработан метод обработки данных на основе применения логической сети правил, управляемой потоком данных. Так как миварный подход существенно расширяет возможности обработки информации в АСОИ, то разработан новый метод обработки данных. Суть данного метода в том, что в миварном пространстве из всех правил и объектов формируется логическая сеть вывода (обработки), управляемая потоком данных. В зависимости от имеющихся значений переменных и правил, подграфы обработки-вывода могут динамически изменяться (рис. 8.9.1). При этом возможно формирование новых правил, существенно сокращающих процесс вывода.

Отметим, что адаптивная активная логическая сеть, управляемая потоком данных, объединяет и обработку известных данных на основе известных правил, и одновременно осуществляет обучение, изменение системы правил со своевременным внесением изменений в соответствующие процессы, что обеспечивает эволюционное развитие всей компьютерной системы в целом. Кроме того, такая логическая сеть позволяет максимально распараллелить обработку потока данных.

3. Разработан метод быстрого поиска маршрута логического вывода на основе построения многополюсной сети и поиска ее минимального разреза. Для повышения оперативности обработки данных в системе адаптивного синтеза информационно-вычислительных конфигураций был разработан новый метод быстрого поиска маршрута вывода. Суть этого метода в преобразовании логической сети правил в многополюсную сеть теории графов и поиска ее минимального разреза. Для этого строится граф обработки, с одной стороны от исходных, известных значений, а с другой стороны - от требуемых, искомых. Затем осуществляется анализ связности этого графа путем поиска его минимального разреза. Если минимальный разрез больше или равен 1, то решение существует и определяется кратчайший, минимальный путь достижения требуемых результатов. На основе полученного пути непосредственно запускается механизм вывода, что позволит значительно экономить вычислительные ресурсы. Если минимальный разрез равен 0, то цепочки вывода нет, но можно определить значения каких именно переменных являются критическими для уточняющего запроса пользователю, что и является признаком "активности" обработки данных.

Кроме того, нами был разработан квадратичный метод поиска минимального разреза многополюсных сетей. Внедрение этих методов позволит активно формулировать уточняющие запросы и снизить вычислительную сложность поиска маршрута логического вывода от NP-полной до квадратичной, а в некоторых случаях до линейной, что существенно повысит оперативность функционирования АССОИ.

4. Разработан метод максимального распараллеливания множественного доступа к общей базе данных путем построения виртуальных потоковых баз данных. Как было отмечено выше, оценка возможностей известных методов корректного распараллеливания множественного доступа к общей базе данных показала их низкую эффективность, поэтому был разработан новый метод. Допустим, что элементы входного потока поступают на обработку в различные моменты времени, тогда, порождаемые ими процессы обработки, могут быть разделены на старших (поступивших на обработку ранее) и младших.

Напомним, что суть данного метода в том, что самый старший процесс работает непосредственно с реальной БД, а для всех остальных действующих процессов создается виртуальная потоковая база данных - ВПБД, в которой для каждого процесса создается и ведется соответствующая персональная база. Все изменения, которые процесс должен внести в БД фиксируются в его персональной базе. Младшие процессы получают доступ по чтению последовательно через персональные базы старших процессов, а все изменения вносят только в "свои" персональные базы. Отметим, что ВПБД фактически является надстройкой над реальной БД. Таким образом, получаем, что все изменения вносятся в общую БД, но младшие процессы не оказывают никакого влияния на одновременно обрабатываемые с ними старшие процессы. Следовательно, из всех известных способов обработки баз данных только потоковая параллельная обработка обеспечивает корректное максимальное распараллеливание множественного доступа к общей базе данных.

5. Разработан линейный метод единично-инкрементного суммирования чисел для решения некоторых классов специальных задач. Как было отмечено выше, анализ основных факторов, влияющих на процесс обработки информации, выявил возможность повышения быстродействия многопроцессорных вычислительных систем при использовании запатентованного способа быстрого суммирования чисел, который и позволил разработать линейный метод переборного суммирования. Напомним, что суть этого метода состоит в алгоритмической минимизации количества операций сложения при переборном (единично-инкрементном) суммировании чисел. Кроме того, дополнительно были разработаны две модификации этого метода для его внедрения в различных условиях и на различных вычислительных средствах. Внедрение этого метода и двух его модификаций позволит перейти от степенной к линейной зависимости общего количества операций сложения от разрядности суммируемых чисел, что существенно повысит оперативность решения некоторых классов специальных задач.

Для решения всех пяти задач, изложенных в данной работе, были применены методы теорий множеств, графов, структур данных, принятия решений, математической логики, информатики, системного анализа, реляционной алгебры математического программирования и методы оптимизации на сетях и графах.

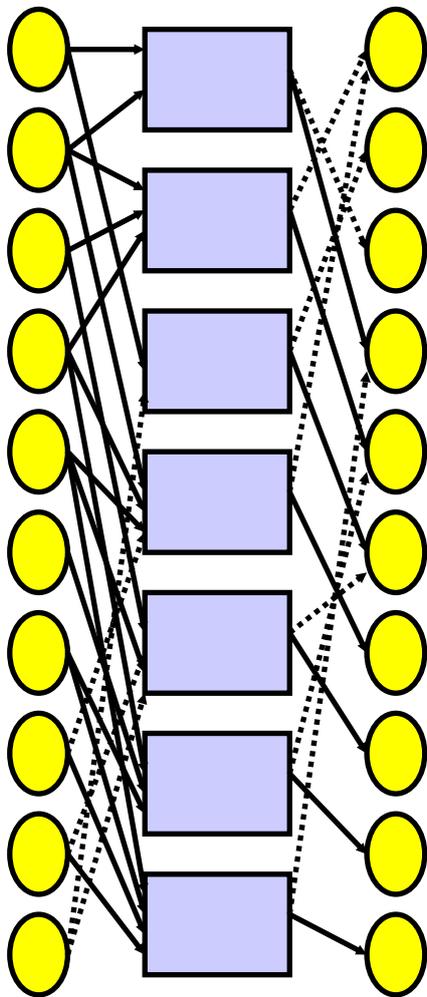
Результаты научных исследований, изложенных в данной книге, неоднократно обсуждались на различных научных конференциях и семинарах, печатались в различной научной литературе [Л. 41-46, 427-430], а метод линейного единично-инкрементного суммирования был запатентован [Л. 225].

Таким образом, в данной книге сформулирована и решена актуальная, сложная научная проблема создания эволюционных баз данных и правил для адаптивного синтеза интеллектуальных систем, которая имеет важное практическое значение. Перспективы данного научного направления заключаются в том, что в ходе дальнейших исследований на следующих этапах предполагается:

- 1) провести математическое и вычислительное моделирование конфигураций МВС,
- 2) провести математическое моделирование различных классов сложных и специальных задач, которые могут быть решены с использованием САС ИВК,
- 3) исследовать возможные направления и проблемы реализации эволюционных БДП,
- 4) исследовать возможные направления и проблемы реализации САС ИВК,
- 5) провести анализ и оценку функционирования САС ИВК,
- 6) исследовать возможности САС ИВК для создания интеллектуальных систем,
- 7) разработать теорию активного отражения и исследовать ее возможности.

ОБЩЕЕ  
ПРЕДСТАВЛЕНИЕ  
ЛОГИЧЕСКОЙ СЕТИ

ПРАВИЛ



 - ОБЪЕКТЫ

 - ПРАВИЛА

МАРШРУТ ЛОГИЧЕСКОГО  
ВЫВОДА

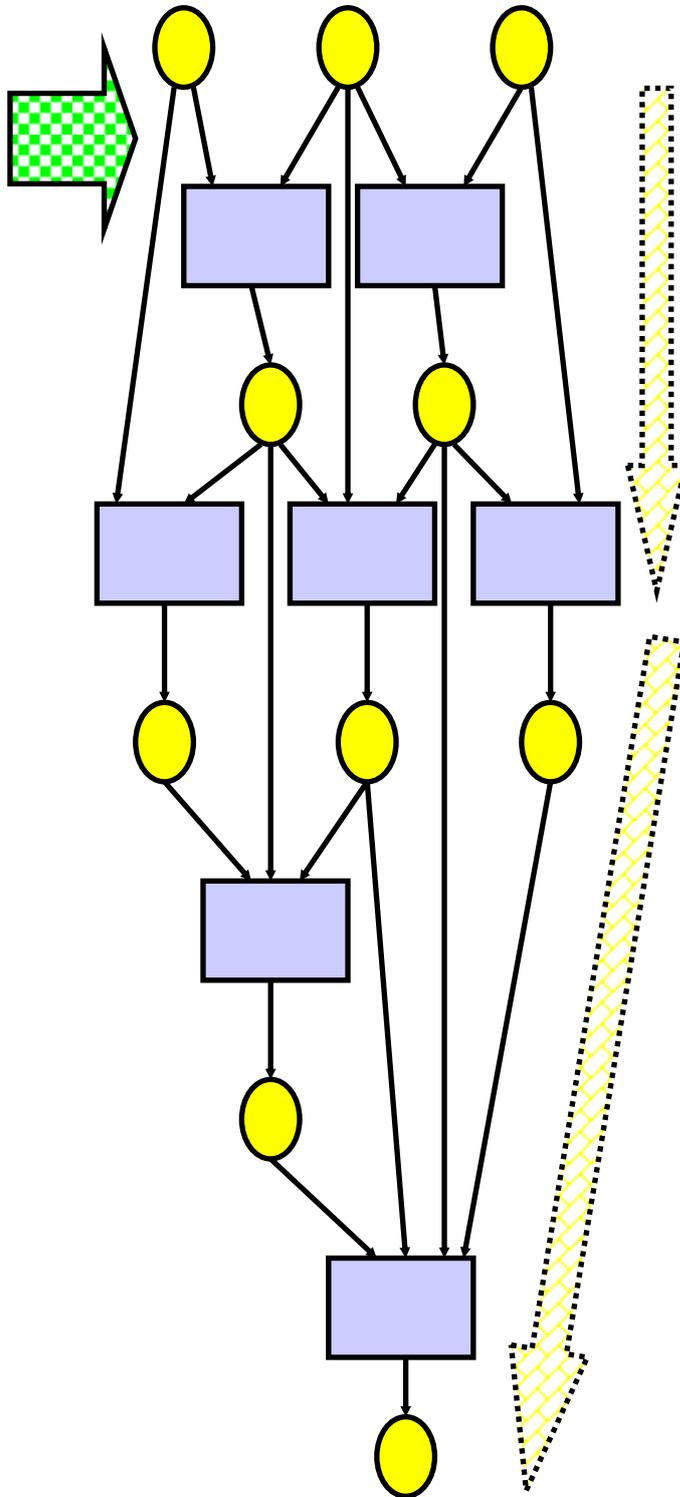


Рис. 8.9.1. Миварная логическая сеть правил.

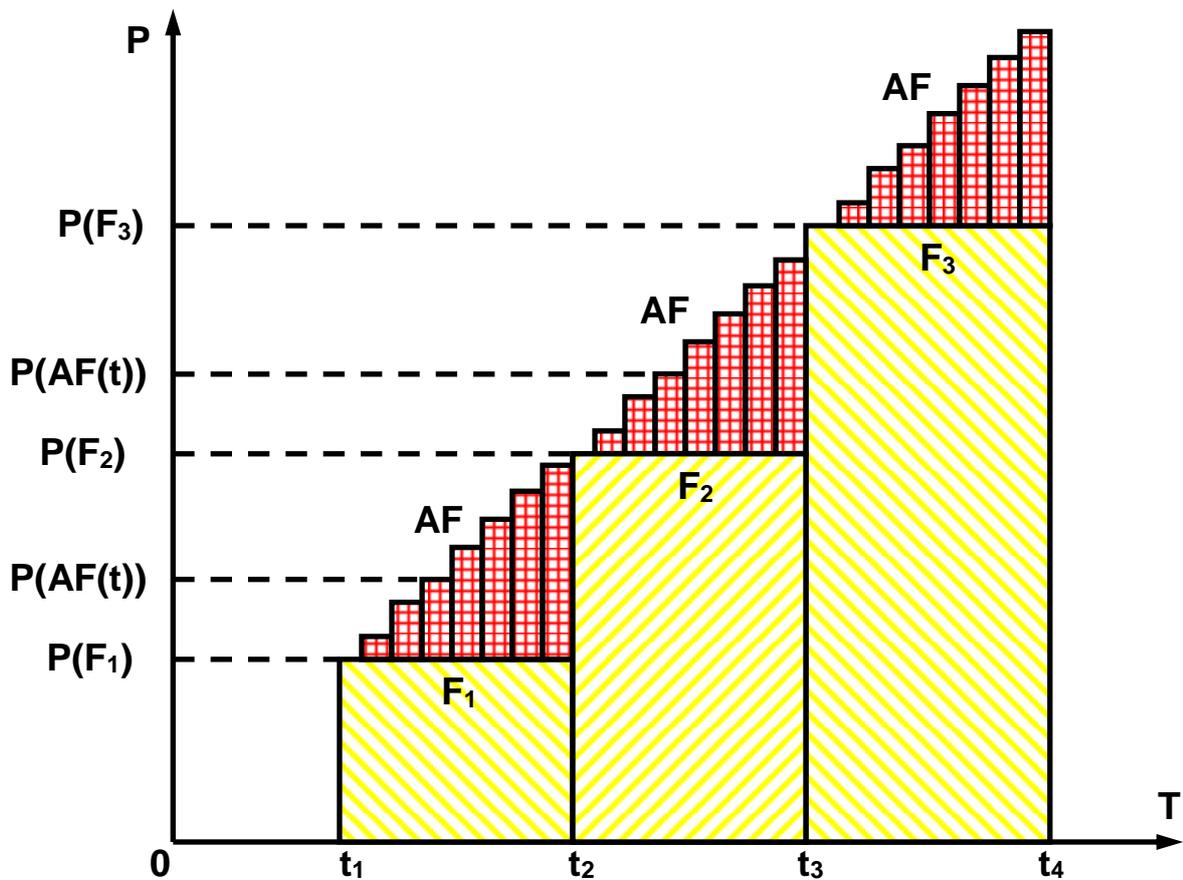
**Практическая ценность** данной монографии обусловлена тем, что внедрение полученных новых научных результатов позволит:

1) Обеспечить возможность создания эволюционных интеллектуальных многопроцессорных систем и комплексов.

2) Уменьшить время решения сложных задач:

а) за счет общего повышения производительности адаптивных ЭВМ относительно традиционных (рис. 8.9.2), по оценкам специалистов, в среднем за период эксплуатации на половину разности производительности следующих и современных поколений ЭВМ - сейчас на 1,5 порядка (в 500 раз);

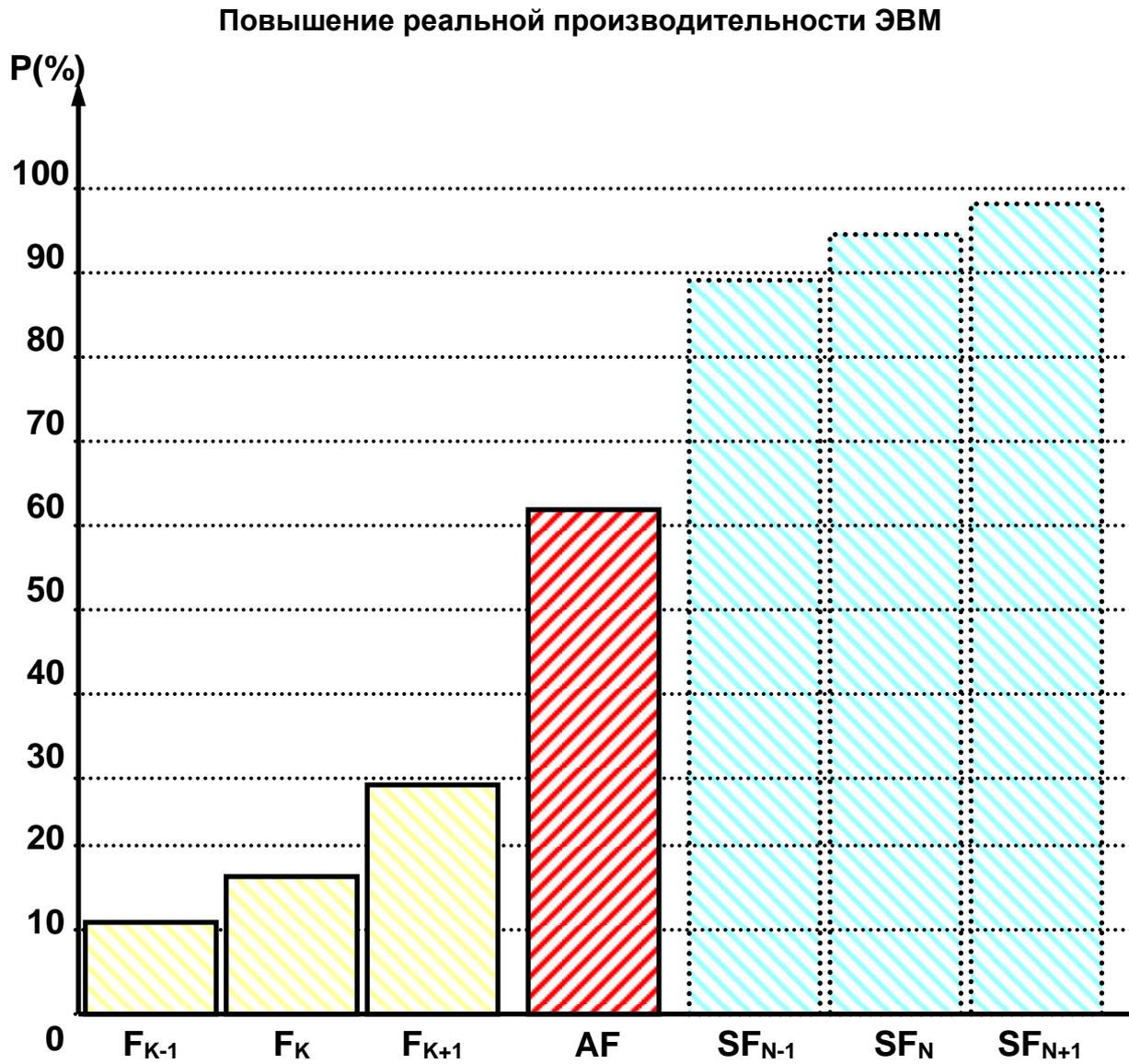
### ПРОИЗВОДИТЕЛЬНОСТЬ АДАПТИВНЫХ ЭВМ



$$\forall t P(AF(t)) \geq P(F_j); P_{CP}(AF(t_j)) \approx P(F_j) + 0,5 \cdot [P(F_{j+1}) - P(F_j)].$$

Рис. 8.9.2. Повышение производительности адаптивных ЭВМ.

- b) М путем повышения реальной производительности адаптивных ЭВМ вдвое (рис. 8.9.3), т.е. относительно пиковой мощности с 10-30 до 50-70 %.

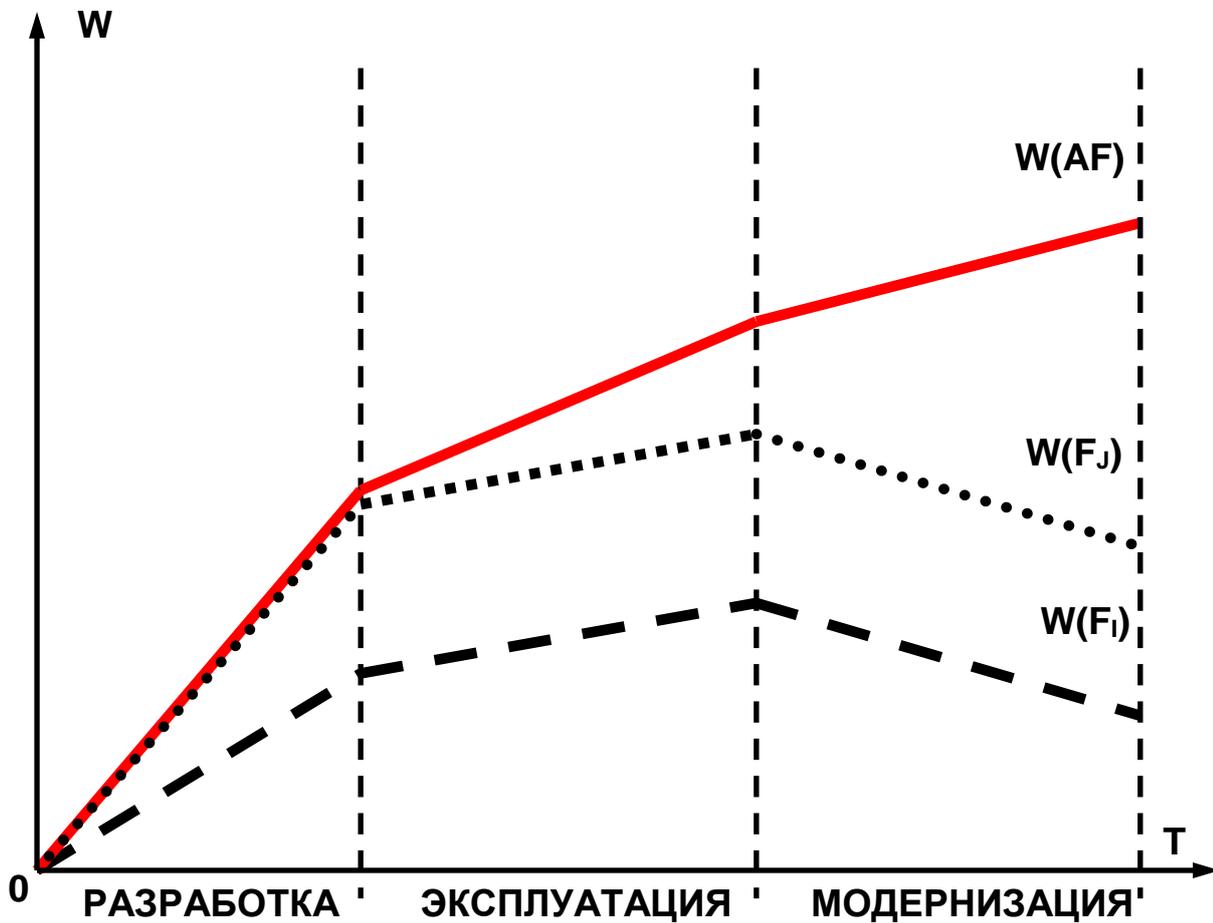


**ГДЕ:  $F_K$ - УНИВЕРСАЛЬНЫЕ ЭВМ, **AF** - АДАПТИВНЫЕ ЭВМ,  
 $SF_N$  - СПЕЦИАЛИЗИРОВАННЫЕ ЭВМ.**

Рис. 8.9.3. Повышение реальной производительности адаптивных ЭВМ.

3) Снизить материально-финансовые затраты и повысить эффективность разработки, эксплуатации и модернизации компьютерных систем (МВС, ВВК, суперЭВМ) за счет оптимизации процесса синтеза конфигураций по критерию отношения реальной производительности к стоимости, а также путем постоянного наращивания вычислительной мощности на основе адаптации и добавления только наиболее современных, на каждом этапе существования ЭВМ, компьютерных модулей (рис. 8.9.4).

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ АДАПТИВНЫХ ЭВМ



**ГДЕ:  $W$  - ЭФФЕКТИВНОСТЬ ПО КРИТЕРИЮ ОТНОШЕНИЯ РЕАЛЬНОЙ ПРОИЗВОДИТЕЛЬНОСТИ К СТОИМОСТИ,  $AF$ - АДАПТИВНЫЕ ЭВМ,  $F_j$  и  $F_i$  - ТРАДИЦИОННЫЕ ЭВМ.**

Рис. 8.9.4. Повышение эффективности адаптивных ЭВМ.

Отметим, что кроме выше указанного, миварный подход позволяет адаптивно и своевременно (оперативно) наращивать объем хранимой информации и обрабатывать ее с применением активной адаптивной логической сети правил (гиперправил с мультиактивизаторами), управляемой потоком данных, что создает предпосылки создания "интеллектуальных" (синергетических, самообучаемых или обучаемых) компьютерных систем (в смысле: программно-аппаратных комплексов) с изменяемой, адаптивной конфигурацией на основе эволюционных неоднородных многопроцессорных вычислительных систем и ЭНКС.

Подчеркнем, что "активная логическая сеть обработки, управляемая потоком данных" объединяет и обработку известных данных на основе известных правил, и одновременно осуществляет обучение, изменение системы правил со своевременным внесением изменений в соответствующие процессы обработки, что обеспечивает эволюционное развитие всей компьютерной системы в целом.

Кроме того, такая активная логическая сеть позволяет максимально распараллелить и ускорить обработку любого потока данных, а также повысить эффективность использования технических средств.

Некоторые научные результаты, представленные в этой работе могут использоваться независимо друг от друга, как имеющие самостоятельное научное значение, например: метод поиска маршрута логического вывода, метод распараллеливания доступа к общей базе данных или метод линейного единично-инкрементного суммирования чисел.

Таким образом, эти и другие, представленные в данной работе, **научные результаты** могут использоваться как по отдельности (самостоятельно, в различных областях науки и техники), так и вместе, позволяя в совокупности:

**решить проблему создания эволюционных, адаптивных баз данных и правил (знаний) для синтеза интеллектуальных систем.**

Решение предыдущей проблемы позволит также решить проблему построения **систем адаптивного синтеза информационно-вычислительных конфигураций**, которые будут способны изменять не только свои программные средства и их конфигурации, но, в перспективе, и наращивать, модернизировать и аппаратные средства информационно-вычислительных комплексов.

Возможно, что при внедрении и эксплуатации предложенных нами методов и теоретических основ возникнут новые задачи и проблемы, которые потребуют дополнительного исследования. Конечно, по многим из указанных выше проблем и задач возможно проведение дополнительных исследований, научных разработок, но, в настоящее время, на данном этапе мы считаем эту работу (на определенном этапе) законченной, пути дальнейшего развития определены.

Оценивать предложенные теоретические основы и методы, а также полученные результаты необходимо на практике. Впрочем, это уже тема будущих исследований и новых научных работ, которые (будем оптимистами!) обязательно будут, ведь развитие и технический прогресс не остановить! Так что, до следующих встреч! Всем - удачи, терпения и творческих успехов!

## ЗАКЛЮЧЕНИЕ

Монография посвящена разработке теоретических и реализационных основ создания эволюционных баз данных и знаний и систем адаптивного синтеза информационно-вычислительных конфигураций для построения АСОИ и интеллектуальных систем.

**Актуальность темы.** Одной из основных проблем, решаемых при создании АСОИ и интеллектуальных систем, является обеспечение, в условиях их непрерывного функционирования, адаптации программно-аппаратных средств для эффективного и оперативного решения сложных задач. Анализ актуальных научных проблем создания интеллектуальных систем показал, что необходимо, прежде всего, решить следующие две взаимосвязанные проблемы. Первая проблема - это создание эволюционных баз данных и знаний, на основе которых возможно создание программного обеспечения интеллектуальных систем. Отметим, что, так как фактически в "базах знаний" хранятся только правила, процедуры и другие отношения объектов, то вместо термина "базы знаний", в этом же смысле, целесообразно применять термин "базы правил", т.е. они являются синонимами, но второй термин все же является более адекватным. Подчеркнем, что особую актуальность решению первой проблемы придает то, что в последнее время теория баз данных и знаний оказывает определяющее воздействие на многие смежные области. Например, базы данных и правил (знаний) используются при создании перспективных ЭВМ, что определяет место и роль другой научной проблемы. Вторая проблема - это создание теоретических основ адаптивного синтеза ИВК интеллектуальных систем на базе известных методов синтеза ЭВМ. Основное направление решения данной проблемы, это именно создание адаптивных программно-аппаратных конфигураций интеллектуальных систем на основе применения известных методов синтеза ЭВМ. Две эти проблемы решают на разных уровнях: на уровне программного обеспечения (первая) и на уровне аппаратных средств (вторая), фактически одну и ту же проблему - создание активных эволюционных АСОИ, которые, в свою очередь, являются основой для создания интеллектуальных систем. Анализ задач, которые необходимо решить как для создания ЭБДП, так и для построения САС ИВК, показал их взаимозависимость, поэтому обе эти проблемы в совокупности образуют одну крупную научную проблему. Следовательно, разработка теоретических и реализационных основ создания ЭБДП и САС ИВК для построения АСОИ, предназначенных для оперативного решения сложных задач, является актуальной крупной научной проблемой.

Решение этой новой проблемы вносит значительный вклад в развитие экономики страны, способствует повышению обороноспособности России и имеет важное хозяйственное значение и практическую ценность для целого ряда областей (медицины, экономики, анализа чрезвычайных ситуаций, метеорологии и других). Отдельные наиболее сложные задачи этих областей могут быть отнесены к классу ресурсоемких научно-практических задач оперативной диагностики. Как правило, это уникальные задачи, решение которых носит эмпирический характер и требует научно-обоснованного оперативного синтеза специализированных ИВК. Известные технологии баз данных и знаний, а также систем автоматизированного проектирования (САПР) не применимы в исследуемом случае, так как основное противоречие состоит в том, что в условиях дефицита времени требуется оперативно синтезировать уникальный эволюционный программно-аппаратный комплекс (ПАК) с использованием всех доступных ресурсов, включая Интернет, для обеспечения экспресс-диагностики сложных уникальных задач. Как правило, это NP-полные зада-

чи, которые в зависимости от конкретной ситуации, могут быть сведены к набору полиномиальных задач, решаемых за обозримое допустимое время с учетом конкретных ограничений. На программном уровне эволюционность такого комплекса обеспечивается ЭБДП, а на аппаратном уровне - САС ИВК. Адаптивность синтеза ИВК обусловлена уникальностью каждой диагностической задачи, необходимостью учета ранее разработанных вариантов конфигураций, быстротой создания, развития и старения программных и аппаратных средств. Для синтеза требуемых конфигураций фактически необходимо разработать некоторую новую САПР на основе распределенных ЭБДП. Исходные данные могут быть в различных форматах представления данных и знаний, но их необходимо хранить в едином структурированном унифицированном пространстве представления данных и правил с целью научно-обоснованного всеобъемлющего синтеза уникальных конфигураций ЭВМ и последующего оперативного решения задачи. Подчеркнем, что САС ИВК сама по себе может являться ядром формируемого уникального программно-аппаратного комплекса оперативной диагностики.

Таким образом, актуальность крупной научной проблемы разработки эволюционных баз данных и правил для создания систем адаптивного синтеза конфигураций АСОИ обусловлена тем, что необходимо в минимальное время достижение максимального быстродействия для решения уникальной диагностической задачи в ситуации, когда нахождение правильного решения имеет жизненно-важное значение, цена которого априори во много раз превышает стоимость затрат на синтез любой компьютерной системы. Следовательно, известные методы синтеза ИВК, основанные на коммерческой эффективности, в таких ситуациях не применимы.

**Научная проблема**, решаемая в монографии, относится к управлению в технических системах, в частности, к области обработки данных, адаптации, обучения и синтеза компьютерных систем, а также к созданию систем искусственного интеллекта. Проблему можно сформулировать следующим образом: разработка теоретических основ создания активных эволюционных баз данных и правил, адаптивного синтеза информационно-вычислительных конфигураций АСОИ (интеллектуальных систем) и быстродействующих методов обработки информации для оперативного решения сложных задач.

**Целью работы** является создание теоретических и реализационных основ построения эволюционного синергетического познавательного-диагностического программно-аппаратного комплекса активной аналитической оперативной обработки информации на основе разработки ЭБДП и САС ИВК.

В соответствии с поставленной целью определены задачи монографии:

- 1) в теории баз данных проанализировать, сравнить и обобщить основные структуры представления данных традиционных моделей данных;
- 2) разработать теоретические основы создания эволюционных баз данных и правил;
- 3) разработать теоретические основы адаптивного синтеза информационно-вычислительных конфигураций путем создания принципов построения и применения САС ИВК;
- 4) создать метод обработки данных на основе применения активной адаптивной многоярусной многотиповой логической сети правил, управляемой потоком данных;
- 5) разработать быстродействующий метод "графового" поиска маршрута логического вывода на основе построения многополюсной сети и поиска ее минимального разреза;

- 6) разработать метод максимального распараллеливания множественного доступа к общей базе данных в условиях недопущения взаимного искажения используемых данных;
- 7) разработать метод алгоритмической минимизации необходимого количества устройств и вычислительных процедур сложения.

**Научная новизна** монографии заключается в том, что в ней впервые:

- 1) разработаны теоретические и реализационные основы создания эволюционных баз данных и правил (знаний);
- 2) предложены принципы построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций;
- 3) осуществлено описание, анализ, сравнение и обобщение основных структур представления данных традиционных моделей данных на основе применения пятиуровневых одномерных таблиц представления данных;
- 4) построено динамическое многомерное дискретное информационное пространство унифицированного представления данных и правил;
- 5) предложен метод обработки данных на основе применения активной адаптивной логической сети правил, управляемой потоком данных;
- 6) созданы принципы "графового" поиска маршрута логического вывода;
- 7) разработан метод поиска маршрута вывода на логической сети правил, управляемой потоком данных, путем ее преобразования в многополюсную сеть теории графов и поиска ее минимального разреза;
- 8) предложен адаптивный механизм логического вывода на эволюционной сети гиперправил с мультиактивизаторами, управляемой потоком данных;
- 9) разработан линейной сложности матричный метод поиска маршрута логического вывода на сети правил;
- 10) в теории графов разработан квадратичной сложности метод поиска минимального разреза двухполюсных сетей;
- 11) в теории графов разработан квадратичной сложности метод поиска минимального разреза многополюсных сетей;
- 12) разработан метод максимального распараллеливания множественного доступа к общей базе данных в условиях недопущения взаимного искажения используемых данных;
- 13) разработан метод алгоритмической минимизации количества процедур сложения единично-инкрементного суммирования чисел;
- 14) исследованы перспективы применения локальных корректировок вычислений и обработки данных на основе адаптивных вычислений;
- 15) обоснованы перспективы применения эволюционных баз данных и правил для адаптивного синтеза интеллектуальных систем.

**Достоверность** полученных результатов обеспечивается использованием аппарата математической логики, теорий множеств, графов, структур данных, принятия решений, системного анализа, математического программирования и оптимизации на сетях и графах, полнотой и корректностью исходных предпосылок, математической строгостью доказанных утверждений и преобразований при получении аналитических зависимостей, а также результатами имитационного моделирования и практической реализации программ и БД.

**На обсуждение выносятся** следующие научные результаты:

- 1. Принципы построения и применения систем адаптивного синтеза информационно-вычислительных конфигураций.

2. Формализованное описание, анализ, сравнение и обобщение СПД традиционных моделей данных на основе пятиуровневых одномерных таблиц представления данных.
3. Теоретические основы построения динамического многомерного объектно-системного дискретного информационного пространства унифицированного представления данных и правил (знаний).
4. Метод обработки данных на основе применения активной адаптивной многоярусной многотиповой динамической логической сети правил, управляемой потоком данных.
5. Принципы "графового" поиска маршрута логического вывода.
6. Метод поиска маршрута вывода на логической сети правил путем определения минимального разреза многополюсной сети теории графов.
7. Адаптивный механизм логического вывода на эволюционной активной сети гиперправил с мультиактивизаторами, управляемой потоком данных.
8. Линейной сложности матричный метод поиска маршрута логического вывода на сети правил.
9. Квадратичной сложности метод поиска минимального разреза двухполюсных сетей.
10. Квадратичной сложности метод поиска минимального разреза многополюсных сетей.
11. Метод максимального распараллеливания множественного доступа к общей БД в условиях недопущения взаимного искажения используемых данных.
12. Метод алгоритмической минимизации количества процедур и устройств сложения для единично-инкрементного суммирования чисел.

**Практическая ценность работы** определяется следующими результатами:

- 1) предложенные основы создания ЭБДП и принципы построения и применения САС ИВК обеспечивают возможность создания логически обучаемых (синергетических, самообучаемых) АСОИ и интеллектуальных систем (программно-аппаратных комплексов) с изменяемой конфигурацией;
- 2) предложенные основы создания ЭБДП позволяют адаптивно и своевременно наращивать объем хранимой информации и обрабатывать ее с применением активной эволюционной логической сети правил (гиперправил с мультиактивизаторами), управляемой потоком данных;
- 3) разработанная активная эволюционная логическая сеть, управляемая потоком данных, объединяет обработку известных данных на основе известных правил и осуществляет обучение, изменение системы правил, что обеспечивает эволюционное развитие всей компьютерной системы в целом;
- 4) разработанные основы эволюционного информационного пространства, метод обработки данных на основе активной адаптивной логической сети и адаптивный механизм логического вывода на сети гиперправил с мультиактивизаторами повышают эволюционные и интеллектуальные способности АСОИ путем активного формирования запросов и ресурсов для развития в изменяющейся внешней среде;
- 5) предложенные принципы "графового" поиска маршрута логического вывода, метод поиска маршрута логического вывода и методы поиска минимального разреза позволяют снизить вычислительную

- сложность поиска маршрута логического вывода с NP-полной до квадратичной;
- 6) методы поиска минимального разреза (максимального потока) двухполюсных и многополюсных сетей позволяют снизить сложность поиска минимального разреза с кубической (2,5) до квадратичной;
  - 7) метод алгоритмической минимизации количества процедур и устройств сложения чисел позволяет перейти от степенной к линейной зависимости общего количества операций сложения от разрядности чисел;
  - 8) матричный метод поиска маршрута логического вывода позволяет снизить сложность поиска маршрута логического вывода с NP-полной до линейной;
  - 9) активная логическая сеть позволяет максимально распараллелить обработку любого потока данных и повысить эффективность использования ЭВМ;
  - 10) метод максимального распараллеливания множественного доступа к общей базе данных реализует максимальное быстродействие (без накопления времени задержки) обработки параллельных данных;
  - 11) предложенные основы и методы в совокупности позволяют повысить эффективность разработки, эксплуатации и модернизации компьютерных систем (систем обработки информации) за счет оптимизации синтеза конфигураций по критерию реальная производительность/стоимость;
  - 12) предложенные основы и методы в совокупности позволяют уменьшить время решения сложных задач за счет общего повышения производительности адаптивных ЭВМ относительно традиционных, в среднем за период эксплуатации, на половину разности производительности следующих и современных поколений ЭВМ - сейчас на 1,5 порядка (т.е., в 500 раз);
  - 13) предложенные основы и методы в совокупности позволяют повысить реальную производительность адаптивных ЭВМ вдвое, т.е. относительно пиковой мощности обычных ЭВМ с 10-30 до 50-70 %.
  - 14) предложенные основы и методы в совокупности позволят снизить материально-финансовые затраты на разработку и эксплуатацию АСОИ путем постоянного наращивания вычислительной мощности на основе адаптации и добавления только наиболее современных, на каждом этапе существования ЭВМ, компьютерных модулей, т.е. путем создания эволюционных неоднородных компьютерных систем;
  - 15) полученные новые научные результаты могут использоваться как по отдельности, так и вместе для решения проблемы создания эволюционных баз данных и правил для адаптивного синтеза интеллектуальных систем, которые будут способны изменять не только свои программные средства, но и изменять, наращивать и модернизировать аппаратные средства ЭВМ.

Итак, в данной монографии изложены основные результаты по разработке теоретических основ создания эволюционных баз данных и правил (знаний) с целью построения систем адаптивного синтеза информационно-вычислительных конфигураций и интеллектуальных систем. Надеемся, что работы в данном на-

правления будут продолжены, как минимум, до достижения конкретных результатов по созданию интеллектуальных эволюционных программно-аппаратных комплексов. Автор благодарен читателям, уделившим внимание этой монографии. Желаем всем творческих и научных успехов, а, кроме того, хотелось бы надеяться, что изложенный в данной книге материал окажется полезным для читателей, какими бы конкретными теоретическими и практическими вопросами они бы ни занимались.

## СПИСОК ЛИТЕРАТУРЫ

1. VI Санкт-Петербургская Международная Конференция "Региональная информатика - 98" ("РИ - 98"), Санкт-Петербург, 2 - 4 июня 1998 г.: Тез. докл. Ч. 1. - СПб., 1998 - 174 с.
2. VI Санкт-Петербургская Международная Конференция "Региональная информатика - 98" ("РИ - 98"), Санкт-Петербург, 2 - 4 июня 1998 г.: Тез. докл. Ч. 2. - СПб., 1998 - 170 с.
3. А. с. 176724 СССР, МКИ G 06 F 7/50, 1965.
4. А. с. 249050 СССР, МКИ G 06 F 7/50, 1969.
5. Абрамов С.М. Метавычисления и их применение: Дис. ... д-ра физ. - мат. наук. - Переславль-Залесский, 1995.
6. Абрамов С.М. Метавычисления и их применения. - М.: Наука, Физматлит, 1995.
7. Абрамов С.М., Адамович А.И., Коваленко М.Р. Т-система - среда программирования с поддержкой автоматического динамического распараллеливания программ. Пример реализации алгоритма построения изображений методом трассировки лучей // Программирование, 1999, № 2, с. 51 - 61.
8. Агеев А.С., Борисов С.А., Диев С.А. и др. Организация и современные методы защиты информации / Под ред. Диева С.А., Шаваева А.Г. - М., Концерн "Банковский Деловой центр", 1998. - 472 с.
9. Адаптивные и экспертные системы в управлении // Тез. докл. 5-го Ленинградского симпозиума по теории адаптивных систем "ТАС - 91", 17 - 19 апреля 1991г. - Л., 1991.
10. Алгоритмы, математическое обеспечение и архитектура многопроцессорных вычислительных систем / Под ред. Ершова А.П. - М.: Наука, 1982.
11. Амамия М., Танака Ю. Архитектура ЭВМ и искусственный интеллект. - М.: Мир, 1993.
12. Анисимов Б.В., Четвериков В.Н. Основы теории и проектирования цифровых вычислительных машин. - М.: Высшая школа, 1990.
13. Ануреев И.С. Метод элиминации структур данных, основанный на системах переписывания формул // Программирование, 1999, № 4, с. 5 - 15.
14. Аристотель. Соч. В 4-х т. - М., 1975.
15. Арлазаров В.Л., Долгопятова Е.Ю., Иванова Н.А., Иофинова М.Е., Леман А.А., Мерков А.Б. Концепции языка программирования над базами данных Модуля-90 // Программирование, 1991, № 3, с. 15 - 33.
16. Артамонов Г.Т., Тюрин В.Д. Топология сетей ЭВМ и многопроцессорных систем. - М.: Радио и связь, 1991.
17. Арутюнян Э.С. Синтез нейросетевых систем управления манипуляционными роботами на основе обратных динамических моделей: Автореферат дис. ... канд. техн. наук. - Уфа, 2000. - 16 с.
18. Архипов А.В. Почему сильные государства любят слабую криптографию // Защита информации. Конфидент, № 3, 1997, с. 65 - 67.
19. Атре Ш. Структурный подход к организации баз данных. - М.: Финансы и статистика, 1983.
20. Ачасова С.М., Бандман О.Л. Корректность параллельных вычислительных процессов. - Новосибирск: Наука, 1990.
21. Бабаян Б.А., Бочаров А.В., Волин А.С. и др. Многопроцессорные ЭВМ и методы их проектирования / Под ред. Смирнова Ю.М. - М.: Высшая школа, 1990.

22. Барский А.Б. Параллельные процессы в вычислительных системах. Планирование и организация. - М.: Радио и связь, 1990. - 256 с.
23. Батыршин И.З. Представление и обработка нечеткой информации в интеллектуальных системах: Дис. ... д-ра физ. - мат. наук. - Переславль-Залесский, 1996.
24. Белеградек О.В., Столбоушкин А.П., Тайцлин М.А. Базы данных над фиксированным бесконечным универсумом // Программирование, 1998, № 1, с. 6 - 17.
25. Белецкий В.Н. Многопроцессорные и параллельные структуры с организацией асинхронных вычислений. - Киев: Наукова думка, 1988.
26. Белов В.В. Технология комплексной оценки качества интеллектуальных программных продуктов двойного применения в процессе сертификационных испытаний: Автореферат дис. ... канд. техн. наук. - М., 2000. - 25 с.
27. Белоусов Н.Н. Конструирование преобразователей моделей данных в системе интеграции неоднородных баз данных: Дис. ... канд. физ. - мат. наук. - М., 1987.
28. Бениаминов Е.М. Алгебраические методы в теории баз данных и представлении знаний: Дис. ... д-ра физ. - мат. наук. - М., 1996.
29. Берг Л.С. Труды по теории эволюции. 1922 - 1930. - Л.: Наука, 1977.
30. Берталанфи Л. фон. Общая теория систем - обзор проблем и результатов // Системные исследования (ежегодник). - М.: Наука, 1969. - С. 30 - 55.
31. Бирюков Б.В., Гутчин И.Б. Машина и творчество. Результаты, проблемы, перспективы. - М.: Радио и связь, 1982.
32. Бриллюэн Л. Научная неопределенность и информация. - М., 1966.
33. Брусенцов Н.П. Блуждание в трех соснах. (Приключения диалектики в информатике) // Программные системы и инструменты. Тематический сборник № 1. - М.: Изд-во факультета ВМ и К МГУ, 2000.
34. Бурцев В.С. Новые подходы к созданию высокопараллельных вычислительных структур // Искусственный интеллект - 2000. Тез. докл. науч. конф. - Таганрог: ТРТУ, 2000.
35. Бурцев В.С. Принципы построения многопроцессорных вычислительных комплексов "Эльбрус". - Препринт. - М., 1977.
36. Буч Г. Объектно-ориентированное проектирование с примерами применения / Пер. с англ. - М.: Конкорд, 1992.
37. Валиев К., Медовников Д., Тюменев В. Работы у наших потомков не будет // Эксперт, № 17, 8 мая 2000г., с. 28 - 29.
38. Вальковский В.А. Распараллеливание алгоритмов и программ. Структурный подход. - М.: Радио и связь, 1989.
39. Вальковский В.А., Малышкин В.Э. Синтез параллельных программ и систем на вычислительных моделях. - Новосибирск: Наука, 1988.
40. Валях Е. Последовательно-параллельные вычисления / Пер. с англ. - М.: Мир, 1985.
41. Варламов О.О. Алгоритм разреза сети по вершинам и ребрам ее графа сложности  $O(n^2)$  // Научно-технический сборник НИИ Радио за 1997г. - М.: Изд-во НИИ Радио, 1997.
42. Варламов О.О. Об одном подходе к разработке квадратичной сложности алгоритма поиска минимального разреза многополюсной сети // Труды 6-го Международного семинара по дискретной математике Мех-Мата МГУ. 3-6.02.1998г. - М.: МГУ, 1998.
43. Варламов О.О. Предложения по основным направлениям и путям создания высокопроизводительных вычислительных комплексов в России // Тез. докл. науч. - техн. конф. ИМС'99 НИИ МВС ТРТУ. - Таганрог: Изд-во ТРТУ, 1999. - С. 104 - 106.

44. Варламов О.О. Проблема адаптивного синтеза конфигураций компьютерных систем для решения сложных задач // Сб. науч. тр. - М.: МИФИ, 2001. Т. 2, с. 228 - 229.
45. Варламов О.О. Проблема обеспечения потоковой параллельной обработки баз данных // Тр. Юбилейной науч. - техн. конф. ЦНИИРЭС. - М.: ЦНИИРЭС, 2001. Ч. 2, с. 106 - 108.
46. Варламов О.О. Проблема создания эволюционных неоднородных компьютерных систем // Тр. Юбилейной науч. - техн. конф. ЦНИИРЭС. - М.: ЦНИИРЭС, 2001. Ч. 2, с. 103-105.
47. Васильев В.И., Шевченко А.И. Формирование и опознавание образов. Серия "Искусственный интеллект". - Донецк: Изд-во ДонГИИИ, 2000. - 360 с.
48. Васильев Г.П. Программное обеспечение неоднородных распределенных систем. Анализ и реализация. - М.: Финансы и статистика, 1986.
49. Вахтомин Н.К. Генезис научного знания. - М., 1973.
50. Вейд А. Стандарты объектных запросов // Системы управления базами данных, 1996, №4.
51. Величко И.В. Разработка и реализация СУБД сетевого типа: Дис. ... канд. физ. - мат. наук. - М., 1981.
52. Винер Н. Кибернетика, или управление и связь в животном и машине. - М.: Радио, 1958.
53. Витгенштейн Л. Логико-философский трактат. - М., 1958.
54. Воеводин В.В. Информационная структура алгоритмов. - М.: Изд-во МГУ, 1997.
55. Воеводин В.В. Математические модели и методы в параллельных процессах. - М.: Наука, 1986.
56. Воеводин В.В. Отображение проблем вычислительной математики на архитектуру вычислительных систем // Вычислительная математика и математическое моделирование. Тр. международной конф. Т. 1. - М.: Ин-т вычисл. математики РАН, 2000. - С. 242 - 255.
57. Воеводин В.В. Параллельные структуры алгоритмов и программ. - М.: ОВМ АН СССР, 1987.
58. Волькенштейн М.В. Сущность биологической эволюции // Успехи физич. наук. - 1984. - Т. 143, Вып. 3. - С. 429 - 466.
59. Вольтерра В. Математическая теория борьбы за существование. - М.: Наука, 1976.
60. Воронин Ю.А. Теория классифицирования и ее приложения. - Новосибирск: Наука, 1985.
61. Воронцов Н.Н. Теория эволюции: истоки, постулаты и проблемы. - М.: Знание, 1984.
62. Высокоскоростные вычисления. Архитектура, производительность, прикладные алгоритмы и программы суперЭВМ // Под ред. Я.К. Ковалика. - М.: Радио и связь, 1988.
63. Вычислительная математика и математическое моделирование // Труды международной конференции. Том 1,2. - М.: Ин-т вычислительной математики РАН, 2000.
64. Вычислительные системы. - М.: Финансы и статистика, 1982.
65. Гаврилова Т.А., Червинская К.Р. Извлечение и структурирование знаний для экспертных систем. - М.: Радио и связь, 1992.
66. Гаек П., Гавранек Т. Автоматическое образование гипотез (Математические основы общей теории). - М.: Наука, 1984. - 262 с.

67. Гайсинович А.Е. Зарождение генетики. - М.: Наука, 1967.
68. Гегель Г. Соч. В 6-ти т. - М., 1939.
69. Гегель Г. Энциклопедия философских наук. - М., 1974.
70. Гиглавый А.В. Исследование и разработка методов управления единым архивом данных применительно к задачам автоматизации проектирования цифровых устройств: Дис. ... канд. техн. наук. - М., 1975.
71. Гиглавый А.В. Теория баз данных. Меморандум трех // Программирование, 1992, № 2.
72. Глушков В.М. Кибернетика, вычислительная техника, информатика. Избранные труды: В 3 т. - Киев: Наукова думка, 1990.
73. Головкин Б.А. Вычислительные системы с большим числом процессоров. - М.: Радио и связь, 1995.
74. Головкин Б.А. Параллельные вычислительные системы. - М.: Наука, 1980.
75. Головкин Б.А. Расчет характеристик и планирование параллельных вычислительных процессов. - М.: Радио и связь, 1983. - 272 с.
76. Горбань А.Н. Обход равновесия. - Новосибирск: Наука, 1984.
77. Горбань А.Н., Хлебопрос Р.Г. Демон Дарвина: Идея оптимальности и естественный отбор. - М.: Наука. Гл. ред. физ. - мат. лит., 1988. - 208 с.
78. Городецкий В.И. Прикладная алгебра и дискретная математика. Часть 1. Алгебраические системы. - МО СССР, 1984. - 174 с.
79. Городецкий В.И. Прикладная алгебра и дискретная математика. Часть 2. Формальные системы нелогического типа. - МО СССР, 1986. - 200 с.
80. Городецкий В.И. Прикладная алгебра и дискретная математика. Часть 3. Формальные системы логического типа. - МО СССР, 1987. - 177 с.
81. Горский Д.П. Вопросы абстракции и образования понятий. - М., 1961.
82. Горшкова Е.А., Некрестьянов И.С., Новиков Б.А., Павлова Е.Ю. Поддержка согласованности для слабоструктурированных данных // Программирование, 2000, № 3, с. 23 - 30.
83. Грант В. Эволюция организмов. - М.: Мир, 1980.
84. Гурин В.Н. Математические основы организации и обработки информации в одном классе многомерных структур данных: Дис. ... канд. физ. - мат. наук. - Грозный, 1982.
85. Дарвин Ч. Происхождение видов. - М.; Л.: Биомедгиз, 1937.
86. Дейкстра Э. Взаимодействие последовательных процессов // Языки программирования. - М.: Мир, 1972.
87. Дейт К. Введение в системы баз данных. - М.: Наука, 1980.
88. Дзюба С.М. Разработка методов исследования периодических процессов в задачах управления: Дис. ... д-ра физ. - мат. наук. - М., 1998.
89. Дмитриев Ю., Задорожный А., Корнеев В. Элементарная машина вычислительной системы с программируемой структурой МИКРОС // Вычислительные системы с программируемой структурой. Выч. системы-94. ИМ СОАН СССР. - Новосибирск, 1982.
90. Дмитриев Ю., Корнеев В., Хорошевский В. Вычислительная система с программируемой структурой МИКРОС // Вычислительные системы с программируемой структурой. Вычислительные системы, 94. ИМ СО АН СССР. - Новосибирск, 1982.
91. Долидзе А.Н. Концептуальная модель предметной области со слабовыраженной структурой: Дис. ... канд. физ. - мат. наук. - Тбилиси, 1988.
92. Дрейфус Х.Л. Чего не могут вычислительные машины. Критика искусственного разума. - М.: Прогресс, 1978.

93. Дунаев В. Молеетроника - это сложное слово // Независимая газета, приложение "НГ - Наука", №8, 15 сентября 1999, с. 4.
94. Евреинов Э., Хорошевский В. Однородные вычислительные системы. - Новосибирск: Наука, 1978.
95. Евреинов Э.В. Однородные вычислительные системы, структуры и среды. - М.: Радио и связь, 1981.
96. Евреинов Э.В., Косарев Ю.Г. Однородные универсальные вычислительные системы высокой производительности. - Новосибирск: Наука, 1966.
97. Ефимов Е.И. Субъектно-предикатные модели обоснования тезисов. - М., 1991.
98. Ефимова С.М. Применение П-графов для представления и поиска знаний в динамических базах знаний: Дис. ... канд. физ. - мат. наук. - М., 1986.
99. Завадский К.М., Колчинский Э.И. Эволюция эволюции. - Л.: Наука, 1977.
100. Замулин А.В. Интегральная языковая среда для описания и реализации баз данных различных моделей: Дис. ... д-ра физ. - мат. наук. - Новосибирск, 1989.
101. Замулин А.В. Перспективы развития системы баз данных третьего поколения // Программирование, 1992, № 2, с. 33 - 46.
102. Замулин А.В. Программное обеспечение информационно-поисковой системы общего назначения: Дис. ... канд. физ. - мат. наук. - Новосибирск, 1974.
103. Замулин А.В. Системы программирования баз данных и знаний. - Новосибирск: Наука, 1990.
104. Замулин А.В. Типы данных в языках программирования и базах данных. - Новосибирск: Наука, 1987.
105. Защита информации. Конфидент, 1995 - 2000 (<http://www.confident.spb.su>).
106. Зимачева Е.М. Способы вербальной презентации образа "Я" и самоотношение субъекта: Дис. ... канд. псих. наук. - М., 1997.
107. Зыкин С.В. Формирование пользовательского представления реляционной базы данных с помощью отображений // Программирование, 1999, № 3, с. 70 - 80.
108. Ивахненко А.Г., Зайченко Ю.П., Димитров В.Д. Принятие решений на основе самоорганизации. - М.: Сов. радио, 1976.
109. Ильин В.П. О стратегиях распараллеливания в математическом моделировании // Программирование, 1999, № 1, с. 41 - 46.
110. Ильин Г.М. Модели информационных систем: Дис. ... канд. физ. - мат. наук. - Л., 1975.
111. Илюшин А.И. Построение системы управления данными на основе ограниченного числа базисных понятий: Дис. ... канд. физ. - мат. наук. - Б. м., 1974.
112. Интеллектуальные и многопроцессорные системы-2001. - Таганрог: ТРТУ, 2001. - 344 с.
113. Интеллектуальные многопроцессорные системы. ИМС-99 - Таганрог: ТРТУ, 1999. - 256с.
114. Интеллектуальные робототехнические системы - 2001. - Таганрог: ТРТУ, 2001. - 318 с.
115. Информационные и рабочие материалы НИИ МВС ТРТУ, 2000 - 2002.
116. Информационные и рабочие материалы НИЦЭВТ, 2000 - 2001.
117. Иофинова М.Е. Решетки помеченных деревьев в задаче согласованного обновления иерархических баз данных: Дис. ... канд. физ. - мат. наук. - М., 1989.
118. Искусственный интеллект-2000 // Тез. докл. науч. конф. - Таганрог: ТРТУ, 2000. - 202 с.
119. История марксистской диалектики. - М., 1971.

120. Калиниченко Л.А. Метод построения коммутативных отображений моделей данных при интеграции неоднородных баз данных // Программирование, 1999, № 6, с. 38 - 47.
121. Калиниченко Л.А. Методы и средства интеграции неоднородных баз данных. - М.: Наука, 1983.
122. Калиниченко Л.А., Манукян М.Г. Языковые средства временно - ориентированной модели данных // Программирование, 1990, № 5, с. 73 - 86.
123. Калиниченко Л.А., Рывкин В.М. Машины баз данных и знаний. - М.: Наука. Гл. ред. физ.-мат. лит., 1990. - 296 с.
124. Калиниченко Л.А., Скворцов Н.А., Брюхов Д.О., Кравченко Д.В., Чабан И.А. Проектирование персонализированных электронных библиотек над Web-сайтами со слабоструктурированными данными // Программирование, 2000, № 3, с. 9 - 22.
125. Калинов А.Я. Неоднородное двумерное блочно-циклическое распределение данных для решения задач линейной алгебры на неоднородных сетях компьютеров // Программирование, 1999, № 2, с. 3 - 11.
126. Каляев А.В. Многопроцессорные системы с программируемой архитектурой. - М.: Радио и связь, 1984.
127. Каляев А.В. Суперкомпьютеры, супернейрокомпьютеры и искусственный интеллект // Искусственный интеллект-2000. Тез. докл. науч. конф. - Таганрог: ТРТУ, 2000. - С.12-14.
128. Каляев А.В., Каляев И.А., Левин И.И., Пономарев И.М. Базовый модуль для построения реконфигурируемых под задачу вычислительных систем // Известия ВУЗов. - Электроника, 1998, № 4, с. 67 - 74.
129. Каляев А.В., Каляев И.А., Левин И.И., Пономарев И.М. Параллельный компьютер с программируемой под структуру задачи архитектурой // Труды шестого международного семинара "Распределенная обработка информации". - Новосибирск, 1998. - С. 25 - 29.
130. Каляев А.В., Левин И.И., Шматок А.В. Средства программирования суперкомпьютеров с массовым параллелизмом и программируемой архитектурой // Искусственный интеллект - 2000. Тез. докл. науч. конф. - Таганрог: Изд-во ТРТУ, 2000. - С. 145 - 148.
131. Каляев А.В., Станишевский О.Б. Принципы построения программно-аппаратных средств супермакрокомпьютеров // Информатика. - М.: 1990, № 2, с. 13 - 21.
132. Калянов Г.Н. Структурный системный анализ (автоматизация применения). - М.: Лори, 1996.
133. Камшилов М.М. Эволюция биосферы. - М.: Наука, 1979.
134. Кант И. Соч. В 6-ти т. - М., 1965.
135. Капитонова Ю.В., Летичевский А.А. Математическая теория проектирования вычислительных систем. - М.: Наука. Гл. ред. физ.-мат. лит., 1988.
136. Ким Вон. Технология объектно-ориентированных баз данных // Открытые системы, 1994, № 4.
137. Китаев А., Медовников Д., Тюменев В. Защита для кубита // Эксперт, №17, 2000, с.30-31.
138. Клайн М. Математика. Утрата определенности / Пер. с англ. - М.: Мир, 1984. - 446 с.
139. Ковтун И.И. Матрично-реляционная модель данных в организационно-производственных системах мониторинга и управления: Дис. ... канд. техн. наук. - М., 1997. - 183 с.

140. Ковтун И.И. Матрично-реляционная модель данных для представления и обработки информации в автоматизированных системах мониторинга и управления // Программирование, 1997, № 6, с. 58 - 72.
141. Когаловский М.Р. Систематика коллекций информационных ресурсов в электронных библиотеках // Программирование, 2000, № 3, с. 30-52.
142. Когаловский М.Р., Новиков Б.А. Электронные библиотеки - новый класс информационных систем (от составителей выпуска) // Программирование, 2000, № 3, с. 3 - 8.
143. Козленко Л.А. Разработка математического обеспечения систем управления вложенными транзакциями в параллельных и распределенных средах: Автореферат дис. ... канд. техн. наук. - Воронеж, 2000.
144. Компьютера, 1998 - 2002.
145. Концепция развития безопасных информационных технологий: обеспечение защиты информации в проектах информатизации России. Отчет. - М., 1992. - 149 с.
146. Корженевич Ю.В., Кобайло А.С. Синтез вычислительных структур. - М.: Наука и техника, 1989.
147. Корнеев В.В. Архитектура вычислительных систем с программируемой структурой. - Новосибирск: Наука, 1985.
148. Корнеев В.В. Параллельные вычислительные системы. - М.: Нолидж, 1999.
149. Корнеев В.В., Киселев А.В. Современные микропроцессоры. - М.: Нолидж, 1998.
150. Костенко В.А., Смелянский Р.Л., Трекин А.Г. Синтез структур вычислительных систем реального времени с использованием генетических алгоритмов // Программирование, 2000, № 5, с. 63 - 72.
151. Коуги П.М. Архитектура конвейерных ЭВМ / Пер. с англ. - М.: Радио и связь, 1985.
152. Кохонен Т. Ассоциативная память. - М.: Мир, 1980.
153. Краткий психологический словарь - хрестоматия. - М.: Высшая школа, 1974.
154. Кручинин С. Стандартные тесты измерения производительности // COMPUTER WEEK - Москва, 1996, № 19, с. 31 - 35.
155. Кузнецов С.Д., Пономаренко В.Н. Выработка оптимальных планов выполнения запросов в реляционных СУБД // Программирование, 1990, № 2.
156. Кузьминский М. Между строк таблиц Linpack // COMPUTER WEEK - Москва, 1998, № 13, с. 41 - 43.
157. Курошев М.Ю., Рывкин В.М. Распараллеливание реляционных операций в сетевой машине баз данных // Управляющие системы и машины, 1991, № 7.
158. Лавров С.С., Залогова Л.А., Петрушина Т.И. Принципы планирования решения задач в системе автоматического синтеза программ // Программирование, 1999, № 6, с. 67 - 73.
159. Ладенко И.С. Имитационные системы. - Новосибирск: Наука, 1981. - 300 с.
160. Ладенко И.С. Интеллектуальные системы в целевом управлении. - Новосибирск: Наука, 1987. - 199 с.
161. Ладенко И.С. Интеллектуальные системы и логика. - Новосибирск: Наука, 1973. - 172 с.
162. Ладенко И.С. Логические методы построения математических моделей. - Новосибирск: Наука, 1980. - 192 с.
163. Лазарев И.А. Информация и безопасность. Композиционная технология информационного моделирования сложных объектов принятия решений. - М.: МГЦНТИ, 1997. - 336 с.

164. Ластовецкий А.Л. Язык и система параллельного программирования для разработки программ, эффективно переносимых в классе распределенных вычислительных систем: Дис. ... д-ра физ.-мат. наук. - М., 1997.
165. Левин И.И. Структурно-процедурное программирование // Искусственный интеллект-2000. Тез. докл. науч. конференции. - Таганрог: Изд-во ТРТУ, 2000. - С. 148 - 151.
166. Левин И.И., Коробкин В.В. Принципы создания интеллектуальной самонастраиваемой элементной базы для эффективного аппаратного решения широкого класса задач // Искусственный интеллект-2000. Тез. докл. науч. конф. - Таганрог: ТРТУ, 2000. - С. 151 - 153.
167. Левин И.И., Пономарев И.М. Методика организации высокоэффективных параллельных вычислений в многопроцессорных системах. // Тезисы международной конференции "Искусственный интеллект-2000". - Таганрог: Изд-во ТРТУ, 2000. - С. 142 - 145.
168. Левонтин Р. Генетические основы эволюции. - М.: Мир, 1978.
169. Ленин В.И. Полн. собр. соч. - М., Гос. изд-во полит. литературы, 1947.
170. Ливчак А.Б. Полнота языков запросов // Программирование, 1993, № 2.
171. Логический подход к искусственному интеллекту: От модальной логики к логике баз данных / Пер. с франц. // Тейз А., Грибомон П., Юлен Г. и др. - М.: Мир, 1998.
172. Лойфман И.Я. Принципы физики и философские категории. - Свердловск, 1973.
173. Ломазова И.А. Каузальная семантика для сетей Петри с контактами // Программирование, 1999, № 4, с. 43 - 53.
174. Лорьер Ж.-Л. Системы искусственного интеллекта / Пер. с франц. - М.: Мир, 1991. - 568с.
175. Лэнгсам Й., Огенстайн М., Тененбаум А. Структуры данных для персональных ЭВМ / Пер. с англ. - М.: Мир, 1989. - 568 с.
176. Любарский Ю.Я. Интеллектуальные информационные системы. - М., Наука. Гл. ред. физ.-мат. лит., 1990.
177. Майр Э. Популяции, виды и эволюция. - М.: Мир, 1974.
178. Мальковский М.Г., Шикин И.Ю. Нечеткий лингвистический интерфейс // Программирование, 1998, № 4, с. 50 - 61.
179. Мальцев А.А. Алгебраические системы. - М.: Наука, 1970.
180. Мальцев А.А. Алгоритмы и рекурсивные функции. - М.: Наука, 1965. - 390 с.
181. Мамардашвили М.К. Формы и содержание мышления. - М., 1968.
182. Маркс К., Энгельс Ф. Соч. - М.; Л., 1930.
183. Мартин Дж. Организация баз данных в вычислительных системах. - М.: Мир, 1980.
184. Мартыненко Б.К. Синтаксически управляемая обработка данных: Дис. ... д-ра физ. - мат. наук. - СПб., 1997.
185. Мартынов М.Г. Пространственные методы доступа // Программирование, 1998, № 3, с. 59 - 69.
186. Материалистическая диалектика как общая теория развития. - М.: Наука, 1983.
187. Медавар П., Медавар Дж. Наука о живом. Современные концепции в биологии. - М.: Мир, 1983.
188. Медников Б.М. Аксиомы биологии. - М.: Знание, 1982.
189. Медовников Д., Тюменев В. Считающий атом // Эксперт, № 17, 8 мая 2000, с. 24 - 27.
190. Мейер Д. Теория реляционных баз данных. - М.: Мир, 1987.

191. Методы и средства создания и исследования экспертных систем: Сб. науч. тр. // АН СССР, Дальневосточное отд. - Владивосток, 1991.
192. Мидоу Ч. Анализ информационно-поисковых систем. - М.: Мир, 1970.
193. Минский М. Вычислительные машины и мышление. - М.: Мир, 1967.
194. Минский М. Фреймы для представления знаний. - М.: Энергия, 1979.
195. Миренков Н.Н. Параллельное программирование для многомодульных вычислительных систем. - М.: Радио и связь, 1989.
196. Михайлова Е.Г. Структуры хранения для временных баз данных // Программирование, 1997, № 6, с. 73 - 80.
197. Монахов О. Параметрическое описание структур однородных вычислительных систем. Вопросы теории и построения вычислительных систем. - Новосибирск: ИМ СОАН СССР, 1979.
198. Монахова Э. Об аналитическом задании оптимальных двумерных диофантовых структур однородных вычислительных систем // Однородные вычислительные системы. - Новосибирск, ИМ СО АН СССР, 1981.
199. Морозов Е.А. Особенности ведения журналов на магнитных дисках // Программирование, 1992, № 2.
200. Мутушев Д.М. Методы обеспечения доступа к объектно-ориентированным базам данных на основе стандартов реляционных систем. Автореферат дис. ... канд. физ. - мат. наук. - М., 1998.
201. Мутушев Д.М. Реализация расширения объектной модели ODMG в среде реляционных СУБД // Программирование, 1998, № 3, с. 46 - 58.
202. Мутушев Д.М., Филиппов В.И. Объектно-ориентированные базы данных // Программирование, 1995, № 6.
203. Назин А.В., Позняк А.С. Адаптивный выбор вариантов: Рекуррентные алгоритмы. - М.: Наука. Гл. ред. физ. - мат. лит., 1986.
204. Научная сессия МИФИ - 2000. // Сборник научных трудов. В 13 томах. Т. 2. Информатика и процессы управления. Информационные технологии. Сетевые технологии и параллельные вычисления. - М.: Изд-во МИФИ, 2000. - 208 с.
205. Научная сессия МИФИ - 2000. // Сборник научных трудов. В 13 томах. Т. 3. Банки данных и анализ данных. Интеллектуальные системы. Технологии разработки программных систем. - М.: Изд-во МИФИ, 2000. - 228 с.
206. Научная сессия МИФИ - 2000. // Сб. науч. трудов. В 13 томах. - М.: Изд-во МИФИ, 2000.
207. Научная сессия МИФИ - 2001. // III Всероссийская научно-техническая конференция "Нейроинформатика-2001". - М.: МИФИ, 2001. - 212 с.
208. Научная сессия МИФИ - 99. // Всероссийская научно-техническая конференция "Нейроинформатика-99". Дискуссия о нейрокомпьютерах. - М.: МИФИ, 2000. - 224 с.
209. Научная сессия МИФИ-2001. // Сборник научных трудов. В 14 томах. Т. 12. Компьютерные системы и технологии. - М.: Изд-во МИФИ, 2001. - 208 с.
210. Научно-технические основы информатизации России. - М., 1992.
211. Научные сессии МИФИ - 2000, 2001. // Квантовые нейронные сети: Материалы рабочего совещания. - М.: Изд-во МИФИ, 2001. - 104 с.
212. Нейман Дж. фон. Теория самовоспроизводящихся автоматов / Пер. с англ. - М.: Мир, 1971.
213. Нигяян С.А. Функциональные и логические языки программирования (формализация, анализ, интерпретация): Дис. ... д-ра физ.-мат. наук. - Ереван, 1997.
214. Николаи Дж. От чего защищают ограничения на экспорт суперкомпьютеров? // COMPUTER WEEK - Москва, 1998, № 21, с. 27.

215. Нильсон Н. Искусственный интеллект. Методы поиска решений. - М.: Мир, 1973.
216. Новиков Б.А. Системы хранения баз данных и знаний // Программирование, 1993, 2, с. 3-30.
217. О сущности жизни. - М.: Наука, 1964.
218. Овчинников Н.Ф. Качество и свойство // Вопросы философии, 1960, № 6.
219. Окунишникова Е.В. Временные сети Петри без перекрытий интервалов срабатывания // Программирование, 1998, № 5, с. 15 - 29.
220. Олле Т.В. Предложения КОДАСИЛ по управлению базами данных. - М.: Финансы и статистика, 1981.
221. Оно С. Генетические механизмы прогрессивной эволюции. - М.: Мир, 1973.
222. Орфеев Ю.В., Тюхтин В.С. Мышление человека и искусственный интеллект. - М.: Мысль, 1978.
223. Пантеев С.Р. Строеие самоотношения как эмоционально-оценочной системы: Дис. ... канд. псих. наук. - М., 1989.
224. Параллельные вычисления / Пер. с англ. // Под ред. Г. Родрига. - М.: Наука, 1986.
225. Пат. RU 2145113 С1 Россия, МКИ 7 G 06 F 7/50. Способ суммирования чисел. / Варламов О.О. (Россия). - № 98119301/09(021284); Заявлен 23.10.1998; Опубликовано 27.01.2000. Бюллетень № 3. - 12 с., 1 л. ил.
226. Першиков В.И., Савинков В.М. Толковый словарь по информатике. - М.: Финансы и статистика, 1991. - 543 с.
227. Печников А.А. Вопросы логической и физической организации информационных систем: Дис. ... канд. физ. - мат. наук. - Л., 1981.
228. Питерсон Дж. Теория сетей Петри и моделирование систем / Пер. с англ. - М., 1984. - 264 с.
229. Планк М. Смысл и границы точной науки // Вопросы философии, 1958, № 5, с. 111.
230. Плеханов Г.В. Избранные философские произведения. - М., 1956.
231. Плоткин Б.И. Универсальная алгебра, алгебраическая логика и базы данных. - М.: Наука, 1991.
232. Попов Э.В. Искусственный интеллект: экспертные системы. - М.: Наука, 1990.
233. Попов Э.В., Фридман Т.Р. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта. - М.: Наука, 1976.
234. Поршнева Б.Ф. О начале человеческой истории. - М., 1974.
235. Поспелов Г.С., Поспелов Д.А. Искусственный интеллект - прикладные системы. - М.: Знание, 1985. - 48 с.
236. Поспелов Д.А. Введение в теорию вычислительных систем. - М.: Сов. радио, 1983. - 272 с.
237. Поспелов Д.А. Логико-лингвистические модели в системах управления. - М.: Энергоиздат, 1981. - 232 с.
238. Прикладная статистика: Классификация и снижение размерности: Справ. изд. // Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. // Под ред. Айвазяна С.А. - М.: Финансы и статистика, 1989.
239. Принципы самоорганизации. - М.: Мир, 1960.
240. Проблемы логики научного познания. - М., 1964.
241. Проблемы программно-целевого планирования и управления. // Под ред. Поспелова Г.С. - М.: Наука, ГРФМЛ, 1981.
242. Проблемы проектирования экспертных систем // Тез. докл. Всесоюзной школы - совещания, 22 - 30 апреля 1988.

243. Проблемы разработки и внедрения экспертных систем // Тез. докл. Всесоюзной конференции. - М., 1989.
244. Программные системы и инструменты: Тематический сборник факультета ВМ и К МГУ № 1 // Под ред. Королева Л.Н. - М.: МАКС Пресс, 2000. - 164 с.
245. Пудовченко Ю.Е. Когда наступит время подбирать ключи // Защита информации. Конфидент, 1998, № 3, с. 65 - 71.
246. Райбекас А.Я. Вещь, свойство, отношение как философские категории. - Томск: Изд-во Томского университета, 1977.
247. Рамьялг А.Е. Разработка метода отображения сетевой модели данных в реляционную и его реализация: Дис. ... канд. физ. - мат. наук. - М., 1979.
248. Решение задач в интеллектуальных компьютерных средах: сб. науч. тр. // АН Украины. - Киев, 1991.
249. Ривкин М.Н. Векторные операции для моделирования процедур преобразования данных // Программирование, 1991, № 3.
250. Рокотов В.П. Автоматизированное управление формированием и реализацией планов: Дис. ... д-ра техн. наук. - М., 1998.
251. Ростовцев Ю.Г. Информация и представление знаний // Известия ВУЗ. Приборостроение, 1981, № 4, с. 92 - 96.
252. Ростовцев Ю.Г. Математические методы и модели оценивания военно-политической обстановки. - МО СССР, 1986. - 312 с.
253. Ростовцев Ю.Г., Рахматуллин М.Ю. Формализация целей в системе взаимодействующих роботов // Известия ВУЗ. Приборостроение, 1984, № 11, с. 3 - 9.
254. Рубинштейн С.А. Бытие и сознание. - М., 1957.
255. Рыбин С.В. Программно-аналитический метод анализа и управления распределением ресурсов вычислительных систем на основе композитных моделей: Автореферат дис. ... канд. физ. - мат. наук. - М., 2000.
256. Саати Т.Л. Математические модели конфликтных ситуаций. - М.: Сов. радио, 1977.
257. Сазонов В.Ю. Базы данных типа Web, антифундированные множества и индуктивная определимость // Программирование, 1999, № 5, с. 26 - 43.
258. Саймон Г. Науки об искусственном. - М.: Мир, 1972.
259. Сапожникова Р.Б. Психологические факторы формирования самооотношения личности: Дис. ... канд. псих. наук. - Новосибирск, 1998.
260. Северцов А.Н. Морфологические закономерности эволюции. - М.; Л.: АН СССР, 1939.
261. Сети и системы связи, 1996 - 2002 (<http://www.ccc.ru>).
262. Сигнаевский В.А., Коган Я.А. Методы оценки быстродействия вычислительных систем. - М.: Наука. Гл. ред. физ. - мат. лит., 1991.
263. Сиротнюк В.О. Разработка моделей, методов и инструментальных средств анализа и синтеза оптимальных структур баз данных в автоматизированных информационно - управляющих системах: Дис. ... д-ра техн. наук. - М., 1999.
264. Системы параллельной обработки / Пер. с англ. // Под ред. Д. Ивенса. - М.: Мир, 1985.
265. Слуцкий В.М. Влияние оценки взрослого на формирование отношения к себе у детей: Дис. ... канд. псих. наук. - М., 1985.
266. Создание и применение гибридных экспертных систем // Тез. докладов Всесоюзной конференции. - Рига, 1990.
267. Спиркин А.Г. Сознание и самосознание. - М.: Политиздат, 1972.
268. Средство построения ЭС XLEX, предназначенное для автоматизации процессов // Всесоюзный центр переводов, № С-54349, 1989.

269. Станишич П. Метод трансляции реляционных SQL-запросов в эквивалентные запросы к трансформированной объектно-ориентированной базе данных // Труды факультета ВМ и К МГУ. Прикладная математика и информатика. 1999, № 1.
270. Станишич П. Трансформация реляционных баз данных в объектно-ориентированные, включая трансляцию запросов // Программирование, 1999, № 5, с. 12 - 45.
271. Стогний А.А., Калиниченко Л.А. 5-я Всесоюзная конференция по базам данных и знаний // Управляющие системы и машины, 1991, № 7, с. 3 - 4.
272. Столин В.В. Познание себя и отношение к себе в структуре самосознания личности: Дис. ... д-ра псих. наук. - М., 1985.
273. Стоунс Э. Психопедагогика. Психологическая теория и практика обучения. - М.: Педагогика, 1984. - 470 с.
274. Тарануха В.М. Разработка и исследование принципов и методов организации вычислений на основе параллельной вертикальной арифметики: Автореферат дис. ... д-ра техн. наук. - Таганрог, 2000.
275. Тарануха В.М. Теоретические основы и принципы построения вычислительных средств параллельной вертикальной арифметики. - Таганрог, Изд-во "Таганрог", 1996. - 140 с.
276. Тарасюк И.В. Понятия эквивалентностей для разработки параллельных систем с использованием сетей Петри // Программирование, 1998, № 4, с. 19 - 39.
277. Таубе М. Вычислительные машины и здравый смысл. - М.: Мир, 1964.
278. Телерман В.В., Ушаков Д.М., Швецов И.Е. Уровни параллельности в задачах удовлетворения ограничений с недоопределенными объектами // Программирование, 1998, № 6, с. 60 - 69.
279. Тербер К. Дж. Архитектура высокопроизводительных вычислительных систем / Пер. с англ. - М.: Наука, 1985.
280. Технология разработки экспертных систем // Тез. докл. респ. школы-семинара. - Кишинев, 1987.
281. Тимирязев К.А. Соч. В 5-ти т. - М., 1938.
282. Тимофеев-Ресовский Н.В., Воронцов Н.Н., Яблоков А.В. Краткий очерк теории эволюции. - М.: Наука, 1977.
283. Тимофеев-Ресовский Н.В., Яблоков А.В., Глотов Н.В. Очерк учения о популяции. - М.: Наука, 1973.
284. Тиори Т., Фрай Дж. Проектирование структур баз данных. - М.: Мир, 1985.
285. Трахтенгерц Э.А. Введение в теорию анализа и распараллеливания программ ЭВМ. - М.: Наука, 1981.
286. Трахтенгерц Э.А. Программное обеспечение параллельных процессов. - М.: Наука, 1981.
287. Труды Юбилейной научно-технической конференции, посвященной 30-летию образования ЦНИИРЭС, 12-14 сентября 2001 г. "Прогрессивные направления развития радиоэлектронных комплексов и систем". В 2 частях. - М.: Изд-во ЦНИИРЭС, 2001.
288. Турчин В.Ф. Феномен науки: Кибернетический подход к эволюции. - М.: Наука, 1993.
289. Тыгу Э.Х. Объектно-ориентированное программирование // Программирование, 1990, № 6, с. 16 - 26.
290. Уемов А.И. Вещи, свойства и отношения. - М.: Изд-во АН СССР, 1963.
291. Узнадзе Д.Н. Психологические исследования. - М., 1966.
292. Уинстон П. Искусственный интеллект. - М.: Мир, 1980.
293. Ульман Ж. Основы систем баз данных. - М.: Финансы и статистика, 1983.

294. Уоссермен Ф. Нейрокомпьютерная техника. - М.: Мир, 1992.
295. Уотсон Дж. Молекулярная биология гена. - М.: Мир, 1978.
296. Уэлдон Дж. - Л. Администрирование баз данных / Пер. с англ.; Предисл. Будзко В.И. - М.: Финансы и статистика, 1984. - 207 с.
297. Федосеев П.Н. Философия и научное познание. - М.: Наука, 1983.
298. Фейгенбаум Э., Фельдман Дж. Вычислительные машины и мышление. - М.: Мир, 1967.
299. Фейнман Р., Лэйтон Р., Сэндс М. Фейнмановские лекции по физике. Вып. 3. - М.: Мир, 1965.
300. Филиппов В.И., Пржиялковский В.В., Шкотин А.В. Инструментальные средства информационного моделирования // Управляющие системы и машины, 1991, № 7.
301. Филипченко Ю.А. Эволюционная идея в биологии. - М.: Наука, 1977.
302. Философия. Методология. Наука. - М., 1972.
303. Философский словарь. - М.: Политиздат, 1987.
304. Финк Д. Вычислительные машины и человеческий разум. - М.: Мир, 1967.
305. Флорес И. Структуры и управление данными / Пер. с англ. Будзко В.И.; Предисл. Савинкова В.М. - М.: Финансы и статистика, 1982. - 319 с.
306. Фогель Л., Оуэнс А., Уолш М. Искусственный интеллект и эволюционное моделирование. - М.: Мир, 1969.
307. Фрир Дж. Построение вычислительных систем на базе перспективных микропроцессоров / Пер. с англ. - М.: Мир, 1990.
308. Фролов А.В. Нахождение и использование ориентированных разрезов реальных графов алгоритмов // Программирование, 1997, № 4, с. 71 -80.
309. Хендерсон П. Функциональное программирование / Пер. с англ. - М.: Мир, 1983.
310. Хоар Ч. Взаимодействующие последовательные процессы / Пер. с англ. - М.: Мир, 1989.
311. Хокни Р., Джессхоуп К. Параллельные ЭВМ. Архитектура, программирование и алгоритмы / Пер. с англ. - М.: Радио и связь, 1986.
312. Холдэн Дж. Факторы эволюции. - М.; Л.: Биомедгиз, 1935.
313. Хэндлер В. Новая архитектура ЭВМ - как увеличить параллелизм, не увеличивая сложности / Пер. с англ. // Системы параллельной обработки. - М.: Мир, 1985.
314. Цаленко М.Ш. Моделирование семантики в базах данных. - М.: Наука. Гл. ред. физ.-мат. лит., 1989. - 288 с.
315. Цаленко М.Ш. Реляционные модели баз данных // В сб.: Алгоритмы и организация решения экономических задач. - М.: Статистика, 1977, вып. 9, с. 18 - 32.
316. Цикритзис Д., Лоховски Ф. Модели данных. - М.: Финансы и статистика, 1985.
317. Чадеев В.М. Теория адаптивных систем управления с идентификатором: Дис. ... д-ра техн. наук. - М., 1997.
318. Черняев В.Г. Распределенные управляющие и вычислительные системы. - М.: Мир, 1987.
319. Четвериков В.Н., Ревунков Г.И., Самохвалов Э.Н. Базы и банки данных. - М.: Высшая школа, 1987.
320. Четвериков С.С. Проблемы общей биологии и генетики (воспоминания, статьи, лекции). - Новосибирск: Наука, 1983.
321. Чечкин А.В. Математическая информатика. - М.: Наука. Гл. ред. физ. - мат. лит., 1991. - 416 с.

322. Чукин Ю.В., Вайнтруб В.М., Гриб О.В., Донец Н.Б., Яблоновский В.А. Концептуальная модель системы управления базой видео-данных // Управляющие системы и машины, 1991, № 2, с. 77 - 87.
323. Шаймарданов Р.Б. Моделирование и автоматизация проектирования структур баз данных. - М.: Радио и связь, 1984.
324. Шалютин С.М. Искусственный интеллект. Гносеологический аспект. - М.: Мысль, 1985.
325. Шенк Р. Обработка концептуальной информации. - М.: Энергия, 1980.
326. Шеремет И.А. Интеллектуальные программные среды для АСОИ. - М.: Наука, 1994.-544с.
327. Шеремет И.А. Об одном формализме для представления знаний // Программирование, 1993, № 2, с. 43 - 57.
328. Шишков Д.П. Архитектура компьютерных систем 5 поколения и компьютеризация естественных языков // Тез. докл. научн. конф. - Таганрог: ТРТУ, 2000. - С. 95-100.
329. Шкловский И.С. Вселенная, жизнь, разум. - М.: Наука, 1976.
330. Шмальгаузен И.И. Кибернетические вопросы биологии. - Новосибирск: Наука, 1968.
331. Шпаковский Г.И. Архитектура параллельных ЭВМ. - М., 1989.
332. Шредингер Э. Что такое жизнь с точки зрения физика. - М.: Атомиздат, 1972.
333. Эволюционные вычисления и генетические алгоритмы. Обзор прикладной и промышленной математики. - М.: Научное издательство "ТВП", том 3, вып. 5, 1996.
334. Эволюция. - М.: Мир, 1981.
335. Эдельман Л.В. Схема реализации СООБД сетевой структуры на базе СОМ // Программные системы и инструменты. Тематический сборник № 1. - М.: Изд-во факультета ВМ и К МГУ, 2000. - С. 81 - 97.
336. Эйнштейн А., Инфельд Л. Эволюция физики. - М., 1965.
337. Эксперт, 1998 - 2001 (<http://www.expert.ru>).
338. Экспертные системы и анализ данных: сборник научных трудов // Под ред. Загоруйко Н.Г. - Новосибирск, 1991.
339. Электронные библиотеки: перспективные методы и технологии, электронные коллекции // Труды Первой Всероссийской научной конференции. - Санкт-Петербург, 19-22 октября 1999г. - СПб.: Изд. Санкт-Петербургского университета, 1999.
340. Эндрю А. Искусственный интеллект. - М.: Мир, 1985.
341. Яблоков А.В. Фенетика: Эволюция, популяция, признак. - М.: Знание, 1980.
342. Яглом И.М. Математические структуры и математическое моделирование. - М.: Сов. радио, 1980.
343. Abramov S.M., Adamowitch A.I., Nesterov I.A., Pimenov S.P., Shevchuck Yu.V. Auto transformation of evaluation network as a basis for automatic dynamic paralleling // Proceedings of NATUG 1993 Meeting "Transputer: Research and Application", May 10 - 11, 1993.
344. Abramov S.M., Nesterov I.A., Shevchuk Yu.V. T-language. Preliminary description, RCMS Tech. Report #09/18/1994.
345. Abrial J.R. Data Semantics. - In: Proc. of IFIP TC-2 Working Conference of Data Base Management, Amsterdam: North-Holland, 1974.
346. Adamovich A.I. cT: an Imperative Language with Parallelizing Features Supporting the Computation Model "Auto transformation of the Evaluation Network" // LNCS #964 1995, Parallel computing technologies: third international conference; proceedings / PaCT-95, St. Petersburg, Russia, September 12 - 25, 1995. Victor Malyshkin (ed), pp. 127-141.

347. Arndt Bode, Mikroprozessoren: Technology and Architecture, CW-Moscow, №35, 1994, c.47-49.
348. Automating knowledge acquisition for expert systems / Ed. by S. Marcus. - Boston, 1988.
349. Banchilhon F. On the Completeness of Query Language for Relational Data Bases. Lect. Notes in Comp., Sci. v. 64, Springer-Verlag, 1978, pp. 76-98.
350. Banchilhon F. ODMG and the ODMG-93 Standard. // Object Currents. 1996, № 5. SIGS Publications, Inc., New York, NY, USA.
351. Banchilhon F. The ODMG Standard: The Object Model. // Object Currents. 1996, № 6. SIGS Publications, Inc., New York, NY, USA.
352. Banchilhon F. Those Persistent Objects. // Object Currents 1996, № 2, SIGS Publications, Inc., New York., NY, USA.
353. Belegradek O.V., Stolboushkin A.P., Taitslin M.A. Extended order-generic queries. Manuscript, 1996.
354. Benedikt M., Dong G., Libkin L., Wong L. Relational expressive power of constraint query languages. Proc. 15th ACM Symp. on Principles of Database Systems. 1996. - P. 5 - 16.
355. Benedikt M., Libkin L. On the structure of queries in constraint query languages. Proc. 11th IEEE Symp. on Logic in Computer Science. Los Alamitos, CA, 1996. IEEE Computer Society Press.
356. Berchtold S., Keim D.A., Kriegel H.-P. The X-tree: an index structure for high-dimensional data. VLDB. 1996. - P. 28 - 39.
357. Boden N., Cohen D., Federman R., Kulawik A., Seitz C., Seizovic J., Su W. Myrinet: a Gigabit-per-Second Local Area Network. IEEE Micro. Feb. 1995.
358. Bodnar Gy. Искусственный интеллект и экспертные системы / ВЦП - № Т-01473, 1988.
359. Buneman P., Winston I. The Use of Data Type Information in an Interactive Database Environment. - In: Proceedings of the Workshop on Data Abstraction, Databases and Conceptual Modelling. - SIGPLAN Notices, 16, 1, 1981, pp. 104 - 106.
360. Burstall R.M., Goguen J.A. The Semantics of Clear, a Specification Language In: Proceedings, of the 1979 Copenhagen Winter School on Abstract Software Specification, Springer-Verlag, Berlin, 1980, pp. 292 - 332.
361. Caandrasekaran B. Блоки высокого уровня для конструирования экспертных систем / ВЦП - № 44489, 1987.
362. Cattell R.G.G., Barry D.K. The Object Database Standard: ODMG 2.0. Morgan Kaufmann Publishers, Inc., 1997.
363. Chen P.P.S. The entity-relationship model - toward a unified view of data. - ACM Trans. on Database Systems, 1976, 1, № 1.
364. CODASYL COBOL Committee Journal of Development 1981, Publishing Center, Canadian Government, 1981.
365. CODASYL Data Descriptor Language Committee Journal of Development 1981, Publishing Center, Canadian Government, 1981.
366. CODASYL End User Facilities Journal of Development 1980, Publishing Center, Canadian Government, 1980.
367. Codd E.F. A relational model of Data for Large hared Data Banks. - ACM, 1970, 13, № 6, pp. 337 - 387.
368. Codd E.F. Extending the Database Relational Model to capture more meaning. - ACM Trans. on Database Systems, 1979, 4, № 4.

369. Cray T3D Systems Journal, vol. 34, no. 2, 1995.
370. Digital Libraries // Comm. of ACM. Special Issue. V. 38. № 4. April 1995.
371. Dolphin Interconnect Solutions, Inc. (<http://www.dolphinics.com>).
372. Ehrig H., Mahr B. Fundamentals of Algebraic Specification. v.1, Springer-Verlag, Berlin, 1985.
373. Faloutsos C., Jagadish H. J., Manolopoulos Y. Analysis of the n-dimensional quadtree decomposition for arbitrary hyperentangles // TKDE. 1997, V. 9, № 3, pp. 373 - 383.
374. Gattell R.G.G., Atwood T., Duhl J., Ferran G., Loomis M., Wade D. The Object Database Standard: ODMG-93. San-Mateo, CA: Morgan Kaufmann, 1993.
375. Gattell R.G.G., Douglas K. Barry. The Object Database Standard: ODMG 2.0. 1997, Morgan Kaufmann, Inc., 1997.
376. Girow A. Limitations of Object Data Models. // Object Currents. 1997, № 1. New York: SIGS Publications, Inc., 1997.
377. Goguen J.A., Thatcher J.W., Wagner E.G. An initial algebra approach to the specification, correctness and implementation of abstract data types. In Current Trends in Programming Methodology 4: Data Structuring, Prentice Hall, 1978, pp. 80 - 144.
378. Gruber T., Cohen P. Проектирование интеллектуальных систем в расчете на автоматизированное усвоение знаний. / ВЦП - № PC-01539, 1990.
379. Grumbach S., Su J. Dense-order constraint databases. Proc. 14th ACM Symp. on Principles of Database Systems. 1995. - P. 66 - 77.
380. Hardgrave W.T. Technique for implementing a set processor. - ACM Signal Notices, 1976, №2.
381. Hayes-Roth Fr. Экспертные системы с базой знаний / ВЦП, 1987.
382. Herrod R.A., Tietz L. Оперативные экспертные системы и их роль в развитии систем управления будущего / ВЦП - № T-24741, 1990.
383. Kahn B.K. A method for Describing the Information Required by the Data Base Design Process, Proc. Int. ACM / SIGMOD Conf. Manag. Data, 1976, pp. 53 - 64.
384. Kalyaev A.V. Multimicroprocessor Systems. Information Processing 86. -Amsterdam. New York. Oxford. Tokyo. -1986, - pp. 949 - 954.
385. Kalyaev A.V. Multiprocessor Systems with Programmable Architecture. Fifth Generation Computer Architectures. -Amsterdam. New York. Oxford. Tokyo.-1986, - pp. 291 - 300.
386. Kalyaev A.V. Ultra-high Performance Multiprocessor Supersystems with Programmable Architecture. Aspects of computation on Asynchronous Parallel Processors. -Amsterdam. New York. Oxford. Tokyo. -1989, - pp. 111 - 123.
387. Kanellakis P.C., Kuper G.M., Revesz P.Z. Constraint query languages // Journal of Computer and System Sciences. 1995, V.51, № 1, pp. 26 - 52.
388. King N.H. Object DBMSs: Now or Never - Will ODBMSs capitalize on new opportunities to manage Web and complex datatypes? // DBMS and Internet Systems. 1997. Miller Freeman, Inc.
389. Linticum D.S. Objects Meet Data. // DBMS online Magazine, № 9, 1996, Miller Freeman, Inc. 1996.
390. Linticum D.S. Reevaluating Distributed Objects. // DBMS online Magazine. 1997. № 1. Miller Freeman, Inc.
391. Linux Journal's Linux Resources (<http://www.ssc.com/linux>).
392. Minsky M. A framework for representing knowledge. - In: The Psychology of Computer Vision. NY: McCraw-Hill, 1975.

393. Nesterov I.A., Suslov I.V., Towards programming of numerical problems within the system providing automatic parallelizing // Proceedings of 7th SIAM conference on parallel processing for scientific computing, p. 716, San-Francisco, CA, 1995.
394. Noakes M.D., Wallach D.A., Dally W.J. The J-Machine Multicomputer: An Architectural Evaluation. Proc. Of the IEEE, 1993.
395. Notley M.G. The Peterlee IS/1 System. IBM (UK) Scientific Center Report, UKSC - 0018, March 1972.
396. Object Database Management Group. Response to the March 1994 ODMG-93 Commentary. SIGMOD Record September 1994.
397. Otto M., Van den Bussche J. First-order queries on databases embedded in an infinite structure. Manuscript, 1995.
398. Paradaens J., Van den Bussche J., Van Gucht D. First-order queries on finite structures over realize. Proc. 10th IEEE Symp. on Logic in Computer Science. IEEE Computer Society Press, 1995. - P. 79 - 87.
399. Reed P. Using ODBC to Access Nontabular Data. // DBMS online Magazine 1996, № 4. Miller Freeman, Inc.
400. Revesz P.Z. Safe stratified Datalog with integer order programs. Manuscript, August 1995.
401. Senko M.E. Data structures and accessing in database systems. - IBM Systems J., 1973, v.12, № 1, pp. 161 - 169.
402. Senko M.E. DIAM as a detailed example of ANSI/SPARC Architecture. - In. proc. of IFIP TC-2 Working Conf. on Modelling in data base management systems. Amsterdam: North-Holland, 1976.
403. Smith J.M., Smith D.C.P. Principles of Database Design. - NJU Symp. Database Design, May 1978, pp. 35 - 40.
404. SP - суперкомпьютеры семейства RS/6000: технологии, архитектура, применение / CW-Москва, 1997, № 14-16, сс.1, 43 - 46, 56, 38, 39, 50.
405. SPEC (<http://www.specbench.org>).
406. TCP (<http://www.tcp.org>).
407. Wade A.E. Ph. D. Queries on an ODBMS? Objectivity Inc. For Object Magazine, 1996.
408. Wade D. ODMG-93 The Object Database Standard. Objectivity, Inc., 20-06-96.
409. Walker A. Экспертные системы с базами знаний: принципы и практика / ВЦП - № М-35808, 1991.
410. Weis R., Third-generation RISC processors, EDN, - 1993, - Vol. 37, pp. 96 - 108.
411. Won Kim. Observations on the ODMG-93 Proposal. SIGMOD Record March 1994.
412. Zabrodin A.V., Levin V.K., Korneev V.V. The massively parallel computer system MBC-100. - Lecture Notes in Computer Science, N 964. Parallel Computing Technologies. Third International Conference, PaCT-95, St. Petersburg, Russia, Sept. 1995, Springer.
413. Zilles S.N. Introduction to data algebra's. Lect. Notes Computing Sci., v.86, Springer-Verlag, 1980.
414. Клаус Г. Кибернетика и философия. - М.: Изд-во ин. лит-ра, 1963.
415. Розов М.А. О системности интеллектуальных систем. Тезисы докладов и сообщений к 6-й научно. - метод. конференции "Интеллектуальные системы и имитация", 28-30 окт. 1985г., - Новосибирск, 1985.
416. Гаврилов Г.П., Сапоженко А.А. Сборник задач по дискретной математике. - М.: Наука, 1977.
417. Евстигнеев В.А. Применение теории графов в программировании // Под ред. Ершова А.П. - М.: Наука, 1985.

418. Зыков А.А. Основы теории графов. - М.: Наука, 1987.
419. Кристофидес Н. Теория графов. Алгоритмический подход / Пер. с англ. - М.: Мир, 1978.
420. Лекции по теории графов: Учеб. пособие. / Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. - М.: Наука, 1990.
421. Майника Э. Алгоритмы оптимизации на сетях и графах / Пер. с англ. - М.: Мир, 1981.
422. Яблонский С.В. Введение в дискретную математику. - М.: Наука, 1979.
423. Asano T. An approach to the homomorphism problem and max-cut problem. IEEE, Symposium on circuit theory proceeding of the IEEE. Kyoto, Japan, 1985, vol. 3, pp. 1657-1660.
424. Seymour P.D. On odd cuts and plane multicommodity flows. Proceedings of London Mathematical Society, ser. 3, vol.42, London, 1981, pp.178-192.
425. Yannakakis M. Et al. Cutting and partitioning a graph after a fixed pattern. Lecture Notes Computers Science, USA, 1983, pp. 712-722.
426. Yannakakis M. Node-deletion problems on bipartite graphs. - SIAM Journal Computer, 1981, vol. 10, N 2, pp. 310-327.
427. Адамова Л.Е., Варламов О.О. Влияние психологических аспектов самоотношения при разработке систем искусственного интеллекта // Интеллектуальные и многопроцессорные системы - 2001. Тез. докл. Международной научной конференции. Таганрог: Изд-во ТРТУ, 2001. С. 24-27.
428. Варламов О.О. Проблема адаптивного синтеза конфигураций интеллектуальных многопроцессорных систем // Интеллектуальные и многопроцессорные системы - 2001. Тез. докл. Международной научной конференции. Таганрог: ТРТУ, 2001. С. 114-117.
429. Варламов О.О., Адамова Л.Е. Обеспечение адаптивности и активности логического вывода на эволюционной сети // Научная сессия МИФИ-2002. Сборник научных трудов. В 14 томах. Т.2. Программное обеспечение. Информационные технологии. - М.: Изд-во МИФИ, 2002. С. 99 - 101.
430. Варламов О.О., Адамова Л.Е. Создание эволюционных баз данных и знаний // Научная сессия МИФИ-2002. Сборник научных трудов. В 14 томах. Т.2. Программное обеспечение. Информационные технологии. - М.: Изд-во МИФИ, 2002. С. 101 - 103.
431. Реан А.А., Коломинский Я.Л. Социальная педагогическая психология. – СПб.: Питер, 1999.
432. Реан А.А. Рефлексивно-перцептивный анализ деятельности педагога // Вопросы психологии, 1990, № 2, с. 77-82.
433. Якунин В. А. Обучение как процесс управления. – Л., 1988.
434. Практическая психология. Учебник / Под ред. Тутушкиной М.К. - М., СПб., 1997.
435. Хекхаузен Х. Мотивация и деятельность. В 2-х томах. - М., 1986.
436. Орлов А. Б. Экспериментальные и прикладные исследования мотивационных образований в зарубежной когнитивной психологии. Псих. журнал. т. 11, № 6, 1990.
437. Платонов К.К. Структура и развитие личности. - М.: Наука, 1986.
438. Научная сессия МИФИ-2002. Сб. научных трудов. В 14 томах. - М.: Изд-во МИФИ, 2002.
439. Асеев В.Г. Мотивация поведения и формирования личности. М.: Мысль, 1976.
440. Бодалев А.А. Психология о личности. М.: Изд. МГУ, 1998.

441. Выготский Л.С. Собрание сочинений в 6 т.: Т.4. М., 1984.
442. Додонов Б.И. Структура и динамика мотивов деятельности // Вопросы психологии, 1984, №4, с.126-130.
443. Дюк В.А. Компьютерная психодиагностика. СПб., 1994.
444. Зимняя И.А. Педагогическая психология. Уч. пособие. - Ростов-на-Дону: "Феникс", 1997.
445. Ковалев В.И. Мотивы поведения и деятельности. - М.: Наука, 1988.
446. Котова И.Б. Психология личности в России. Столетие развития. - Ростов-на-Дону, 1994.
447. Леонтьев А.Н. Деятельность. Сознание. Личность. М.: Политиздат, 1997.
448. Леонтьев А.Н. Лекции по общей психологии. - М.: Смысл, 2000.
449. Общая психодиагностика / Под. ред. А.А. Бодалёва, В.В. Столина. - М., 1987.
450. Пантिलеев С.Р. Самоотношение как эмоционально-оценочная система. - М.: МГУ, 1994.
451. Пантилеев С.Р., Зимачева Е.М. Способы вербальной презентации образа «Я» и самоотношения субъекта // Психологическое образование, 1997, № 2(5).
452. Пантилеев С.Р., Столин В.В. Методика исследования самоотношения // Вестник МГУ. Серия 14, Психология. 1989, №1.
453. Платонов К.К. Краткий словарь системы психологических понятий. Изд. 2е пер. и доп. - М.: Высшая школа, 1984.
454. Психология личности. В 2-х томах. - Самара, 1999.
455. Реан А.А. Психодиагностика личности в педагогическом процессе. - М., 1996.
456. Рубинштейн С.Л. Основы общей психологии. В 2х томах. - М.: "Педагогика", 1989.
457. Столин В.В. Проблема самосознания личности с позиций теории деятельности. А.Н. Леонтьева. - М., 1983.
458. Столин В.В. Самосознание личности. М., 1983.
459. Столин В.В., Пантилеев С.Р. Опросник самоотношения // Практикум по психодиагностике: Психодиагностические материалы. - М., 1988.
460. Захарченко А.Н. Обработка информации в комплексных системах мониторинга. - М., 2001. - 147 с.
461. Гусева Т.И., Башин Ю.Б. Проектирование баз данных в примерах и задачах. - М.: Радио и связь, 1992. - 160 с.
462. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. - СПб.: Питер, - 384 с.
463. Гаврилова Т.А., Котова Е.Е., Писарев А.С. Активные схемы как инструмент семантического анализа // Труды межд. семинара "Диалог 99". - Таруса, 1999. С. 26-27.
464. Гаврилова Т.А., Лещев И.А. ВИКОНТ: Визуальный Конструктор ОНТологий для структурирования семантической информации // Труды Первой Всерос. науч. конф. "Электронные библиотеки: Перспективные методы и технологии, электронные коллекции". - СПб., 1999. С. 97-98.
465. Хант Э. Искусственный интеллект. - М.: Мир, 1978. - 558 с.
466. Калинин В.Н., Резников Б.А., Варакин Е.И. Теория систем и оптимального управления. Ч. 2. Понятия, модели, методы и алгоритмы оптимального выбора. - Л., 1987. - 589 с.
467. Шахтарин Б.И. Анализ систем синхронизации при наличии помех. - М., 1996. - 252 с.

468. Шахтарин Б.И. Статистическая динамика систем синхронизации. - М.: Радио и связь, 1998. - 488 с.
469. Семь нот менеджмента. 3-е изд. - М.: ЗАО "Журнал Эксперт", 1998. - 424 с.
470. Котлер Ф. Основы маркетинга. - М.: Прогресс, 1991. - 736 с.
471. Имери В. Как сделать бизнес в Internet. 3-е изд. - К.; М.; СПб.: Диалектика, 1998. - 464 с.
472. Эшби У.Р. Введение в кибернетику / Пер. с англ. - М.: Изд-во иностранной литературы, 1959. - 432 с.
473. Кузин Л.Т. Основы кибернетики. Т. 1. Математические основы кибернетики. Учеб. пособие для студентов вузов. - М.: Энергия, 1973. - 504 с.
474. Кузин Л.Т. Основы кибернетики: В 2-х т. Т. 2. Основы кибернетических моделей. Учеб. пособие для вузов. - М.: Энергия, 1979. - 584 с.
475. Джордж Ф. Основы кибернетики / Пер. с англ. - М.: Радио и связь, 1984. - 272 с.
476. Труевцев К.М. Партия для парламента и партия президента (закономерности развития многопартийности и избирательного процесса в России). - М.: ГУ-ВШЭ, 2001.
477. Водолазкий В.В., Колядов А.Н. Путь к LINUX. Изд. второе. - М.: Нолидж, 2001. - 560 с.
478. Пашков Ю.Д., Казеннов В.Н. Организация защиты информации от несанкционированного доступа в автоматизированных системах. - СПб. 1995. - 76 с.
479. Расторгуев С.П. Программные методы защиты информации в компьютерах и сетях. - М.: "Яхтсмен", 1993. - 188 с.
480. Фомин Я.А., Савич А.В. Оптимизация распознающих систем. - М.: Машиностроение, 1993.
481. Экспертные системы. Принципы работы и примеры / Пер. с англ. // Под ред. Р. Форсайта. - М.: Радио и связь, 1987. - 224 с.
482. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ / Пер. с англ. - М.: Финансы и статистика, 1990. - 320 с.
483. Уотермен Д. Руководство по экспертным системам / Пер. с англ. - М.: Мир, 1989. - 388 с.
484. Экспертные системы. Базы знаний и данных. Материалы семинара // Под рук. Э.В. Попова. - М.: ЦРДЖ, 1992. - 171 с.
485. Кирсанов Б.С., Попов Э.В. Отечественные оболочки экспертных систем // Справочник по искусственному интеллекту. Т. 1. - М.: Радио и связь, 1990. С. 369-388.
486. Макалистер Дж. Искусственный интеллект и ПРОЛОГ на микроЭВМ. - М.: Машиностроение, 1990.
487. Осипов Г.С. Приобретение знаний интеллектуальными системами. - М.: Наука, 1997.
488. Попов Э.В., Фоминых И.Б., Кисель Е.Б., Шапот М.Д. Статистические и динамические экспертные системы. - М.: Финансы и статистика, 1996.
489. Попов Э.В. Динамические интеллектуальные системы в управлении и моделировании. - М.: МИФИ, 1996.
490. Яшин А.М. Разработка экспертных систем. - Л.: ЛПИ, 1990.
491. Справочник по искусственному интеллекту в 3-х т. // Под ред. Э.В. Попова, Д.А. Поспелова. - М.: Радио и связь, 1990.

492. Кошелев С.А. Программирование без программирования. Использование утилиты Designer пакета Clarion для разработки программ. - М.: Радио и связь, 1992. - 192 с.
493. Герентьев И.М., Грицаенко О.В. CLARION Professional Developer. - М.: "Малип", 1993.
494. Герентьев И.М. CLARION 2.1 для пользователя и программиста. - М.: Малип, 1995. - 240 с.
495. Лима Т. Введение в dBASE IV / Пер. с англ. - М.: Радио и связь, 1993. - 304 с.
496. Капп Д., Лебен Дж. Техника программирования для IMS. Методология использования DL/1 / Пер. с англ. - М.: Финансы и статистика, 1983. - 248 с.
497. Каррабис Дж.-Д. Программирование в dBASE III Plus/ Пер. с англ. - М.: Финансы и статистика, 1991. - 240 с.
498. Справочник по программированию в системе PARADOX. Язык PAL. - М., 1992. - 172 с.
499. Гилула М.М. Множественная модель данных в информационных системах. - М.: Наука, 1992. - 208 с.
500. Райли Д. Абстракция и структуры данных: Вводный курс. - М.: Мир, 1993. - 752 с.
501. Берзтисс А.Т. Структуры данных / Пер. с англ. - М.: Статистика, 1974. - 408 с.
502. Громов Г.Р. Очерки информационной технологии. - М.: ИнфоАрт, 1993. - 336 с.
503. Гаффин А. Путеводитель по глобальной компьютерной сети Internet. - М.: Сфера, 1995.
504. Markov A.S., Kernozitsky V.A. Economically effective data bases diagnostics method // Advances in Modeling & Analysis. - 1995. - В, Vol.33, № 3. Pp. 5-11.
505. Марков А.С., Керножицкий А.В. Программирование в среде Microsoft Access 2.0: Уч. пособие // Под ред. А.С. Маркова. - СПб.: БГТУ, 1996. - 64 с.
506. Марков А.С., Кильчик С.М., Назаров С.В. Пакеты программ офисного назначения // Под ред. С.В. Назарова. - М.: Финансы и статистика, 1997. - 320 с.
507. Марков А.С., Карпов Е.А., Котухов М.М. и др. Законодательно-правовое и организационно-техническое обеспечение информационной безопасности АС и ИВС // Под ред. И.В. Котенко. - СПб.: ВУС, 2000. - 190 с.
508. Марков А.С. Решение вычислительной задачи при наличии временных ограничений // Изв. вузов. Приборостроение. - 1993. Т.36, № 7-8. С. 89-92.
509. Марков А.С. Как расширить возможности СУБД CLARION // Библиотека информационных технологий: Вып. 10 // Под ред. Г.Р. Громова. - М.: Инфоарт, 1994. - С. 3- 10.

Адрес для связи с автором: 101000, г. Москва, Главпочтамт, До востребования,  
Варламову Олегу Олеговичу

Научное издание

**Варламов Олег Олегович**

**"Эволюционные базы данных и знаний для адаптивного синтеза  
интеллектуальных систем. Миварное информационное пространство"**

Научный редактор Г.О. Крылов  
Технический редактор  
Корректор

Компьютерный набор и верстка - О.О. Варламов

ИБ № 3069

---

Сдано в набор 17.01.2002г.  
Формат 60x80 1/16  
Печать офсетная  
Тираж 1000 экз.

Бумага офсетная.  
Усл. печ. л. 40,1.

Подписано в печать 29.04.2002г.  
Гарнитура Таймс  
Уч.-изд. л. 35,7  
Заказ № \_\_\_\_

Издательство "Радио и связь"  
103473, г. Москва, 2-й Щемилковский пер., д. 4/5.  
Издательская лицензия № 010164 от 29.01.97 г.

---

Отпечатано с оригинал-макета  
в типографии