

# **ИНФОРМАЦИОННЫЕ СИСТЕМЫ В МЕТАЛЛУРГИИ**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ – УПИ

**Н.А.Спирин, В.В.Лавров**

# **ИНФОРМАЦИОННЫЕ СИСТЕМЫ В МЕТАЛЛУРГИИ**

**Конспект лекций**

**(отдельные главы из учебника для вузов)**

*Рекомендовано Учебно-методическим объединением по образованию в области металлургии в качестве учебника для студентов высших учебных заведений, обучающихся по направлениям 651300 – Металлургия (дипломированные специалисты) и 550500 – Металлургия (бакалавры)*

*ЕКАТЕРИНБУРГ  
2004*

**УДК 669.162.2.012–52(075.8)**

**ББК 34.3:32.9я7**

**И74**

*Рецензенты:*

**Кафедра автоматизированных систем управления Московского института стали и сплавов** (заведующий кафедрой, заслуженный деятель науки и техники РФ, лауреат Государственной и Правительственной премий СССР, профессор, доктор технических наук **А.Г.Дьячко**);

Заведующий кафедрой «Автоматика и управление в технических системах» Уральского государственного технического университета – УПИ, заслуженный деятель науки и техники РФ, лауреат премии Правительства РФ, профессор, доктор технических наук **В.Г.Лисиенко**;

Заведующий лабораторией пирометаллургии восстановительных процессов Института металлургии УрО РАН, доктор технических наук **А.Н.Дмитриев**

**И74 Информационные системы в металлургии:** Конспект лекций (отдельные главы из учебника для вузов) / **Н.А.Спирин, В.В.Лавров.** Екатеринбург: Уральский государственный технический университет – УПИ, 2004. – 495 с.

ISBN 5–321–00061–1

Рассмотрены вопросы структуры информационных систем технологических процессов. Основное внимание уделено рассмотрению основных информационных технологий в металлургии. Процедура всех информационных технологий последовательно проиллюстрирована на примере доменного производства.

Учебник предназначен для студентов, обучающихся по специальностям металлургического профиля, может быть полезен студентам других специальностей политехнических вузов.

Библиогр.: 61 назв. Табл. 11. Рис. 156.

ISBN 5–321–00061–1

© Уральский государственный  
технический университет – УПИ, 2004

© Авторы, 2004

---

## ОГЛАВЛЕНИЕ

Предисловие .....	4
ГЛАВА 1. ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ .....	1-1
ГЛАВА 2. АРХИТЕКТУРА СОВРЕМЕННЫХ ПРОГРАММНЫХ СРЕДСТВ .....	2-1
ГЛАВА 3. ОСНОВЫ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ...	3-1
ГЛАВА 4. ИНФОРМАЦИОННЫЕ СЕТЕВЫЕ ТЕХНОЛОГИИ .....	4-1
ГЛАВА 5. АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ .....	5-1
ГЛАВА 6. ПРИНЦИПЫ ПОСТРОЕНИЯ И РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ (НА ПРИМЕРЕ ДОМЕННОГО ПРОИЗВОДСТВА) .....	6-1
ГЛАВА 7. ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ .....	7-1
ГЛАВА 8. ЭКСПЕРТНЫЕ СИСТЕМЫ В ДОМЕННОМ ПРОИЗВОДСТВЕ .....	8-1
Рекомендуемый список литературы .....	9-1

## ПРЕДИСЛОВИЕ

Отечественный и зарубежный опыт убедительно доказывает, что развитие предприятий металлургического комплекса, решение проблем качества и конкурентоспособности металлопродукции на мировом рынке требуют коренного совершенствования систем сбора, хранения, обработки передачи и использования информации.

Современные тенденции развития металлургии характеризуются разработкой, внедрением и широким использованием информационных систем и технологий. Основой информационных технологий и систем являются компьютеры и компьютерные сети, имеющие богатейшее программное обеспечение, а также системы управления базами данных, компьютерные системы поддержки принятия решений, методологической основой которых являются теория систем и системный анализ, теория моделирования данных, технологических процессов и знаний.

Принципиальное отличие информационных технологий и систем от других технических дисциплин заключается в том, что их предметная область изменяется весьма и весьма динамично. При этом современные информационные технологии основаны на применении программных продуктов, как правило, обладающих широкими функциональными возможностями и развитым пользовательским интерфейсом. Это зачастую порождает иллюзию необязательности серьезной подготовки в области информационных систем.

Однако практика показывает, что сегодня наиболее ценны те специалисты, которые обладают и успешно совмещают на практике фундаментальные знания в области информатики, математического моделирования с пониманием физики, существа явлений, свойственным металлургическим технологиям. В равной степени это положение справедливо для проектирования систем управления базами данных, для создания компьютерных сетей и систем поддержки принятия решений, для разработки новых информационных технологий.

В последние годы сложилась парадоксальная ситуация, когда обилие, и даже избыточность, издаваемых книг, в том числе и учебников по информатике, отдельным программным средствам, не идет ни в какое сравнение с количеством книг по современным информационным системам, используемым в промышленности. Практически отсутствуют не только учебники, но и монографии, отражающие современные достижения в областях проектирования и эксплуатации современных информационных систем в металлургии, химии, промышленности строительных материалов и многих других областях. В то же время в области информатизации в последнее время наметилась определенная тенденция, заключающаяся в том, что наряду с развитием информационных систем управления предприятием существенно повышается интерес в первую очередь к автоматизированным системам управления технологическими процессами, индустриальной автоматизации и Internet-решениями для предприятий. Связано это, прежде всего, с наметившимся подъемом в отечественной промышленности реального сектора экономики и в том числе металлургии. При этом системы управления технологическими процессами ближе к конкретному производству, более привычны и сейчас выходят на новый виток своего развития.

Основная цель данного учебника состоит в том, чтобы сформировать у студента определенное целостное мировоззрение об информационных системах

технологических процессов в металлургии, научить его основам информационной культуры, т.е. целенаправленности и самостоятельности работы с информацией, использования ее для управления технологическим процессом. Учебник сочетает также широту охвата основных областей, проблем и технологий этой новой области знаний с основательным рассмотрением деталей некоторых информационных технологий, находящихся все большее использование в металлургии.

При подготовке учебника широко использовались материалы книг и статей, опубликованных в открытых сайтах Internet, журналах «Современные технологии автоматизации», «Мир компьютерной автоматизации», «Открытые системы», «Системы и датчики», «Приборы и системы управления», «Сталь» и др., посвященные теории и практике разработки и эксплуатации современных информационных систем в промышленности. Однако в книге в значительной степени отражен и собственный опыт разработки, внедрения и эксплуатации современных информационных систем и технологий на ОАО «Магнитогорский металлургический комбинат» (ОАО ММК). Авторы сочли при этом нецелесообразным иллюстрировать примеры современных информационных технологий и систем различных металлургических переделов вследствие большого разнообразия и сложности таких систем и технологий, каждая из которых имеет свои особенности. Процедура всех основных информационных технологий последовательно проиллюстрирована на примере доменного производства. Последнее обусловлено тем, что это производство является основным, наиболее сложным, энерго- и ресурсоемким переделом в современной черной металлургии, при этом на отечественных доменных печах созданы и успешно эксплуатируются современные технические средства информатики и информационные технологии.

Появление данной книги связано с открытием в 1998 году впервые в России в Уральском государственном техническом университете – УПИ (УГТУ-УПИ) новой специальности - Информационные системы в металлургии. Материалы книги лишь отчасти отражают содержание курсов лекций "Системы управления базами данных в металлургии", "Компьютерные сети", "Компьютерные системы поддержки принятия решений", "Операционные системы", "Информационные системы в металлургических технологиях" и другие, которые читают авторы данной книги.

Книга написана совместно преподавателями кафедр «Теплофизика и информатика в металлургии», «Металлургия железа и сплавов» УГТУ-УПИ и сотрудниками центра АСУ ОАО ММК, разрабатывающих и эксплуатирующих современные информационные системы.

По просьбе коллектива авторов в написании главы 7 принимала участие доцент, кандидат технических наук Л.Ю.Гилева, главы 6 – зав. доменной лабораторией центра АСУ ОАО ММК П.Н.Поляков, а главы 4 – ведущий специалист центра АСУ А.Н.Барда. Авторы считают своим долгом выразить благодарность директору Научно-инженерного центра Ассоциации содействия Всемирной лаборатории (НИЦ АСВЛ) (г.Москва), кандидату технических наук В.И.Соловьеву за материалы, представленные в заключительной 10-й главе книги.

Учебник в равной степени может быть полезен студентам, инженерам-технологам и специалистам в области информационных систем. Он может быть рекомендован студентам, обучающимся по направлениям 651300 – Металлургия (дипломированные специалисты) и 550500 – Металлургия (бакалавры) при изучении дисциплины «Информационные технологии в металлургии», а также будет полезен для студентов, обучающихся по специальности 071900 – Информационные системы и технологии. В нем найдут новый материал бакалавры и магистры, аспиранты и ученые.

Создание учебника было бы невозможно без постоянной поддержки со стороны коллег. Выражая им свою благодарность, авторы признательны также коллективу кафедры автоматизированных систем управления Московского института стали и сплавов (заведующий кафедрой, заслуженный деятель науки и техники РФ, лауреат Государственной и Правительственной премий СССР, профессор, доктор технических наук А.Г.Дьячко); заведующему кафедрой автоматики и управления в технических системах Уральского государственного технического университета–УПИ, заслуженному деятелю науки и техники РФ, лауреату премии Правительства РФ, профессору, доктору технических наук В.Г.Лисиенко и доктору технических наук, заведующему лабораторией пирометаллургии восстановительных процессов Института металлургии УрО РАН А.Н.Дмитриеву за ценные замечания, сделанные при рецензировании учебника.

Авторы считают своим долгом поблагодарить заслуженного деятеля науки и техники РФ, профессора УГТУ-УПИ, доктора технических наук Ю.Г.Ярошенко, директора Уральского филиала ЗАО РТСофт "Средства и системы автоматизации" С.А.Андрианова, генерального директора НПФ ПРОСОФТ–Е «Системы промышленной автоматизации» В.Н.Махова за обсуждение материалов, весьма ценные замечания и критическую оценку рукописи. Учет этих замечаний, несомненно, способствовал улучшению содержания представленного материала.

Авторы осознают, что ряд важных вопросов, относящихся к информационным системам, в силу ограниченности объема книги не могли быть рассмотрены достаточно глубоко и полагают, что с выходом учебника в свет работа над созданием новых учебников в области информационных систем и технологий в металлургии продолжится.

Авторы просят читателей свои отзывы направлять по адресу:  
620002, Екатеринбург, ул. Мира, 19, Уральский государственный технический университет – УПИ;

Мы будем рады узнать ваше мнение!

---

ГЛАВА 1. ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ	
.....	1-2
1.1. ИНФОРМАТИКА – ПРЕДМЕТ И ЗАДАЧИ .....	1-2
1.2. СТРУКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ .....	1-7
1.3. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ .....	1-14
1.3.1. <i>Информационные технологии сбора и обработки информации</i> .....	1-15
1.3.2. <i>Информационные технологии хранения данных. Системы управления базами данных</i> .....	1-23
1.3.3. <i>Модельные системы поддержки принятия решений</i> .....	1-25
1.3.4. <i>Информационная технология экспертных систем</i> .....	1-28
1.3.5. <i>Информационные системы передачи данных (сетевые технологии)</i> .....	1-31
1.3.6. <i>Понятие традиционной и новой информационных технологий</i> .....	1-32
1.4. КОНТРОЛЬНЫЕ ВОПРОСЫ .....	1-32



## Глава 1. ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

### 1.1. Информатика – предмет и задачи

Мир стоит на пороге новой цивилизации. По многочисленным признакам, которые стали все наиболее отчетливо проявляться в последние годы XX века, эта цивилизация будет представлять собой высокоавтоматизированное информационное общество, основанное на широкомасштабном использовании информации и научных знаний. Информация станет главным ресурсом развития человека и общества в XXI веке, а новые информационные технологии будут являться не только средством, но и катализатором этого развития.

Об информации сегодня говорят, как о стратегическом ресурсе общества, определяющем уровень развития государства, его экономический потенциал и положение в мировом сообществе. Так, объем затрат на развитие информационной сферы в США сегодня превышает затраты на развитие топливно-энергетического комплекса этой страны, а прибыль от этой сферы больше, чем от сферы развлечений и досуга.

Специалисты считают, что в начале XXI века в цивилизованных странах произойдет смена основной информационной среды. Удельные объемы информации, получаемой обществом по традиционным информационным каналам (радио, телевидение, почта, связь), неуклонно сокращаются, а по компьютерным сетям возрастают, к 2005 году они будут равны между собой, к 2030 году доля информации, получаемая по компьютерным сетям, возрастет до 0,90 – 0,95.

Информатика – достаточно новая научная дисциплина. Своим появлением она обязана развитию глобального процесса информатизации общества, который, в свою очередь, является проявлением общей закономерности развития цивилизации. Термин «информатика» возник во Франции в 60-х годах от французского слова *Informatique*, образованного в результате объединения терминов *Information* (информация) и *Automatique* (автоматика), что выражает ее суть как области науки, занимающейся автоматизированной обработкой информации с помощью ЭВМ. В переводе с французского языка информатика означает «автоматизированная переработка информации». В англоязычных странах этому термину соответствует синоним *Computer Science* (наука о компьютерной технике). В нашей стране подобная трактовка термина «информатика» утвердилась с момента принятия решения в 1983 году на сессии годовичного собрания АН СССР об организации нового отделения информатики, вычислительной техники и автоматики. Сегодня в России, несмотря на общеизвестные проблемы, связанные, в основном, с экономикой, успешно работает Институт информатики РАН.

Как наука информатика переживает сейчас период своего бурного развития и быстро расширяет свою предметную область. Из технической дисциплины о методах и средствах обработки данных при помощи вычислительной техники информатика превращается в науку об информации и информационных процессах не только в технических системах, но также в природе и обществе. Сегодня вокруг информатики формируется целый комплекс новых направлений научных исследований в области информации. К их числу относятся социальная инфор-

матика (наука о процессах информатизации общества), биологическая информатика (наука об информационных процессах в биологических системах) и др.

Определений информатики как науки очень много, но их обобщение позволяет использовать следующую трактовку, которой мы и будем придерживаться в дальнейшем.

В самом общем понимании **информатика** – наука о наиболее общих закономерностях построения и преобразования информационной модели мира, определяющей роль человека и технических средств в процессах обработки информации в технических, биологических и социальных системах.

В современном понимании информатика представляет собой комплексное научное направление, имеющее междисциплинарный характер. Она активно содействует развитию ряда других научных направлений и тем самым выполняет в определенной степени и интегративную функцию в среде ряда других направлений развития научного знания, как естественнонаучных, так и гуманитарных наук.

Выделение информатики как самостоятельной области человеческой деятельности в первую очередь связано с развитием компьютерной техники. В связи с этим под информатикой иногда понимается область человеческой деятельности, связанная с процессами преобразования информации с помощью компьютеров.

Сегодня информатика – это комплексная инженерная дисциплина, изучающая все аспекты разработки, проектирования, создания, оценки функционирования на ЭВМ систем переработки информации, их применения в различных областях человеческой деятельности.

Отметим сходство и различие между информатикой и кибернетикой.

В связи с этим напомним, что **кибернетика** – это наука об общих принципах управления в различных системах: технических, биологических, социальных и др. Управление немислимо без использования информации, процессы управления неотделимы от информационных процессов. В связи с этим существует довольно распространенное мнение, что информатика является всего лишь одним из направлений кибернетики. Действительно, между информатикой и кибернетикой провести четкую границу не всегда возможно в связи с ее неопределенностью и размытостью. Кибернетика и информатика различаются в расстановке акцентов. В отличие от кибернетики, внимание которой сосредоточено в основном на исследовании систем и процессов управления, главными объектами изучения для информатики являются информационные системы, а также методы и средства хранения, передачи и использования информации в различных условиях, получивших в последние годы обобщенное название **информационной среды**. Информатика занимается изучением процессов преобразования и создания новой информации более широко, практически не решая задачи управления различными объектами, как это делает кибернетика. Информатика появилась благодаря развитию компьютерной техники, базируется на ней и совершенно немислима без нее. Кибернетика же развивается сама по себе, строя различные модели и рассматривая различные принципы и алгоритмы управления, активно используя достижения компьютерной техники.

Прежде чем предпринять какие-то действия, необходимо провести работу по сбору и переработке информации, ее осмыслению и анализу. Отыскание рациональных решений в любой сфере требует обработки больших объемов информации, что подчас невозможно без привлечения специальных технических средств. Так, например, общая сумма знаний менялась вначале очень медленно, но уже с 1900 г. она удваивалась каждые 50 лет, к 1950 г. удвоение происходило каждые 10 лет, к 1970 г. – уже каждые 5 лет, с 1990 г. – ежегодно. Появились противоречия между ограниченными возможностями человека по восприятию и переработке информации и существующими мощными потоками и массивами хранящейся информации.

Мы все являемся свидетелями беспрецедентного в истории человечества и удивительного по своему содержанию социального феномена. Его суть заключается в том, что развитие информационной техники и информационных технологий существенно опережает возможности человека по их эффективному использованию. В связи с этим возникает проблема перестройки всей системы образования. Ведь система образования отстает от фундаментальной науки на 5–10 лет, что при высоких темпах информатизации недопустимо. Сегодня требуется формирование у будущего специалиста нового, информационного мировоззрения, а также существенное повышение информационной культуры.

На любом металлургическом предприятии кроме энергии и сырья используется также информация – жизненно важный продукт, необходимый для того, чтобы быть в курсе всей деятельности предприятия и руководить ею, обеспечивать нормальное функционирование предприятия. Заметим, что объем информации, циркулирующей на крупных металлургических предприятиях, за последние 5 лет возрос примерно на порядок.

Сегодня, несмотря на все трудности, металлургический комплекс Российской Федерации включает около 200 предприятий черной и 130 предприятий цветной металлургии. В комплексе занято около 1,5 млн. человек. В 1998 году по сравнению с 1990 годом объем производства проката составил 54 %, труб 24 %, алюминия 103 %, меди 80 %, никеля 70 %, титана 50 %, при этом потребность проката снизилась в 3,8 раза. Несмотря на все эти изменения предприятия металлургии и сегодня играют важную роль в формировании макроэкономических показателей экономики страны. На них сосредоточено 12,1 % стоимости основных фондов промышленности и занято 9 % производственного потенциала, они производят 14 % промышленной продукции, обеспечивают около 20 % налоговых поступлений промышленности. Доля металлургии в валовом объеме, например, Свердловской области, значительно выше, чем в среднем по стране, за последние 10 лет она возросла с 30 % до 50 %. Огромные производственные мощности России остаются нереализованными.

Главная цель развития металлургического комплекса страны состоит в сохранении и расширении рынков металлопродукции на базе повышения конкурентоспособности продукции. Конкурентоспособность предприятий, при прочих равных условиях (высокие технологии, богатое сырье, ресурсосбережение и экология), будет определяться соответствием качества и объема промышленной информации рациональным технологическим требованиям. Это в равной степе-

ни относится к производству проката из титана на Верхнесалдинском металлургическом производственном объединении, труб на Первоуральском новотрубном заводе, листа на Магнитогорском металлургическом комбинате.

Отечественный и зарубежный опыт убедительно доказывает, что развитие предприятий металлургического комплекса, решение проблемы качества и конкурентоспособности металлопродукции на мировом рынке требуют коренного совершенствования систем сбора, хранения, обработки, передачи и использования информации, используемых как для управления технологическими процессами, так и управления производством в целом.

Как нервные импульсы заставляют работать мышцы, так и информация приводит в движение оборудование и людей на металлургическом предприятии. Информационные потоки, циркулирующие на промышленном предприятии с помощью систем связи хранения и передачи информации, образуют его «нервную систему». Первоочередная задача для крупнейших предприятий металлургии – это создание информационных систем анализа сквозной технологии от добычи руды до получения готовой продукции.

Основное внимание в дальнейшем мы будем уделять использованию информатики на металлургических предприятиях. Главной целью изложения будет являться формирование определенного мировоззрения в информационной сфере и освоении информационной культуры, т.е. умения целенаправленно работать с информацией. Предварительно отметим, что в любой области экономической и социальной деятельности, на любом металлургическом предприятии, занимающемся производством и распределением продукции, создается и используется информация.

Заметим, что под *информацией* обычно понимают сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии.

Система информации существует в естественном виде, она образует информационную среду. При наличии систем автоматики, современной электронной вычислительной техники или даже без них информация существовала и существует, она непрерывно циркулирует. Совокупность информационных потоков и обменов составляет систему информации.

Система информации простирается далеко за пределы предприятия, поскольку она охватывает все данные, имеющие отношение к деятельности этого предприятия и представляющие для него интерес. Таким образом, система информации оказывается связанной с информационной средой и вне данного предприятия.

Несмотря на наличие мощных технических средств, при использовании информации иногда все еще прибегают к кустарным методам – каждая производственная ячейка создает свою собственную небольшую систему информации аналогично тому, как на заре электричества отдельные предприятия располагались вдоль рек, получая энергию от своих турбин. Теперь снабжение энергией производится централизованно – от сетей распределения электричества, газа, тепловой энергии. Так и в сфере информационной деятельности наступила эра автоматизированных информационных центров, которые предусматривают обра-

щение на больших расстояниях к банкам данных и банкам знаний, обеспечивают возможность коллективного использования информационных ресурсов.

Внесло ли это какие-либо изменения в информационную сферу металлургического предприятия? Конечно. Независимо от того, идет ли речь о местном (локальном) устройстве обработки или об удаленном устройстве, используемом несколькими потребителями, необходимость в системе сбора, первичной обработке данных не только не отпадает, а, наоборот, увеличивается в связи с быстрым развитием автоматизированных информационных систем. Опыт массового внедрения в промышленность в 70–90-е гг. вычислительной техники без надежного информационного фундамента привел не только к потере их потенциального эффекта, но и в ряде случаев – к дискредитации на производстве систем управления с применением ЭВМ.

Современные информационные технологии управления характеризуются, прежде всего, широкой компьютеризацией измерительных операций. В компьютерных системах получают распространение многооконный интерфейс, специальные интерпретирующие языки, традиционные операционные системы дополняются рядом новых компонентов в виде прикладных программ, модулей для разработки и отладки программ, обслуживания аппаратных средств для объектно-ориентированной среды управления, совершенствуется человеко-машинный интерфейс в диалоговых информационных системах, широко используются средства телекоммуникаций. Информационные системы создаются с открытой архитектурой для реконфигурации и наращивания дополнительных функций управления. Для передачи и обмена информацией между территориально распределенными подсистемами в условиях мощных промышленных помех применяются беспроводные и оптоволоконные световые линии связи, а хранение информации успешно осуществляется в банках данных.

К эволюции информационных технологий управления в металлургической промышленности относятся замена классических щитов контрольно-измерительных приборов с множеством вторичных приборов и одноконтурных, локальных регуляторов на распределенные компьютерные рабочие модули с цветными дисплеями на базе локальных информационных сетей. Широко внедряются модели самых разных классов и видов, осуществляется диалог промышленного персонала (лица, принимающего решение) с информационной средой, начаты разработки перспективных информационных систем на базе искусственного интеллекта, в частности, – экспертных систем управления технологиями.

Заметим, что объем средств, затрачиваемых на создание и функционирование информационных систем, значителен, очень часто напрямую зависит от размера предприятия и составляет (исходя как из мирового, так и из российского опыта) от одного до трех процентов от объема годового оборота металлургического предприятия.

С точки зрения производства технических средств информатизации и их технологии, мы отстали от передовых развитых стран. А вот с точки зрения создания информационных систем, наши передовые предприятия металлургии могут оказаться впереди этих стран. Действительно, у большинства предприятий

средства на замену появились только во времена появления 386-го процессора. Именно поэтому многие российские предприятия были укомплектованы гораздо лучше аналогичных западных фирм. Тот же путь предстоит пройти и информационным системам на российских предприятиях: сегодня имеется возможность применять такие технические средства информатизации, которые получают на западе распространение только в XXI веке. Здесь справедлива истина – скупой платит дважды. Сегодня необходимо вкладывать средства в передовые информационные технологии.

Для характеристики роли отдельных информационных технологий рассмотрим общую структуру информационной системы.

## 1.2. Структура информационной системы

Сотрудник отдела снабжения и сбыта, подтверждающий получение заказа, технический директор, определяющий техническую политику предприятия, любой металлургический агрегат, «выдающий продукцию», – все это является источником информации.

Последствия поведения или функционирования каждого из них в соответствующей сфере деятельности (заказ, принятие решения о техническом перевооружении, состав получаемого материала и т.п.) порождают то, что принято называть *событием* в самом широком смысле этого слова.

События по-разному отражаются на функционировании предприятия: одни остаются без внимания, другие представляются на усмотрение профессионально компетентных лиц, которые располагают большой свободой действий для локального управляющего воздействия в соответствии с полученной информацией; информация о третьей разновидности событий используется в соответствии с установленными правилами, что в определенный момент времени вызывает *управляющее воздействие* на уровне основных ресурсов, участвующих в рассматриваемом процессе.

Подобно термину «событие», термин «управляющее воздействие» здесь понимается в самом широком смысле: он может означать поставку сырья, изъятие изделия из каталога выпускаемой продукции, составление отчета, перестановку оборудования, ремонт металлургического агрегата, управление технологическим режимом данного конкретного металлургического агрегата и т.п. Таким образом, в общем случае *под управлением понимается совокупность воздействий на систему, переводящих ее в требуемое целевое состояние.*

Эти управляющие воздействия влияют на ход событий и сами в соответствующие моменты времени проявляются через другие события, так что цикл замыкается.

Таким образом, мы имеем здесь дело с тем, что принято называть *системой информации*. Из этой системы мы рассмотрим лишь установившийся информационный цикл (**рис. 1.1**). Событие генерирует информацию, которая затем используется с промежуточной обработкой или без нее; это использование вызывает управляющее воздействие.

С этим процессом мы встречаемся во всех системах управления. Система информации представляет собой информационную среду, которая позволяет определить: где, когда, при каких обстоятельствах произошло событие? Каким образом его воспринять, уловить, зафиксировать? Где, когда и как следует применить управляющее воздействие?

Использование информации подразумевает ряд воздействий, выполняемых искусственно созданной **автоматизированной информационной системой**. Следовательно, автоматизированная информационная система создается человеком искусственным путем. В дальнейшем уточним это понятие.

Рассмотрим основные процессы в информационной системе.

Автоматизированная информационная система в общем случае осуществляет следующие операции (**рис. 1.1**):

- 1) сбор, первичная обработка и оценка достоверности информации;
- 2) преобразование информации, т.е. возможное преобразование информации (перекодирование, перезапись), когда способ представления информации или ее носитель не совместимы с блоком ее использования;
- 3) передача информации в пункт хранения;
- 4) хранение информации;
- 5) возможная вторичная обработка, когда полученную информацию нельзя использовать непосредственно, т.е. когда она в том виде, в каком есть, не может вызвать требуемого управляющего воздействия;
- 6) передача информации и выдача информации пользователю (представление информации);
- 7) компьютерная поддержка принятия решений;
- 8) использование информации лицом, принимающим решение, для осуществления задач управления.

Следует отметить особую роль моделей разного класса и вида на всех этапах. Известный русский академик А.Н. Крылов в 1937 г. еще в период зарождения вычислительной техники и информатики предупреждал, «что если в вычислительное устройство загрузить информационный мусор, то на выходе вычислителя получим также мусор». Последующий опыт показывает, что только использование адекватных моделей разного класса (математических моделей технологических процессов, моделей знаний, моделей данных и т.п.) может обеспечить успешность и эффективность функционирования информационных систем.

В дальнейшем мы будем рассматривать преимущественно автоматизированные информационные системы, а предметом нашего рассмотрения является бурно развивающийся раздел информатики – компьютерная информатика, на наш взгляд, в большинстве случаев неоправданно отождествляемая с самой информатикой.

Под **компьютерной информатикой** мы будем понимать естественно-научную дисциплину, занимающуюся вопросами сбора, обработки, передачи, хранения, отображения и использования информации с применением средств вычислительной техники.





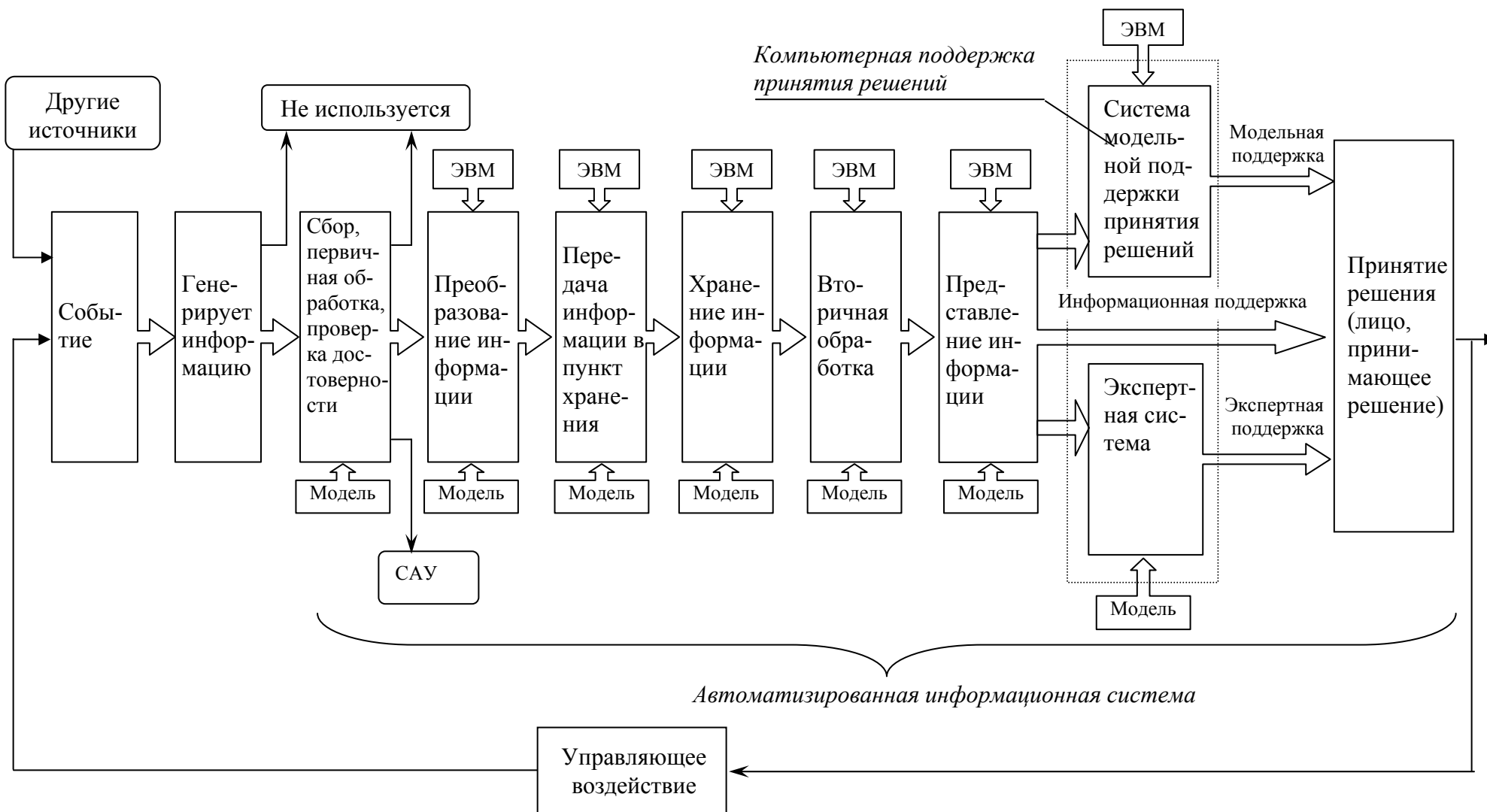


Рис. 1.1. Структура информационной системы

Особо отметим, что на всех этапах, указанных на **рис. 1.1**, современные информационные системы предусматривают широкое использование компьютерной техники.

Конкретизируем понятие автоматизированной информационной системы.

***Автоматизированная информационная система** – это искусственно созданная человеком взаимосвязанная совокупность средств (в том числе и компьютерных), методов и персонала, используемых для получения, хранения, обработки, манипулирования и выдачи информации в интересах достижения поставленной цели.*

Иными словами, информационная система – это человеко-компьютерная система для поддержки принятия решений и производства информационных продуктов, использующая компьютерную информационную технологию.

При всем многообразии информационных систем *главная их цель*, как следует из **рис. 1.1**, одна: *предоставить достоверную информацию в определенное время, определенному лицу, в определенном месте и за определенную плату.*

Отметим различие между компьютерными и информационными системами.

Компьютеры, оснащенные специализированными программными средствами, являются технической базой для информационных систем. Информационная система включает и *персонал*, взаимодействующий с компьютерами.

В силу исключительной сложности большинства металлургических технологий, осуществление управления ими даже при высоком уровне компьютеризации и моделирования металлургических систем без участия человека невозможно. В связи с этим отметим важную роль человека (промышленного персонала, лица, принимающего решение) при окончательном принятии решения в металлургии. Именно поэтому участие человека в современных информационных системах, особенно при принятии решений, как мы покажем в дальнейшем, жизненно необходимо.

Различают *фактографические и документальные информационные системы*.

В *фактографических информационных системах регистрируются факты* – конкретные значения данных об объектах реального мира. Структурированность информации в фактографических информационных системах позволяет им однозначно отвечать на запросы пользователя типа: «Какой объем продукции определенного типа произведен на предприятии за последнюю рабочую смену?», «Каково тепловое состояние доменной печи, оцениваемое по тем или иным параметрам?».

*Документальные информационные системы* обслуживают принципиально иной класс задач, которые не предполагают однозначного ответа на поставленный вопрос. Они оперируют неструктурированными текстовыми документами (статьи, книги, тексты законов и т.д.) и снабжены тем или иным формализованным аппаратом поиска. Цель такой системы, как правило, – выдать в ответ на запрос пользователя список документов, в какой-то мере удовлетворяю-

щих сформулированным в запросе условиям. Например, выдать список всех статей, в которых встречается слово «теплообмен».

Деятельность любого промышленного предприятия, в том числе и металлургического, можно условно разделить на две части: первая – это непосредственно производственный процесс, вторая – финансово-экономическая деятельность предприятия. Требования к информационным системам по финансово-экономической деятельности не имеют, пожалуй, особой специфики для различных областей, но производственная деятельность крупного металлургического производства, включающего множество технологических циклов и потребляющего разное сырье (как исходное, так и промежуточное), всегда ставит задачу контроля технологических цепочек на всех этапах. В металлургии сбои в технологическом цикле могут иметь как тяжелые финансовые последствия, так и приводить к крупным авариям. Соответственно контроль должен осуществляться в реальном времени и непрерывно, что выдвигает требования к производительности информационных систем, гарантии качества услуг и их надежности. Впрочем, надежность и защищенность систем не в меньшей степени требуются и для финансово-экономической деятельности, так как объем входящих и исходящих финансовых потоков, а также циркулирующих внутри предприятия весьма велик.

Любое более-менее серьезное предприятие металлургической отрасли нередко представляет собой конгломерат нескольких, в известной степени независимых друг от друга, но связанных производств. В зависимости от размеров предприятия и области металлургии, в которой оно специализируется, количество этих производств может варьироваться. Относительная автономность всех производств тем не менее подразумевает их слаженную работу и сопряженность технологических циклов. В связи с этим необходимо создание ряда независимых друг от друга информационных систем и обеспечение их интеграционного взаимодействия друг с другом.

Главная цель информационных систем в металлургии – это создание эффективной и надежной информационной структуры анализа сквозной технологии, пригодной к промышленному использованию. Под сквозной технологией в черной металлургии для предприятий с полным металлургическим циклом обычно понимают комплекс технологических операций в системе подготовка руд к плавке – доменный передел – сталь – металлопрокат. В мировой практике принято рассматривать комплексные системы автоматизации предприятий в виде 5-уровневой пирамиды. Структуру информационной системы крупного промышленного предприятия обычно представляют в виде пирамиды (рис. 1.2).

Исторически процесс информатизации проникал на производство с двух сторон – «сверху» и «снизу».

«Сверху» (самый верхний, пятый уровень) в офисах создаются информационные структуры, отвечающие за работу предприятий в целом. Это автоматизация бухгалтерского учета, управления финансами и материально-техническим снабжением, организацией документооборота, анализом и прогнозированием и др. Этот уровень называется *планирование ресурсов производства, т.е. мате-*

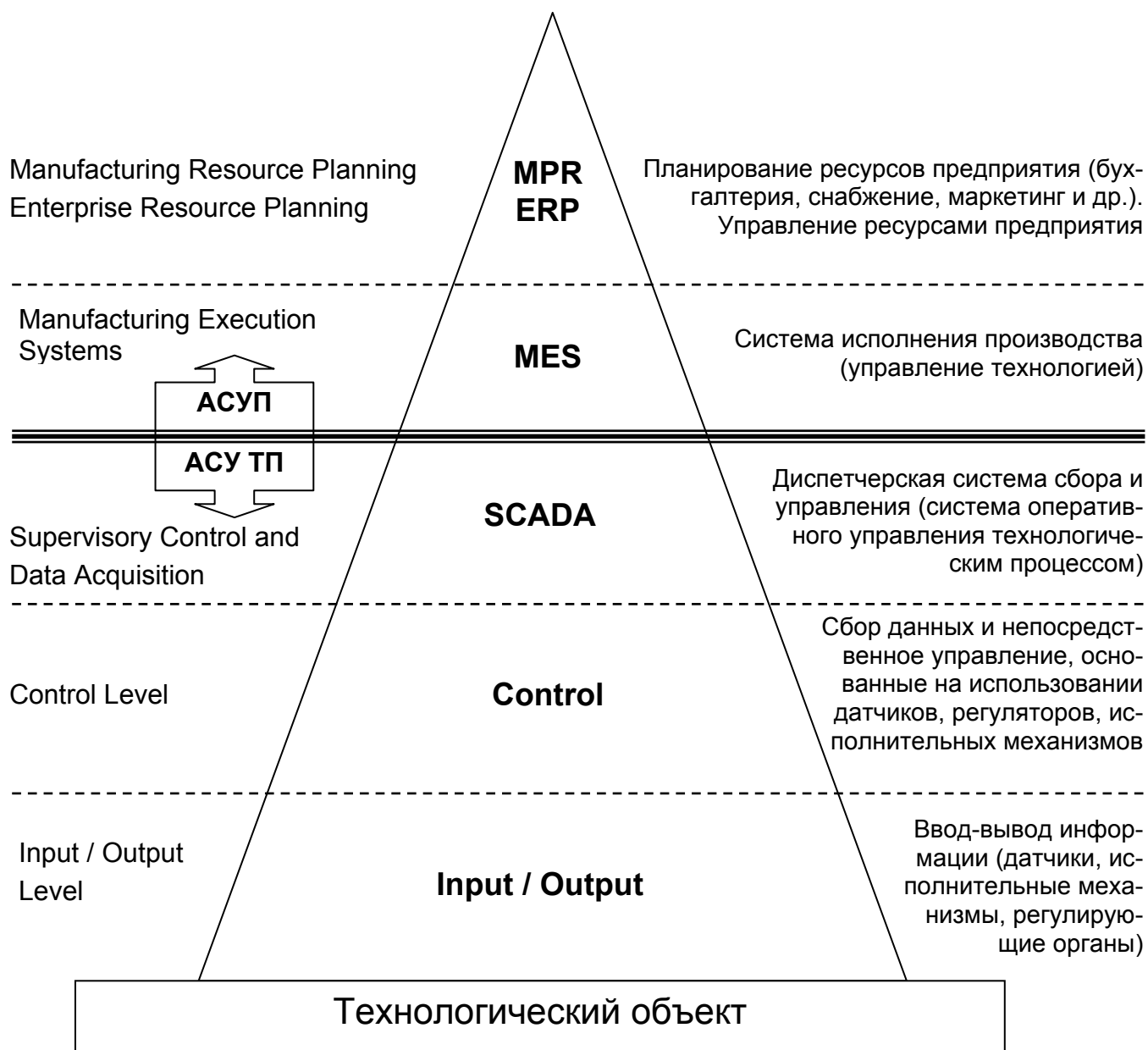


Рис. 1.2. Уровни автоматизированной информационной системы промышленного предприятия

*риальных ресурсов (MRP, Manufacturing Resource Planning), или управление всеми ресурсами предприятия (ERP, Enterprise Resource Planning).*

Самый нижний, первый уровень представляет собой набор датчиков, исполнительных механизмов и других устройств, предназначенных для сбора первичной информации и реализации управляющих воздействий. Этот уровень называется *I/O (Input/Output, ввод/вывод)*.

Следующий, второй уровень предназначен для *непосредственного управления производственным процессом с помощью различных устройств связи с объектом (УСО), программируемых логических контроллеров (ПЛК, PLC – Programmable Logic Controller) или (и) промышленных (индустриальных) компьютеров (PC, ПК)*. Это уровень (*Control Level – простое управле-*

ние), на котором замыкаются самые «короткие» контуры управления производством.

Третий уровень называется *SCADA (Supervisory Control and Data Acquisition – буквально сбор данных и диспетчерское управление)*. На уровне (*SCADA Level*) осуществляется диспетчеризация систем сбора данных и оперативное управление технологическим процессом, принимаются тактические решения, прежде всего направленные на достижение стабильности процесса.

Очевидно, что первичная информация с третьего уровня должна “добираться” до пятого, верхнего уровня, уровня принятия стратегических решений. Очевидно также, что поток сырых данных, без надлежащей обработки, послужит скорее «информационным шумом» для менеджеров и экономистов. Необходимым связующим звеном выступает новый класс средств управления производством – *MES (Manufacturing Execution Systems – или системы исполнения производства)*. Этот уровень выполняет упорядоченную обработку информации о ходе производства продукции в различных цехах, обеспечивает управление качеством, а также является источником необходимой информации в реальном времени для самого верхнего уровня управления.

Заметим, что на каждом из указанных уровней промышленной информационной системы, как мы покажем в дальнейшем, в мировой практике очень широко используются компьютерная техника и различное программное обеспечение.

### 1.3. Информационные технологии

Информационная система по своему составу напоминает предприятие по переработке данных и производству информации. Как и в любом производственном процессе, в информационной системе присутствует технология преобразования данных в другую информацию (**рис. 1.3**).

Под *информационной технологией* будем понимать систему методов и способов сбора, накопления, хранения, поиска, обработки и передачи информации на основе применения средств вычислительной техники.

Цель информационной технологии – производство информации для ее анализа человеком и принятия на его основе решения по выполнению какого-



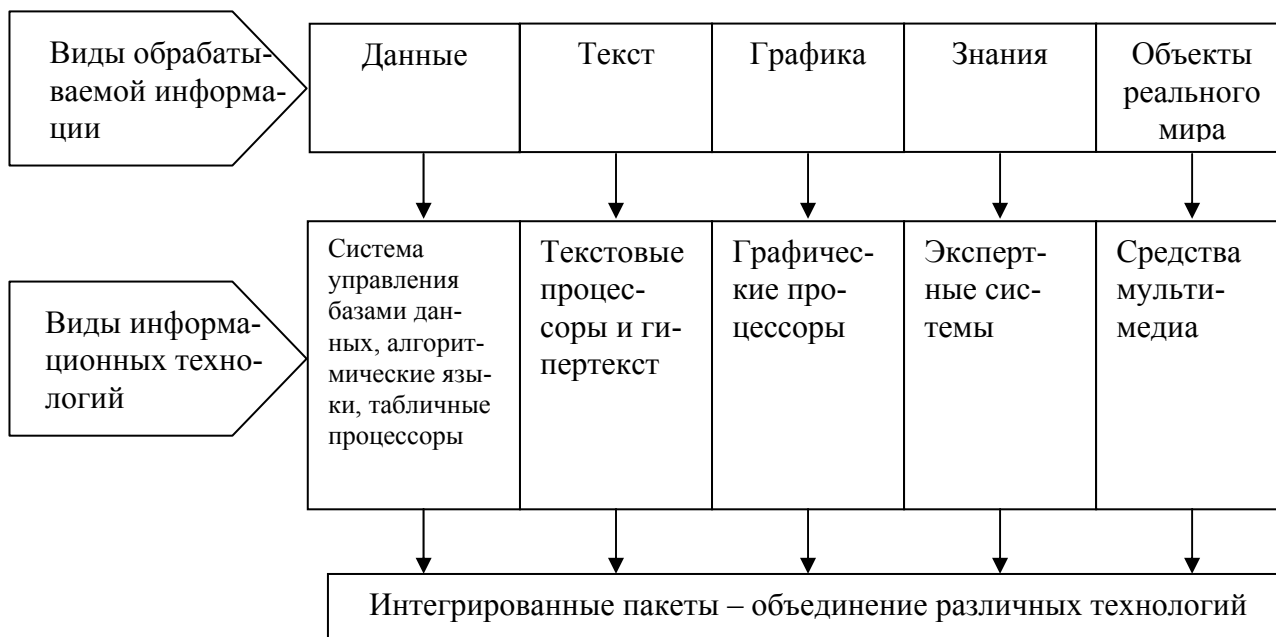
**Рис. 1.3.** Информационная технология как аналог технологии переработки материальных ресурсов

либо действия.

Цель технологии материального производства – выпуск продукции, удовлетворяющей потребности человека или системы.

Информационные технологии, в частности, могут отличаться *по типу обрабатываемой информации* (рис. 1.4), но могут объединяться в интегрированные технологии.

Выделение, предложенное на этом рисунке, в известной мере условно, поскольку большинство этих информационных систем, как мы увидим в даль-



**Рис. 1.4.** Классификация информационных технологий в зависимости от типа обрабатываемой информации

нейшем, позволяет поддерживать и другие виды информации. Так, табличные процессоры могут обрабатывать не только цифровую, но и текстовую информацию, а также обладают встроенным аппаратом генерации графиков. Однако каждая из этих технологий все-таки в большей мере акцентирована на обработку информации определенного вида.

Рассмотрим назначение и дадим краткую характеристику основным типовым информационным технологиям.

### 1.3.1. Информационные технологии сбора и обработки информации

Анализ задач, решаемых на нижнем уровне пирамиды информационной системы (уровень Control, см. рис. 1.2) показывает, что здесь имеется определенная взаимосвязь задач, решаемых информационными системами, с задачами автоматизированных систем управления (АСУ). Особенностью создания цифровой нервной системы промышленного предприятия является необходимость тесной интеграции автоматизированных систем управления технологическими

процессами (АСУТП) и автоматизированными системами управления предприятием (АСУП). В связи с этим конкретизируем эти понятия.

*Автоматизированная система управления (АСУ) – это человеко-машинная система, обеспечивающая автоматизированный сбор и обработку информации, необходимой для оптимизации управления в различных сферах человеческой деятельности. Обычно среди АСУ выделяют автоматизированные системы управления производством (АСУП) и автоматизированные системы управления технологическими процессами (АСУ ТП).*

**АСУП** предназначена для основных задач управления производственно-хозяйственной деятельностью предприятия в целом и (или) его самостоятельных частей на основе применения экономико-математических методов и средств вычислительной техники.

**АСУ ТП** предназначены для выработки и реализации управляющих воздействий на технологический объект управления в соответствии с принятым критерием управления.

В наиболее общем случае АСУ ТП представляет собой замкнутую систему, обеспечивающую автоматизированный сбор и обработку информации, необходимой для оптимизации управления технологическим объектом в соответствии с принятым критерием, и реализацию управляющих воздействий на технологический объект.

Идеология систем управления предприятиями в настоящее время строится по принципу:

- интеграции всех систем управления, и в первую очередь АСУП и АСУ ТП, в единую систему;
- структурированной и модульной организации входящих в ее состав компонентов.

Функции АСУ ТП подразделяются на информационные, управляющие и вспомогательные.

**Информационные функции АСУ ТП** – это функции системы, содержанием которых является сбор, обработка и представление информации для последующей обработки. К информационным функциям АСУ ТП относят централизованный контроль и измерение технологических параметров, вычисление параметров процесса, формирование и выдачу текущих и обобщающих технологических и экономических показателей оперативному персоналу АСУ ТП, подготовку и передачу информации в смежные системы управления.

**К управляющим функциям АСУ ТП** относят регулирование (стабилизацию) отдельных технологических переменных, логическое управление операциями или аппаратами, программное логическое управление оборудованием, оптимальное управление установившимися или переходными режимами или отдельными стадиями процесса, адаптивное управление объектов в целом и др.

**Вспомогательные функции** состоят в обеспечении контроля за состоянием функционирования технических и программных средств системы.

*Как правило, только информационные и вспомогательные функции и являются областью интересов информационных систем, в то время как управ-*

ляющие функции не рассматриваются в информационных системах, а являются прерогативой теории управления, кибернетики и т.д.

В зарубежной практике типовая архитектура АСУ ТП, как правило, включает в себя следующие уровни (см. **рис. 1.2**):

- уровень Input/Output, т.е. непосредственного взаимодействия с технологическим объектом, на котором осуществляется сбор данных от датчиков и воздействие на технологический процесс с помощью исполнительных механизмов и регулирующих органов;
- уровень Control, на котором осуществляется непосредственное управление технологическими параметрами. На этом уровне часто используются программируемые логические контроллеры – ПЛК (PLC – Programmable Logic Controllers) с открытой архитектурой или свободно программируемые контроллеры различных отечественных и зарубежных фирм;
- уровень автоматизированного рабочего места оператора, включающий диспетчерскую систему сбора и управления технологическим процессом (SCADA-система). Это верхний уровень управления в системе АСУТП, на котором собирается необходимая информация от многих источников низшего уровня и который включает контуры управления и принятия решения не только на основе вычислительных средств, но и человека (оператора). На этом же уровне предусматривается решение задач оптимизации, прогнозирования технологического процесса. Здесь предусматривается использование мощных вычислительных ресурсов в экспертных и моделирующих системах реального времени.

Заметим, однако, что в России, чаще всего, первые четыре уровня относят к АСУ ТП, а пятый уровень – к АСУП.

Исторически сложилось так, что самый верхний уровень MRP и нижние три развивались независимо друг от друга и, следовательно, между собой никак не стыковались, т.е. фактически отсутствовал достаточно интеллектуальный интерфейс, который бы их объединял. Это стало тормозящим фактором на современном этапе развития промышленности, когда для эффективной работы производственного предприятия и для принятия на верхнем уровне как стратегических, так и тактических решений требуется его комплексная автоматизация.

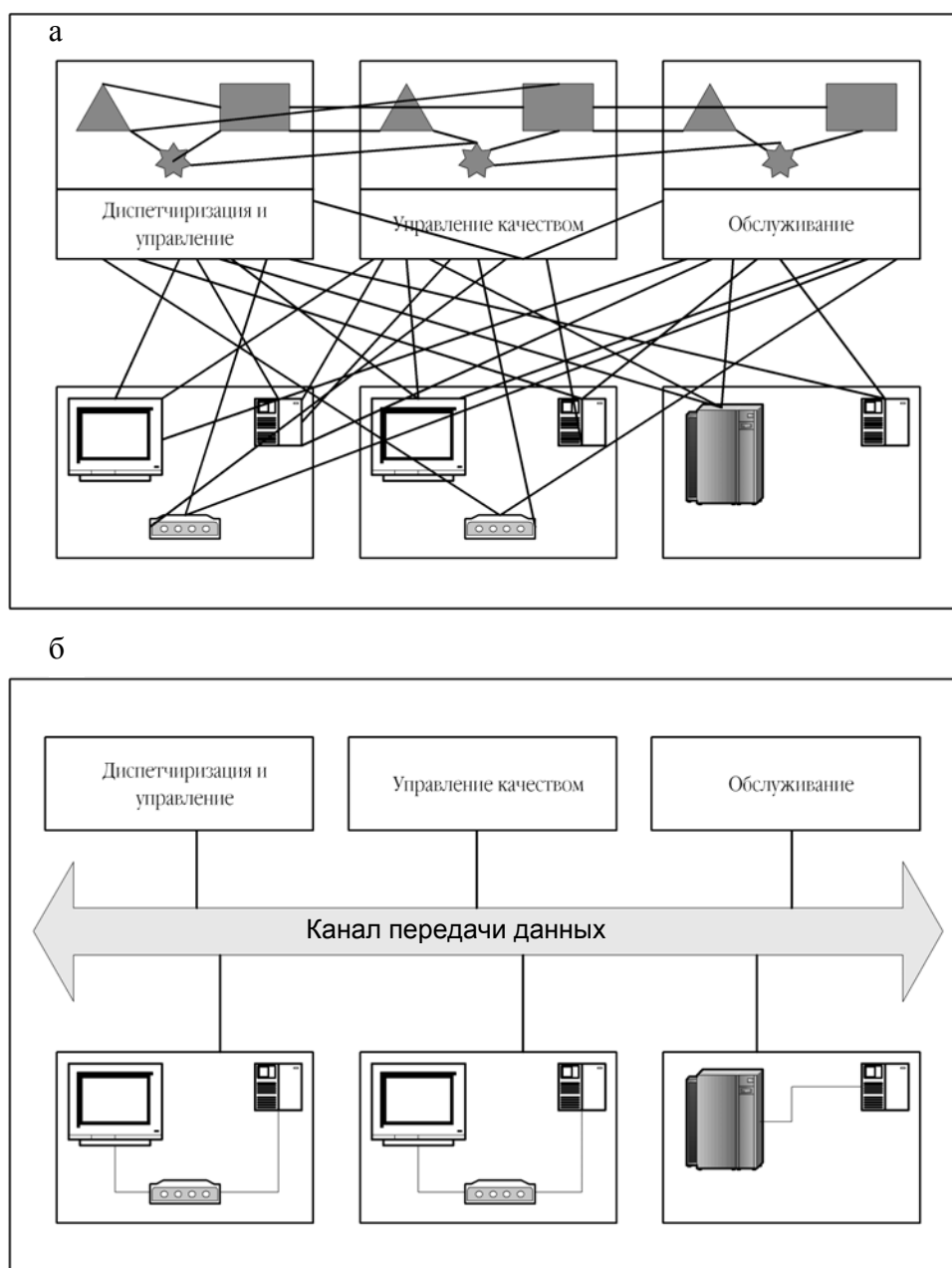
Эволюция структуры управления технологическими процессами от традиционных АСУ ТП к структурированным показана на **рис. 1.5**, который наглядно показывает недостатки традиционного варианта построения АСУ ТП:

- множество интерфейсов, сложность и запутанность связей между объектами;
- несовместимость форматов данных и структуры сообщений;
- как результат – сложность внесения изменений, что может вызвать переработку большого объема программ.

Структурированный (системный) подход обладает следующими особенностями:



- нормализацией данных;
- стандартными формами сообщений;
- гибкими средствами интеграции приложений, включая АСУП.



**Рис. 1.5.** Эволюция структуры управления:  
*а* – традиционный подход; *б* – структурированный подход

Такой модульный систематизированный подход к построению АСУ ТП обеспечивает возможность эффективной модернизации системы, облегчает внесение в нее изменений, что в совокупности гарантирует защиту ранее вложенных инвестиций и уменьшает стоимость информационной системы в целом.

На самом нижнем уровне на вход управляющего вычислительного комплекса от датчиков (термопар, индуктивных датчиков, счетчиков продукции и

др.) поступает измерительная информация о текущих значениях параметров, характеризующих ход технологического процесса (состояние и параметры заготовок, качество обработанных деталей, их количество и др.). Компьютер обрабатывает эту информацию в соответствии с принятым законом управления (алгоритмом управления), определяет управляющие воздействия, которые необходимо приложить к исполнительным механизмам для изменения управляемых параметров, с тем, чтобы управляемый процесс протекал оптимальным образом.

Многие первичные преобразователи вырабатывают свои сигналы в виде напряжения, силы тока, сопротивления, угла поворота и т.п. в форме непрерывного (аналогового) сигнала. Подводимые к исполнительным механизмам управляющие воздействия должны вырабатываться в форме напряжений, т.е. также в аналоговой или дискретной форме.

Так как процессор компьютера оперирует с цифровыми величинами, то поступающие на его вход величины должны предварительно быть преобразованы в цифровую форму, а вырабатываемые управляющим вычислительным комплексом величины управляющих воздействий – из цифровой формы в аналоговую или дискретную, т.е. в соответствующие напряжения. Некоторые входные параметры (например, выдаваемые конечными выключателями, фотореле и др.) и некоторые выходные управляющие сигналы (например, включение двигателей) имеют релейный характер.

Неотъемлемой частью автоматизированной системы управления технологическими процессами являются устройства связи с объектом (УСО), назначение которых заключается в сопряжении датчиковой аппаратуры и исполнительных механизмов контролируемого объекта и/или технологического процесса с вычислительными средствами системы. *УСО представляет собой комплекс в виде специализированных функциональных блоков, осуществляющий необходимый информационный обмен между технологическим объектом и управляющей информационной системой.*

На **рис. 1.6** изображена общая структурная схема системы связи компьютера с объектом управления.

Как правило, на УСО возлагаются следующие функции.

- Нормализация аналогового сигнала – приведение границ шкалы первичного непрерывного сигнала к одному из стандартных диапазонов входного сигнала аналого-цифрового преобразователя измерительного канала. Наиболее распространены следующие диапазоны: 0–20 мА, 4–20 мА, от 0 до 5 В; от –5 до 5В; от 0 до 10 В, а также сигналы датчиков с естественными выходными сигналами (термопары, термометры сопротивления, тензометрические датчики и т.п.).
- Предварительная низкочастотная фильтрация аналогового сигнала – ограничение частот первичного сигнала с целью снижения влияния на результаты измеренных помех различного происхождения. На промышленных объектах наиболее распространены помехи с частотой сети переменного тока, а также хаотические импульсные помехи, вызванные влиянием на технические средства измерительного канала переходных процессов и наводок при коммутации исполнительных механизмов повышенной мощности.

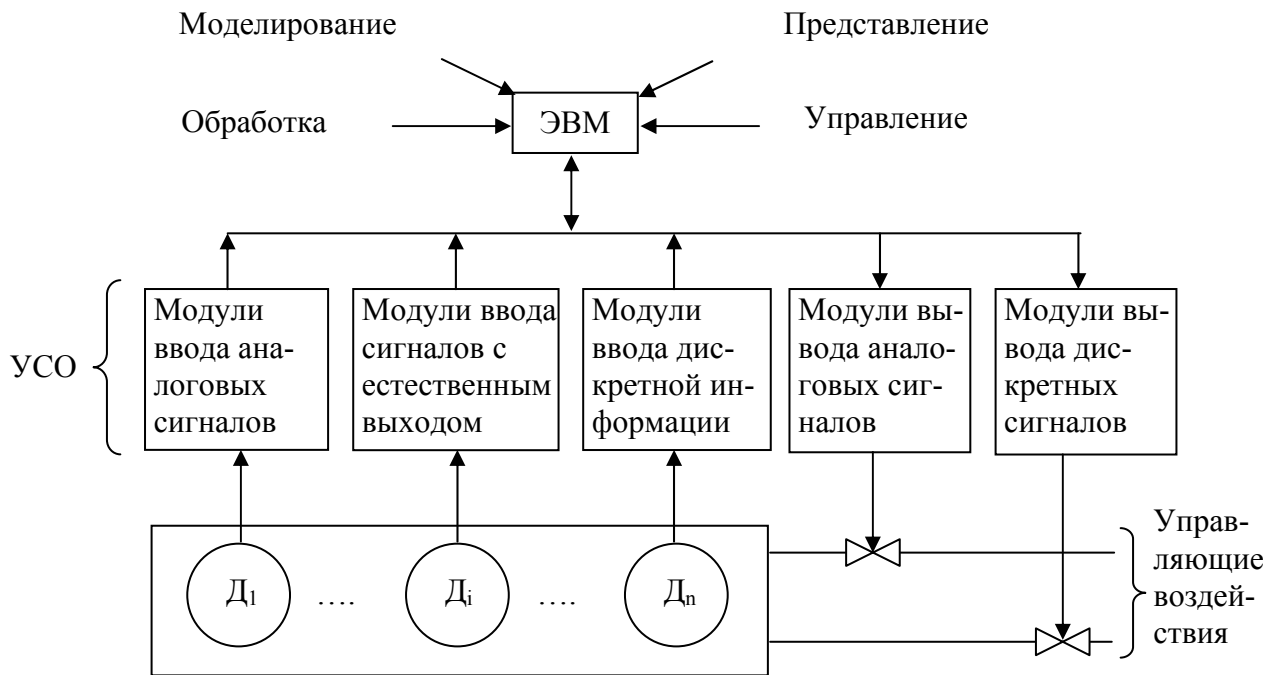


Рис. 1.6. Структурная схема связи компьютера с объектом управления

- Обеспечение гальванической развязки между источником аналогового или дискретного сигнала и измерительными каналами системы. В равной степени это относится к изоляции между каналами дискретного вывода системы и управляемым силовым оборудованием. Помимо собственно защиты выходных и входных цепей, гальваническая изоляция позволяет снизить влияние на систему помех по цепям заземления за счет полного разделения вычислительной системы и контролируемого оборудования.

Помимо перечисленных функций, ряд устройств связи с объектом может выполнять более сложные задачи за счет наличия в их составе подсистемы аналого-цифрового преобразования и дискретного ввода-вывода, микропроцессора и средств организации одного из интерфейсов последовательной передачи данных.

Распределенность физических параметров и широкий диапазон их видов и размерностей на металлургических объектах требуют многовариантного подхода к организации измерительных схем. Так, в одном случае превращение многоканальной информации в одноканальную, для ее обработки, осуществляется с помощью многоканального аналогового коммутатора, подключаемого на вход одного аналого-цифрового преобразователя, а в другом случае на каждый канал включается индивидуальный аналого-цифровой преобразователь. Преобразование многоканальной информации в одноканальную осуществляется с помощью цифрового мультиплексора. На этот мультиплексор данные поступают либо от аналого-цифрового преобразователя, либо от микропроцессорного измерительного блока. Ядром же многопроцессорной измерительной системы является *аналого-цифровой преобразователь, входящий в состав любой современной измерительной процедуры.*

Следует отметить, что к 80-м годам технология приборной связи 0,4...20МА Государственной системы приборов (ГСП) стала стандартом де факто. В результате производители контрольно-измерительной аппаратуры получили стандарт, на основе которого их продукты можно было интегрировать в информационные системы без особого труда.

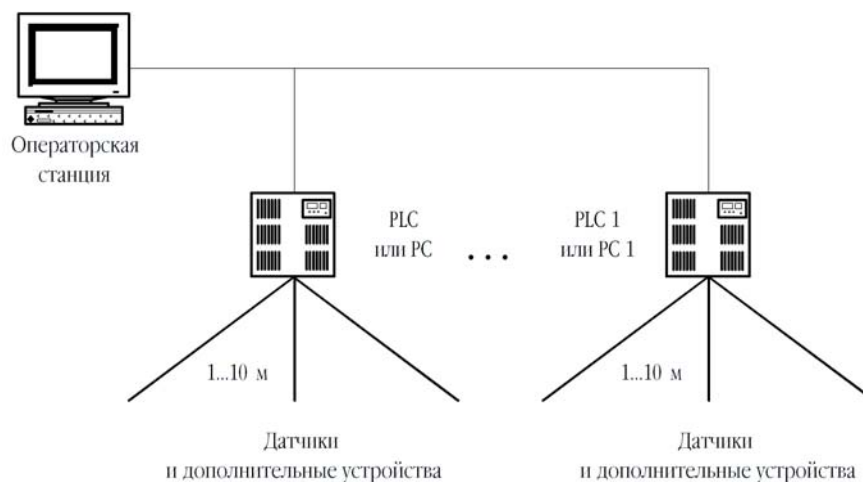
С развитием цифровой технологии ситуация в этой области стала изменяться. Пользователи постепенно переходят *от централизованных систем управления к распределенным системам управления*. В результате централизованные архитектуры начинают сдавать свои позиции на рынке, в то время как открытые распределенные системы начинают его завоевывать. Проходят те времена, когда нормой жизни считался огромный шкаф, напичканный автоматикой, с выходящими из него толстыми пучками кабелей, ведущими к датчикам и исполнительным механизмам. Сегодня в большинстве случаев становится более экономически целесообразной установка в цехе или на участке нескольких локальных контроллеров или УСО, объединенных в единую сеть, чем прокладка разветвленных кабельных систем. Системы с централизованным управлением обычно требуют, чтобы каждый датчик или группа датчиков подключались к центральному компьютеру отдельным и довольно дорогим высококачественным кабелем. В противоположность этому в распределенной системе рядом с каждой группой датчиков располагается один интеллектуальный узел, преобразующий сигналы датчиков в цифровую последовательность и передающий их в этом виде в систему управления. При этом прокладка кабелей и развертывание систем обходится значительно дешевле. Благодаря таким преимуществам, как экономичность решений, надежность и безопасность, в настоящее время осуществляется «цифровая революция», т.е. переход от аналоговой технологии к цифровой.

Таким образом, современная идеология построения сложных автоматизированных систем управления технологическими процессами уверенно развивается в направлении применения распределенных принципов построения систем (**рис. 1.7**) в противоположность централизованным (**рис. 1.8**). Объясняется это различными причинами, наиболее важными из которых являются:

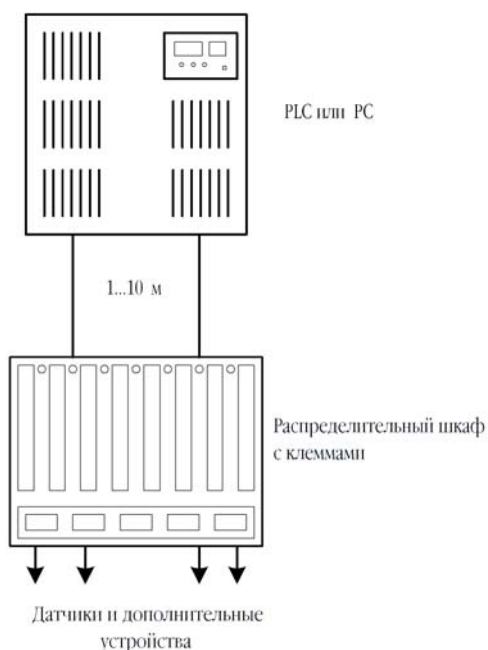
- значительное сокращение общих затрат на кабельную сеть, включающих как стоимость самих подключаемых кабелей, так и стоимость монтажных работ;
- стремительное удешевление вычислительной техники, позволившее применять автономные вычислители в каждом из узлов АСУ ТП в непосредственной близости от исполнительных устройств и датчиков.

Появление распределенных АСУ ТП привело к необходимости разработки специальных сетевых решений, ориентированных на эксплуатацию в промышленных условиях (Fieldbus – промышленная сеть). Основными требованиями к ним являются: высокая помехозащищенность, достаточная скорость передачи и низкая степень стоимости соединительного кабеля. При построении промышленных сетей, объединяющих PLC и компьютеры PC автоматизированных рабочих мест оператора, предложено множество различных решений. Однако в настоящее время отсутствует единый международный стандарт. Основными же «игроками» в области промышленных сетей сегодня являются: Profibus, CAN,

Interbus, Industrial Ethernet и др. В дальнейшем, при описании информационных систем в металлургии, мы остановимся на особенностях некоторых из этих решений более подробно.



*Рис. 1.7. Структура распределенной информационной системы*



*Рис. 1.8. Структура централизованной информационной системы*

Сегодня никакой производитель не может поставлять всю номенклатуру изделий, требующихся в современных информационных системах. От специалистов требуется умение применять для построения информационных систем высокотехнологичные изделия различных компаний, и, естественно, эти изделия должны быть совместимыми. Для гарантии совместимости различных производителей необходимы открытые стандарты аппаратных и программных средств.

Системы, являющиеся уникальными (их делает и поддерживает только один производитель), работающие по уникальным протоколам связи, получили название «Закрытые системы» (closed systems). Большинство таких систем за-

родилось во времена, когда проблема интеграции изделий других производителей не была актуальной.

«Открытые системы» (open systems) приводят специфические требования в соответствие интересам всех. Только при использовании принципов открытых систем интеграция изделий разных производителей в одну сеть может быть решена без особых проблем.

Открытость означает:

- отсутствие патентов или авторских прав на спецификацию стандарта и его расширений;
- отсутствие лицензионной платы за использование стандарта;
- широкий доступ к спецификациям стандарта и его расширениям;
- получение спецификаций в результате открытого обсуждения и конкурса между экспертами крупнейших промышленных ведущих мировых фирм-производителей и пользователей;
- принадлежность прав собственности некоммерческим профессиональным национальным и международным организациям.

Таким образом, архитектура современных информационных систем технологических процессов должна удовлетворять требованиям:

- открытости;
- стандартности;
- модульности;
- распределенности.

### **1.3.2. Информационные технологии хранения данных. Системы управления базами данных**

Известно, что компьютеры изначально были созданы для удовлетворения нужд крупных учреждений при реализации *большого объема вычислений*, для которых существенными являлись *точность и время вычислений*.

Как правило, вычисления представляют собой длинные цепочки итераций и требуют сохранения высокой точности (примерами таких вычислений могут служить решение дифференциальных уравнений, операции с матрицами и векторами, решение задач математического программирования и т.п.). Такие вычисления характерны для *числовой обработки*. Становление этого направления способствовало развитию математического моделирования технологических процессов в металлургии, идентификации методов численного решения сложных математических задач, развитию языков программирования и т.п. Эти возможности компьютеров и сегодня широко используются, в частности, при математическом моделировании технологических процессов.

Однако по мере распространения компьютеров и их совершенствования (главным образом, запоминающих устройств, сохраняющих информацию после выключения электрического питания) возникли другие области их применения. Так возникло другое направление использования средств вычислительной техники, которое предполагает отсутствие большого объема вычислений и их высокой точности. Обычно достаточно одно сложение или умножение. Однако объем

хранимых данных велик. Кроме того, необходимо предварительно найти требуемую запись, обработать ее и определить форму вывода обработанных данных. Для этого требуются такие операции, как поиск и сортировка. Весь этот процесс *характеризует нечисловую обработку данных*.

В этом случае информационная система – это программно-аппаратный комплекс, функциями которого являются:

- надежное хранение информации в электронном виде;
- предоставление доступа к информации пользователям системы;
- выполнение функций по преобразованию информации, специфичных для данного приложения;
- предоставление удобного интерфейса для конечных пользователей.

Напомним, что *оперативная память* компьютера хранит данные только в пределах одного сеанса работы пользователя и при выключении стирается. Этот вид памяти отличает высокое быстродействие, высокая стоимость и, как следствие, – ограниченный объем.

*Внешняя память* – более медленное, но более дешевое устройство хранения информации. Как правило, по объему оно во много раз превосходит оперативную память.

В сегодняшних условиях оба вида памяти стремительно дешевеют (в долларах), причем темпы удешевления оперативной памяти значительно выше, что оказывает существенное влияние на развитие информационных систем в частности.

Характеристики информационных систем:

- имеют дело с большими (огромными) объемами информации, которые во много раз превышают объем оперативной памяти, вся информация расположена на устройствах внешней памяти;
- работают, как правило, в многопользовательском режиме;
- работают, как правило, в реальном времени. Часто относятся к классу *mission-critical applications*, то есть приложений, нестабильность работы которых ведет к серьезным убыткам;
- развиваются, как правило, постепенно, а не сразу целиком, что повышает ответственность проектировщика и разработчика;
- обеспечивают «среднее» время ответа для всех пользователей, причем время ответа существенно не увеличивается при росте числа одновременно работающих пользователей.

Развитие второй области стало возможно после появления недорогих устройств внешней памяти, основанных на магнитных дисках. С другой стороны, именно это направление информационных систем способствовало развитию накопителей на магнитных дисках (НМД).

Типичными примерами больших информационных систем являются:

- системы бронирования и продажи авиа- и железнодорожных билетов;
- банковские системы, в том числе системы обслуживания кредитных карт;
- системы автоматизации торговли.

Сложность таких систем – не в отдельных частях (алгоритмах обработки данных), а в том, что они велики сами по себе, то есть широки по номенклатуре обрабатываемой информации, сложны по структуре аппаратной платформы и программного обеспечения, а также часто территориально распределены. Все это требует особенной тщательности при планировании, проектировании и реализации. Ошибки на начальных стадиях создания информационной системы особенно дороги.

Действительно, современные информационные системы на крупнейших металлургических предприятиях характеризуются огромными объемами хранимых данных (включающих сотни тысяч, миллионы и даже миллиарды отдельных записей), сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей. Для решения этих задач в современных информационных системах используются *базы и банки данных*. В дальнейшем дадим этому понятию более строгое определение, а пока лишь отметим, что **база данных** – это совокупность сведений об объектах в какой-либо предметной области. Создавая базу данных, пользователь стремится упорядочить информацию по различным критериям, хранить ее и быстро извлекать по определенным признакам. Сделать это возможно, только если данные *структурированы, т.е. введено соглашение о способах представления данных*.

Как мы увидим в дальнейшем, современный подход к организации данных предполагает использование **концепции централизованного управления данными**. Создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляется с помощью специального программного инструментария – системы управления базами данных (СУБД).

Таким образом, **система управления базами данных (СУБД)** – это комплекс программных средств, необходимых для создания баз данных, поддержки их в требуемом состоянии и организации поиска в них необходимой информации.

В дальнейшем мы рассмотрим подробнее методы моделирования данных, основы организации, проектирования и использования баз данных информационных систем в металлургии. А пока лишь отметим, что к основным задачам, которые решаются СУБД, относятся:

- хранение информации;
- контроль и защита информации;
- просмотр и поиск нужной информации;
- выборка необходимых данных;
- ввод и редактирование информации;
- формирование отчетов.

### 1.3.3. Модельные системы поддержки принятия решений

Модельные системы поддержки принятия решений и соответствующие им информационные технологии появились в основном в 70-80-е годы, чему способствовали развитие теории моделирования, математики, в особенности



численных методов решения, широкое распространение персональных компьютеров, стандартных пакетов программ.

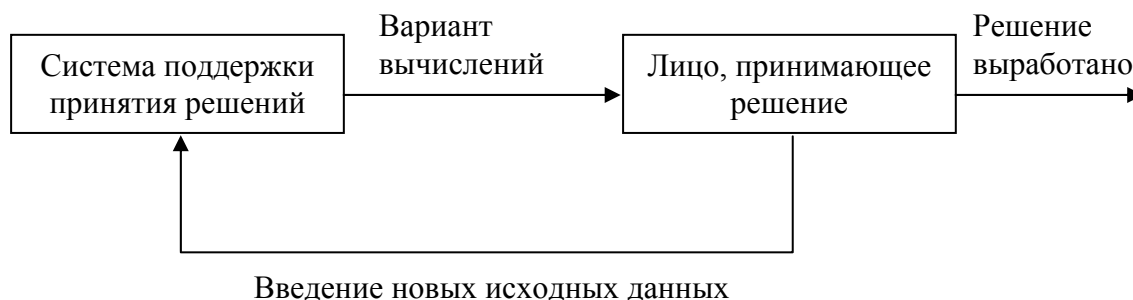
*Модельные системы поддержки принятия решений представляют собой вид компьютерных информационных систем, помогающих лицу, принимающему решение (ЛПР), в принятии решений при наличии плохо структурированных задач посредством прямого диалога с компьютером с использованием данных и математических моделей.*

Задачи считаются хорошо *структурированными*, если лицу, принимающему решение, известны все их элементы и взаимосвязи между ними. Обычно при таком высоком уровне понимания задачи удается выразить ее содержание в форме математических моделей, имеющих точный алгоритм решения. Эти задачи носят повторяющийся, рутинный характер. Целью использования информационных систем для решения структурированных задач является практически полная автоматизация их решения. Решение хорошо структурированных задач по жестким, раз и навсегда созданным алгоритмам, производится компьютером и, в принципе, не требует участия технолога (ЛПР).

Задачи, по которым ЛПР не удается выделить отдельные элементы и установить связи между ними, называются *неструктурированными*. Для решения неструктурированных проблем компьютер оказывается бесполезным, здесь основная работа остается за технологом (человеком). В практике управления технологическими процессами имеется сравнительно немного полностью структурированных или совершенно неструктурированных задач. О большинстве же задач можно сказать, что ЛПР имеет о них неполное представление, зная лишь часть их элементов и связей между ними. Такие задачи называются *плохо структурированными*. Именно такие задачи чаще всего и встречаются в металлургии при управлении технологическими процессами, качеством продукции и т.п. Решение плохо структурированных задач требует *использования компьютера совместно с усилиями человека* (производственного персонала, лица, принимающего решение). При этом информационные системы могут оказывать лицу, принимающему решение, три вида поддержки: *информационную, модельную и экспертную*, это и отражено на **рис. 1.1**.

Главной особенностью модельных систем поддержки принятия решений является качественно новый метод организации взаимодействия человека и компьютера. Выработка решения, что является основной целью этой технологии, происходит в результате итерационного процесса (**рис. 1.9**), в котором участвуют:

- система поддержки принятия решений в роли вычислительного звена и объекта;
- человек как управляющее звено, задающее входные данные и оценивающее полученный результат вычислений на компьютере.



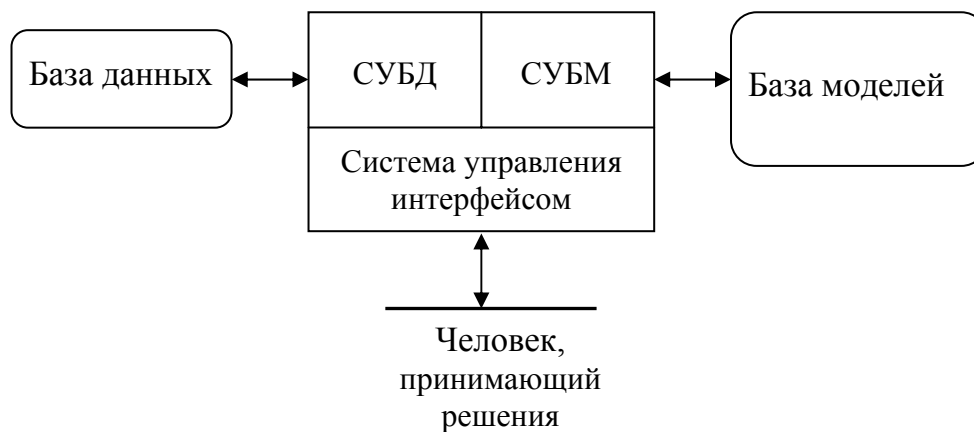
*Рис. 1.9. Технология модельной системы поддержки принятия решений как итерационный процесс*

Окончание итерационного процесса происходит по воле человека (оператора, технолога, лица, принимающего решение).

Рассмотрим в самом общем виде структуру модельной системы поддержки принятия решений.

В состав модельной системы поддержки принятия решений входят три главных компонента (**рис. 1.10**):

- 1) база данных;
- 2) база моделей;
- 3) система управления интерфейсом между пользователем и компьютером.



*Рис. 1.10. Основные компоненты модельной системы поддержки принятия решений*

*База данных* в модельной системе поддержки принятия решений играет важную роль, поскольку данные могут использоваться непосредственно пользователем для расчетов при помощи математических моделей.

*База моделей* (комплекс различных моделей) используется для описания и оптимизации процесса. Без моделей осуществлять процессы в информационных системах можно только методом проб и ошибок (дерганья металлургических систем), что, конечно же, неприемлемо в современной металлургии. Заметим, что комплекс математических моделей является основой модельной системы поддержки принятия решения. Пользователь имеет возможность получить

недостающую ему информацию для принятия решения путем установления диалога с моделью, что облегчает выработку и оценку альтернатив решения.

*Система управления базой моделей (СУБМ)* должна обладать следующими возможностями: создавать новые модели или изменять существующие, поддерживать и обновлять параметры моделей (осуществлять идентификацию параметров), манипулировать моделями.

*Система управления интерфейсом* определяет язык пользователя, язык сообщений компьютера, организующий диалог на экране дисплея. Язык пользователя – это те действия, которые пользователь производит в отношении системы путем использования возможностей клавиатуры, мыши и т.п. Язык сообщений – это то, что пользователь видит на экране дисплея, данные, полученные на принтере, звук и т.п. В процессе диалога пользователь должен реализовать свои знания. Сюда относится не только план действий, находящийся в голове у пользователя, но и инструкции, справочные данные, выдаваемые компьютером по команде о помощи. Инструкции и справочные данные, выдаваемые системой по просьбе пользователя, обычно не стандартны, а специализированы с точки зрения сложившейся ситуации.

В дальнейшем мы рассмотрим методологию математического моделирования основных процессов в металлургии на примере доменного производства и дадим краткую характеристику программным средствам и решаемым задачам в рамках модельной системы поддержки принятия решений.

#### **1.3.4. Информационная технология экспертных систем**

Наибольший прогресс среди компьютерных информационных систем отмечен в области разработки *экспертных систем*, основанных на использовании *искусственного интеллекта*.

Термин “искусственный интеллект” впервые ввел Джон Маккарти, профессор Стэнфордского университета, автор многих ярких работ по программированию, он же провел и первую конференцию по искусственному интеллекту.

Интерес к теории искусственного интеллекта и его созданию во всем мире существенно возрос после того, как Япония объявила в 1979 г. о проекте создания ЭВМ пятого поколения, обладающих возможностью интеллектуального диалога с непрограммирующими пользователями.

Под *искусственным интеллектом* обычно понимают способности компьютерных систем к таким действиям, которые назывались бы интеллектуальными, если бы исходили от человека. Чаще всего здесь имеются в виду способности, связанные с человеческим мышлением. Работы в области искусственного интеллекта не ограничиваются экспертными системами. Они включают в себя создание роботов, систем, моделирующих нервную систему человека, способность к обучению, распознавание образов и т.д. В металлургии получило наибольшее распространение важнейшее направление работ в области искусственного интеллекта – экспертные системы.

Главная идея использования технологии экспертных систем заключается в том, чтобы получить от эксперта его знания и, загрузив их в память компьютера, использовать всякий раз, когда в этом возникнет необходимость.

Традиционные знания существуют в двух видах – коллективный опыт и личный опыт.

Если большая часть знаний в предметной области представлена в виде коллективного опыта (например, теория тепломассопереноса), то эта предметная область не нуждается в экспертных системах (**рис. 1.11**).

Если в предметной области большая часть знаний является личным опытом специалистов высокого уровня – экспертов, если эти знания по каким-либо причинам слабо структурированы (например, технология задувки доменной печи), то такая предметная область, скорее всего, нуждается в экспертной системе (**рис. 1.12**).



**Рис. 1.11.** Предметная область, не пригодная для создания экспертных систем



**Рис. 1.12.** Предметная область, пригодная для создания экспертных систем

Являясь одним из основных приложений искусственного интеллекта, экспертные системы предназначены для моделирования или имитации поведения опытных специалистов-экспертов при решении задач по какому-либо узкому вопросу.

Таким образом, экспертные системы представляют собой компьютерные программы, трансформирующие опыт экспертов в какой-либо области знаний в форму эвристических правил. Заметим, что в экспертных системах мы имеем дело с использованием принципиально нового компонента информационных

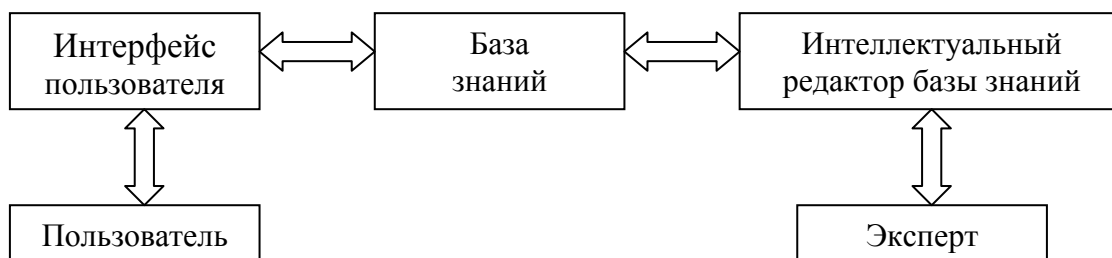
технологий – *базы знаний*, которая содержит факты, описывающие проблемную область, а также логическую взаимосвязь этих фактов. Напомним, что *знания* – выявленные закономерности предметной области (принципы, законы, связи), позволяющие решать задачи в этой области.

Заметим, что наряду с числовой обработкой данных (используемых в основном при математическом моделировании процессов), нечисловой обработкой данных (в системах управления базами данных) компьютеры применяются в новой области – для моделирования и хранения знаний. В последнем случае программно-технические средства используются для решения неформализованных задач, в которых формулировка не может быть формально определена в числовом виде, а определяется лишь в смысловом представлении на ограниченном естественном языке.

В самом общем виде обобщенная структура экспертной системы представлена на **рис. 1.13**. На этом рисунке:

- *интерфейс пользователя* – это комплекс программ, реализующий диалог пользователя с экспертной системой, как на стадии ввода, так и получения результатов;
- *интеллектуальный редактор базы знаний* – программа, представляющая возможность эксперту создавать базу знаний в диалогом режиме.

Чем экспертные системы отличаются от модельных систем поддержки принятия решения? Экспертные системы и модельные системы поддержки принятия решения сходны в том смысле, что обе они обеспечивают высокий уровень поддержки принятия решений. Однако они имеют два существенных различия.



**Рис. 1.13.** К понятию экспертной системы

*Первое отличие* связано с тем, что решение проблемы в рамках модельных систем поддержки принятия решений отражает уровень ее понимания пользователем и его возможности получить и осмыслить решение. Экспертная система, наоборот, предлагает пользователю принять решение, превосходящее его возможности.

*Второе отличие* выражается в способности экспертной системы пояснять свои рассуждения в процессе получения решения. Очень часто эти пояснения оказываются более важными для пользователя, чем само решение.

В дальнейшем мы уточним структуру экспертных систем, рассмотрим некоторые модели знаний, дадим характеристику инструментальных средств по-

строения экспертных систем, а также отразим первый опыт внедрения в металлургии экспертных систем в отечественных и зарубежных фирмах.

### **1.3.5. Информационные системы передачи данных (сетевые технологии)**

Современное металлургическое производство требует высоких скоростей обработки информации, удобных форм ее хранения и передачи. Управление крупными предприятиями, к которым относятся в первую очередь металлургические заводы, невозможно без создания *единого информационного пространства внутри предприятия* с целью обмена потоками данных, т.е. функционирования локальных и региональных компьютерных сетей. Так, ОАО ММК является одним из самых больших металлургических предприятий, имеющим сотни цехов, занимающим большую территорию, имеющим много удаленных объектов, объединенных, однако, единой технологической цепочкой. Удаленность некоторых объектов от центрального комплекса комбината достигает 20 км, а расстояние между отдельными цехами и производствами – 50 км. Необходимость поставок сырья, топлива, интеграции в мировое разделение труда сегодня возможно только при наличии компьютерных сетей не только внутри предприятия, но и *выхода их в региональные и глобальные компьютерные сети*.

Таким образом, единое информационное пространство металлургического предприятия возможно только при наличии ряда независимых друг от друга локальных сетей (как правило, в пределах цеха), а также при обеспечении их интеграционного взаимодействия друг с другом, т.е. функционирования региональной корпоративной сети (в пределах металлургического завода) и выхода последней в глобальную сеть.

Сегодняшний уровень развития науки и техники позволяет при помощи компьютерных сетей из любой точки мира за несколько секунд связаться с любым объектом металлургического завода и зафиксировать его технико-экономические показатели. Для этого не существует принципиальных технических проблем. Сегодня на планете уже имеются сотни миллионов пользователей Internet. Доставка информационных услуг посредством телекоммуникаций занимает сегодня значительную часть мирового рынка. Поэтому одной из разновидностей постиндустриального общества, по мнению многих аналитиков, становится так называемое «информационное общество» (Information Society, IS). Иногда к этому названию добавляют еще слово «глобальное» (Global). Ясные и логические цели глобальной информатизации, будучи достигнутыми, несомненно обещают основательно изменить повседневную жизнь каждого человека, да и общества в целом.

Очевидно, что сегодня информационные сетевые технологии в металлургии – одно из самых важнейших, бурно развивающихся направлений информационных технологий, поэтому они также будут предметом нашего дальнейшего рассмотрения.

### **1.3.6. Понятие традиционной и новой информационных технологий**

*Традиционная информационная технология*, существовавшая до развития теории искусственного интеллекта и создания мощных быстродействующих средств вычислительной техники, объективно предусматривала при использовании ЭВМ непрограммирующим специалистом данной предметной области либо непосредственное участие математиков, алгоритмистов и программистов, либо требовала от специалиста в данной предметной области обучения принципиально новым, не свойственным его специальности знаниям в области математики, программирования ЭВМ. Традиционная информационная технология, которая обеспечивает только переработку данных, позволяет успешно решать с использованием различных проблемно-ориентированных, или алгоритмических, языков программирования сложные вычислительные, или формализованные, задачи.

*Новая информационная технология*, научной основой которой является теория искусственного интеллекта, позволяет специалисту в данной конкретной области, например металлургу, – непрограммирующему конечному пользователю ЭВМ – непосредственно, без промежуточных помощников и без специальных знаний в области математики, программирования и вычислительной техники, обращаться с ЭВМ на ограниченном естественном языке, как при формулировке, так и при поиске решений неформализованных задач. Новая информационная технология, обеспечивающая переработку как разнообразных знаний предметной области, выраженных на ограниченном естественном языке, так и данных, превращает ЭВМ в удобного партнера для непрограммирующего конечного пользователя при поиске решений неформализованных задач, которые возникают в его профессиональной и творческой деятельности.

Таким образом, новую информационную технологию можно охарактеризовать как информационную технологию с “дружественным” интерфейсом работы пользователя, позволяющую получить новую информацию об объекте (технологическом процессе).

В дальнейшем мы рассмотрим примеры реализации некоторых новых информационных технологий в металлургии на примере, пожалуй, самого сложного процесса в черной металлургии – доменного.

### **1.4. Контрольные вопросы**

1. Дайте определение терминам «информация» и «информатика». Чем различаются эти понятия?
2. Что составляет предмет и каковы задачи изучения информатики?
3. Какие функции и операции выполняет автоматизированная информационная система?
4. Перечислите основные уровни современной автоматизированной информационной системы промышленного предприятия, дайте им краткую характеристику. Какие аппаратно-программные средства используются на каждом уровне автоматизации?

5. Что называется информационной технологией, какова цель ее использования? На какие виды подразделяют информационные технологии в зависимости от типа обрабатываемой информации?
6. Какие функции выполняют устройства связи компьютера с объектом управления (УСО)?
7. В чем заключаются отличия открытых от закрытых информационных систем? Какие преимущества дает использование открытых систем?
8. На каких принципах основана технология хранения данных в современных информационных системах?
9. Что называется системой поддержки принятия решений? Какие компоненты она в себя включает?
10. В каких областях используют экспертные системы? Чем экспертные системы отличаются от модельных систем поддержки принятия решения?
11. Какие признаки отличают новые информационные технологии от традиционных?



---

ГЛАВА 2. АРХИТЕКТУРА СОВРЕМЕННЫХ ПРОГРАММНЫХ СРЕДСТВ .....	2-2
2.1. ОСНОВНЫЕ ЭЛЕМЕНТЫ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА .....	2-2
2.2. СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕМ .....	2-5
2.2.1. Принципы организации операционной системы .....	2-6
2.2.2. Общая характеристика операционных систем общего назначения .....	2-18
2.2.3. Операционные оболочки .....	2-23
2.2.4. Сервисное программное обеспечение (утилиты) .....	2-23
2.3. ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ .....	2-24
2.3.1. Принципы организации пакетов прикладных программ .....	2-25
2.3.2. Основные группы пакетов прикладных программ .....	2-27
2.4. ОСОБЕННОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ .....	2-35
2.5. ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ ОПЕРАТОРСКИХ СТАНЦИЙ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ (SCADA) .....	2-41
2.5.1. Общая характеристика пакетов .....	2-41
2.5.2. GENESIS (США) .....	2-45
2.5.3. SIMATIC WinCC – модульная SCADA-система для мониторинга технологических процессов .....	2-50
2.5.4. BASEstar – комплект программного обеспечения для АСУ ТП в реальном времени фирмы DIGITAL (США) .....	2-51
2.6. ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ УПРАВЛЕНИЯ ПРОИЗВОДСТВОМ ....	2-54
2.6.1. Общая характеристика пакетов .....	2-54
2.6.2. MANUFACTURING TOTAL MANAGEMENT ASYSTEMS (MTMS) .....	2-57
2.6.3. Программный комплекс Oracle Applications .....	2-57
2.6.4. Программный комплекс управления предприятием IFS Applications .....	2-58
2.6.5. Интегрированное семейство прикладных программ TRITON (ТРАЙТОН) .....	2-59
2.6.6. Интегрированный программный пакет Factory Suite 2000 .....	2-60
2.7. КОНТРОЛЬНЫЕ ВОПРОСЫ .....	2-61

## Глава 2. АРХИТЕКТУРА СОВРЕМЕННЫХ ПРОГРАММНЫХ СРЕДСТВ

Прежде чем остановиться на архитектуре современных программных средств, в самых общих чертах напомним принцип работы и структуру современного компьютера [16].

### 2.1. Основные элементы персонального компьютера

Напомним, что *компьютер* – это электронный прибор, предназначенный для автоматизации создания, хранения, обработки и транспортировки данных. Компьютер представляет собой устройство, способное исполнять четко определенную последовательность операций, предписанную программой. Заметим, что понятие «компьютер» является более широким, чем понятие «ЭВМ», поскольку в последнем явный акцент делается на вычисления.

Особенно бурное развитие и применение за последние двадцать лет в промышленности вообще и в металлургии в частности получили персональные компьютеры. Многие современные персональные модели превосходят по своим возможностям большие ЭВМ 70-х годов, миниЭВМ 80-х годов и микроЭВМ первой половины 90-х годов. Персональный компьютер характеризуется тем, что им может пользоваться один человек, не прибегая в помощи бригады обслуживающего персонала.

Любой компьютер имеет три основные составные части:

- 1) процессор;
- 2) память;
- 3) периферийные устройства.

Они взаимодействуют между собой с помощью *шин*, стандартизация которых делает архитектуру компьютера открытой.

*Процессор* является основным мозговым узлом, в задачу которого входит исполнение программного кода, находящегося в памяти.

В настоящее время для персональных ЭВМ под словом «процессор» подразумевается микропроцессор – микросхема, которая, кроме собственного процессора, может содержать другие узлы, например КЭШ (быстрая) память.

Процессор в определенной последовательности выбирает из памяти инструкции и исполняет их. Инструкции процессора предназначены для пересылки, обработки и анализа данных, расположенных в пространствах памяти и портов ввода-вывода, а также организации ветвлений и переходов в вычислительном процессе.

В компьютере обязательно должен быть *центральный процессор*, который исполняет основную программу. В многопроцессорных системах функции центрального процессора распределяются между несколькими обычно идентичными процессами для повышения общей производительности, а один из них называют главным. В помощь центральному процессору в компьютер часто вводят *сопроцессоры*, ориентированные на эффективное использование каких-либо специфических функций. Широко распространены математические сопроцессоры, эффективно обрабатывающие числовые данные с плавающей точкой; графиче-

ческие сопроцессоры, выполняющие графические построения и обработку графических изображений и т.п. Возможны и другие сопроцессоры, но все они не самостоятельны, так как исполнение основного вычислительного процесса осуществляется централизованным процессором.

Память компьютера предназначена для кратковременного и долговременного хранения информации – кодов команд и данных. Информация в памяти хранится в двоичных кодах, каждый бит – элементарная ячейка памяти – может принимать значение «0» или «1». Каждая ячейка памяти имеет свой адрес, однозначно ее идентифицирующий в определенной системе координат.

Минимальной адресуемой единицей хранения информации в памяти обычно является байт, состоящий, как правило, из 8 бит.

**Внутренняя память** – электронная память, устанавливаемая на системной плате или на плате расширения.

**Внешняя память** – это память, реализованная в виде устройств с различными принципами хранения информации и обычно с подвижными носителями.

Для процессора непосредственно доступной является внутренняя память, доступ к которой осуществляется по адресу, заданному программой.

*Внутренняя память* подразделяется на **оперативную**, информация в которой может изменяться процессором в любой момент времени, и **постоянную**, информацию которой процессор может только считывать.

Оперативная память компьютера хранит данные только в пределах одного сеанса.

Внутренняя и внешняя память используются существенно разными способами.

Внутренняя (оперативная и постоянная) память является хранилищем постоянного кода, который непосредственно может быть исполнен процессором. В ней же хранятся данные, также непосредственно доступные процессору, а следовательно, и исполняемой программе.

Внешняя память обычно используется для хранения файлов, содержание которых может быть произвольным. Процессор (программа) имеет доступ к содержимому файлов только опосредованно, через отображение их (полное или частичное) в некоторой области оперативной памяти.

Однако реальная жизнь многовариантнее этой упрощенной схемы, и на практике дисковая и оперативная память переплетаются сложным образом.

*Главный недостаток дисковой памяти* – большое время доступа и низкая скорость обмена – устраняется при использовании *виртуального диска*, представляющего собой своеобразно используемую область оперативной памяти. В этой области хранятся файлы, и, с точки зрения операционной системы, она выглядит как обычный, но только очень быстрый диск. Конечно же, его объем ограничен, и этот объем вычитается из объема физически установленной памяти, доступной процессору в качестве обычной оперативной. Кроме того, виртуальный диск в отличие от реального не является энергонезависимым. Более того, информация в нем не переживет даже перегрузки операционной системы. Несмотря на эти ограничения, виртуальный диск во многих случаях может повысить эффективность работы компьютера при интенсивном дисковом обмене.

Другим способом решения проблемы быстродействия дисковой памяти за счет оперативной является кэширование дисков. Кэширование дисков – хранение образов последних из используемых блоков дисковой памяти в оперативной памяти в надежде на то, что к ним вскоре будет послан следующий запрос, который удастся удалить из памяти.

*Основной недостаток оперативной памяти заключается в том, что конструктивно достижимый ее объем во много раз меньше, чем у дисковой.* Решить проблему увеличения объема оперативной памяти за счет дискового позволяет *виртуальная память*, которую можно считать кэшированием оперативной памяти на диске.

Суть ее заключается в том, что программа предоставляет *виртуальное пространство* оперативной памяти, по размерам превышающее объем физически установленной оперативной памяти. Это виртуальное пространство разбито на страницы фиксированного размера, а в физической оперативной памяти в каждый момент времени присутствует только часть из них. Остальные страницы хранятся на диске, откуда операционная система может их «подкачать» в физическую на место предварительно выгруженных на диск страниц.

**Периферийные (внешние) устройства** являются важной составной частью любого вычислительного комплекса. В самом общем виде периферийные устройства предназначены для взаимодействия компьютера с внешним миром, и без них компьютер был бы вещью в себе. Список периферийных устройств весьма разнообразен и может быть классифицирован по ряду признаков. Например, по назначению можно выделить следующие виды периферийных устройств:

- устройства ввода информации;
- устройства вывода информации;
- внешние устройства хранения информации (внешняя память);
- коммуникационные устройства.

Программное обеспечение, используемое в компьютерах, условно можно разбить на три уровня. Уровни программного обеспечения представляют собой пирамидальную структуру: каждый следующий уровень опирается на программное обеспечение предшествующего уровня.

*Первый уровень – базовый уровень.* Этот уровень программного обеспечения хранится в постоянной (энергонезависимой) памяти. Она обеспечивает при включении тестирование и запуск компьютера, загрузку операционной системы, а также связь операционной системы с аппаратными средствами компьютера. Эта часть называется *базовой системой ввода-вывода BIOS (Basic Input Output System)*. Программы и данные этого уровня записываются («прошиваются») в микросхемы ПЗУ на этапе производства и не могут быть изменены в процессе эксплуатации. В тех случаях, когда изменение базовых программных средств во время эксплуатации является технически целесообразным, вместо микросхем ПЗУ применяют *перепрограммируемые постоянные запоминающие устройства (ППЗУ)*. В этом случае изменение содержания ПЗУ можно выполнить как непосредственно в составе вычислительной системы (такая технология называется флэш-технологией), так и вне ее, на специальных устройствах, называемых *программаторами*.

*Второй уровень – операционная система*, основным назначением которой является загрузка прикладных программ и представление им некоторого сервиса. Программы, работающие на этом уровне, обеспечивают взаимодействие прочих программ компьютерной системы с программами базового уровня и непосредственно с аппаратным обеспечением.

И, наконец, *третий – верхний уровень – прикладное программное обеспечение*, ради исполнения которого и гордится весь этот огород.

Именно возможность загрузки любой прикладной программы в совокупности с широким ассортиментом периферийных устройств и позволяет считать персональный компьютер универсальным инструментом с большими возможностями.

Для поддержки информационных технологий выделяются три класса программных продуктов (**рис. 2.1**).

1. Системное программное обеспечение.
2. Пакеты прикладных программ.
3. Инструментарий технологии программирования.



*Рис. 2.1. Классы программных продуктов*

## 2.2. Системное программное обеспечение

**Системное программное обеспечение** – совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ.

Структура системного программного обеспечения, представленная на **рис. 2.2**, состоит из базового и сервисного программного обеспечения.

**Базовое программное обеспечение** – минимальный набор программных средств, обеспечивающих работу компьютера.

Обычно в базовое программное обеспечение входят:

- 1) операционные системы;
- 2) операционные оболочки (текстовые и графические);
- 3) сетевые операционные системы.

**Сервисное программное обеспечение** – программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя.

В состав сервисного программного обеспечения обычно включают:

- 1) программы диагностики работы компьютера;
- 2) антивирусные программы;
- 3) программы обслуживания дисков;
- 4) программы архивирования данных;
- 5) программы обслуживания сети.

### 2.2.1. Принципы организации операционной системы

**Операционная система (ОС)** – основная программа, под управлением которой работает компьютер, т.е. это совокупность программных средств, обеспечивающих управление аппаратной частью компьютера и прикладными программами, а также взаимодействие между собой и пользователем. Она



**Рис. 2.2.** Классификация системного программного обеспечения компьютера

предназначена для управления ресурсами ЭВМ, т.е. выполнения пользовательских программ, планирования и управления вычислительными ресурсами ЭВМ.

Операционные системы занимают особое место среди всех программ и загружаются при включении компьютера.

ОС выполняет две по существу мало связанные функции:

- *обеспечение пользователю-программисту удобств посредством предоставления для него удобного интерфейса;*
- *повышение эффективности использования компьютера путем рационального управления его ресурсами.*

### **ОС как удобный интерфейс**

Использование большинства компьютеров на уровне машинного языка затруднительно, особенно это касается ввода-вывода. Первые машины выпускались вообще без операционных систем, то есть были «голыми». Они управлялись с помощью набора переключателей на пульте управления, и на такой машине, естественно, могли работать только инженеры высшей квалификации. Даже для выполнения такого несложного действия, как копирование файла с одной дискеты на другую, необходимо выполнить тысячи элементарных операций по запуску команд дисководов, проверке их выполнения, поиску и обработке информации и т.д. Стремлением уменьшить влияние специалистов и приблизить ЭВМ к простому пользователю и вызвано появление первых операционных систем. Программа, которая скрывает от программиста все реалии аппаратуры и предоставляет возможность простого, удобного просмотра указанных файлов, чтения или записи, т.е. удобный интерфейс, – это, конечно, операционная система.

### **ОС как система управления ресурсами**

Идея о том, что ОС, прежде всего система, обеспечивающая удобный интерфейс пользователям, соответствует рассмотрению сверху вниз. Другой взгляд, снизу вверх, дает представление об ОС как о некотором механизме, управляющем всеми частями сложной системы. Современные вычислительные системы состоят из процессоров, памяти, таймеров, дисков, накопителей на магнитных лентах, сетевой коммуникационной аппаратуры, принтеров и других устройств. В соответствии со вторым подходом функцией ОС является распределение процессоров, памяти, устройств и данных между процессами, конкурирующими за эти ресурсы. ОС должна управлять всеми ресурсами вычислительной машины таким образом, чтобы обеспечить максимальную эффективность ее функционирования.

Для решения этих общих задач управления ресурсами разные ОС используют различные алгоритмы, что, в конечном счете, и определяет их облик в целом, включая характеристики производительности, область применения и даже пользовательский интерфейс.

Обычно операционная система хранится на жестком диске, а при его отсутствии выделяется специальный диск, который называется системным диском.

**Системный диск** – это диск, где хранятся основные модули операционной системы и сервисные программы, расширяющие ее возможности.

При включении компьютера операционная система автоматически осуществляет загрузку в оперативную память всех ее программ, передает им управление в начале их работы, выполняет различные действия по запросу выполняемых программ и освобождает занимаемую программами оперативную память при их завершении.

Любая прикладная программа связана с операционной системой и может эксплуатироваться только на тех компьютерах, в которых имеется соответствующая системная среда. Прикладные программные средства, разработанные в среде одной операционной системы, не могут быть использованы для работы в среде другой операционной системы, если нет специального комплекса программ (конвертора), позволяющего это сделать. В таком случае говорят о программной несовместимости компьютеров.

Для работы с операционной системой необходимо овладеть языком этой среды, т.к. любая операционная система представляет собой набор программ, управляющих ресурсами ЭВМ. Работа этих программ строго координирована.

### *Требования, предъявляемые к современным ОС общего назначения*

Операционная система является сердцевиной сетевого программного обеспечения, она создает среду для выполнения приложений и во многом определяет, какими полезными для пользователя свойствами эти приложения будут обладать. В связи с этим рассмотрим требования, которым должна удовлетворять современная ОС.

Очевидно, что главным требованием, предъявляемым к операционной системе, является способность выполнения основных функций: эффективного управления ресурсами и обеспечения удобного интерфейса для пользователя и прикладных программ. Современная ОС, как правило, должна реализовывать мультипрограммную обработку, виртуальную память, поддерживать многооконный интерфейс, а также выполнять многие другие, совершенно необходимые функции. Кроме этих функциональных требований к операционным системам предъявляются не менее важные рыночные требования, к рассмотрению которых мы и переходим.

- *Расширяемость.* ОС должна быть написана таким образом, чтобы можно было легко внести дополнения и изменения, если это потребуется, и не нарушить целостность системы. Является ли операционная система маленькой, как MS DOS, или большой, как Unix, для нее неизбежно возникнет необходимость приобрести свойства, не заложенные в ее конструкцию. В то время как аппаратная часть компьютера устаревает за несколько лет, полезная жизнь операционных систем может измеряться десятилетиями. Примером может служить ОС Unix. Поэтому операционные системы всегда эволюционно изменяются со временем, и эти изменения более значимы, чем изменения аппаратных средств. Изменения ОС обычно представляют собой приобретение ею новых свойств. Например, поддержка новых устройств, таких как CD-ROM, возможность связи с сетями нового типа, поддержка новых многообещающих технологий, таких как графический интерфейс пользователя или объектно-ориентированное программное окружение, использование более чем одного процессора. Сохранение цело-



стности кода, какие бы изменения не вносились в операционную систему, является главной целью ее разработки.

Чтобы удовлетворять требованию расширяемости, ОС должна иметь достаточно гибкую архитектуру, которая позволила бы в дальнейшем легко включать в систему поддержку микропроцессорных технологий и периферийных устройств. Расширяемость достигается за счет *модульной структуры ОС*, при которой программы строятся из набора отдельных модулей, взаимодействующих только через функциональный интерфейс. Новые компоненты могут быть добавлены в операционную систему модульным путем, они выполняют свою работу, используя интерфейсы, поддерживаемые существующими компонентами.

- *Переносимость.* ОС должна легко переноситься с процессора одного типа на процессор другого типа и с аппаратной платформы (которая включает наряду с типом процессора и способ организации всей аппаратуры компьютера) одного типа на аппаратную платформу другого типа.

При этом следует учесть, в какое физическое окружение программа должна быть перенесена. Различная аппаратура требует различных решений при создании ОС. Например, ОС, построенная на 32-битовых адресах, не может быть перенесена на машину с 16-битовыми адресами (разве что с огромными трудностями).

При разработке ОС важно минимизировать или, если возможно, исключить те части кода, которые непосредственно взаимодействуют с аппаратными средствами. Зависимость от аппаратуры может иметь много форм. Некоторые очевидные формы зависимости включают прямое манипулирование регистрами и другими аппаратными средствами.

Если аппаратно зависимый код не может быть полностью исключен, то он должен быть изолирован в нескольких хорошо локализуемых модулях. Аппаратно-зависимый код не должен быть распределен по всей системе. Например, можно спрятать аппаратно-зависимую структуру в программно-задаваемые данные абстрактного типа. Другие модули системы будут работать с этими данными, а не с аппаратурой, используя набор некоторых функций. Когда ОС переносится, то изменяются только эти данные и функции, которые ими манипулируют.

Проиллюстрируем решение проблемы переносимости ОС Windows NT. Заметим, что отсутствие переносимости ОС OS/2, безусловно, было основной причиной, по которой Microsoft порвал с IBM и начал в свое время разработку ОС Windows NT. ОС OS/2 была написана на языке ассемблера и поэтому сильно привязана к конкретной аппаратной платформе, что делало перенос OS/2 на другую платформу практически невозможным. Основная часть кода ОС Windows NT написана на языке C, который разрабатывался как модульный и переносимый язык программирования. Ассемблер использовался только в тех кусках кода, где основное внимание уделялось скорости работы или непосредственному доступу к аппаратным устройствам.

- *Надежность и отказоустойчивость.* Система должна быть защищена как от внутренних, так и от внешних ошибок, сбоев и отказов. Ее действия должны быть всегда предсказуемыми, а приложения не должны быть в состоя-

нии наносить вред ОС. Неожиданный сбой в одной из программ никогда не должен приводить к нестабильности другого приложения и тем более операционной системы.

- *Совместимость.* ОС должна иметь средства для выполнения прикладных программ, написанных для старых версий этой ОС и других операционных систем. Кроме того, пользовательский интерфейс должен быть совместим с существующими системами и стандартами.

- *Безопасность.* ОС должна обладать средствами защиты ресурсов одних пользователей от других (рассмотрим это требование подробнее при описании сетевых ОС).

- *Производительность.* Система должна обладать настолько хорошим быстродействием и временем реакции, насколько это позволяет аппаратная платформа.

### *Особенности алгоритмов управления ресурсами*

В зависимости от особенностей использованного процессором алгоритма управления, операционные системы делят на многозадачные и однозадачные, многопользовательские и однопользовательские, на системы, поддерживающие многопотоковую обработку и не поддерживающие ее, на многопроцессорные и однопроцессорные системы.

**Поддержка многозадачности.** По числу одновременно выполняемых задач операционные системы могут быть разделены на два класса:

- *однозадачные* (например, MS DOS);
- *многозадачные* (например, OS/2, Unix, Windows 95/98/NT/2000).

Однозадачность означает, что ОС способна обрабатывать только одну задачу, а многозадачность – более одной задачи в одно и то же время. Однозадачные ОС в основном выполняют функцию предоставления пользователю виртуальной машины, делая более простым и удобным процесс взаимодействия пользователя с компьютером. Однозадачные ОС включают средства управления периферийными устройствами, средства управления файлами, средства общения с пользователем.

Многозадачные ОС, кроме вышперечисленных функций, управляют разделением совместно используемых ресурсов, таких как процессор, оперативная память, файлы и внешние устройства.

**Поддержка многопользовательского режима.** По числу одновременно работающих пользователей ОС делятся:

- *на однопользовательские* (например, MS DOS, Windows 3.x, ранние версии OS/2);
- *на многопользовательские* (например, Unix, Windows NT/2000).

Главным отличием многопользовательских систем от однопользовательских является наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей. Следует заметить, что не всякая многозадачная система является многопользовательской и не всякая однопользовательская ОС является однозадачной.

**Вытесняющая и невытесняющая многозадачность.** Важнейшим разделяемым ресурсом является процессорное время. Способ распределения процессорного времени между несколькими одновременно существующими в системе процессами (или нитями) во многом определяет специфику ОС. Среди множества существующих вариантов реализации многозадачности можно выделить две группы алгоритмов:

- *невытесняющая многозадачность* (Novell NetWare 3.x, Windows 3.x);
- *вытесняющая многозадачность* (Windows NT/2000, OS/2, Unix).

*Основным различием между вытесняющим и невытесняющим вариантами многозадачности является степень централизации механизма планирования процессов.* Вытесняющая многозадачность означает, что ОС контролирует все ресурсы (оперативную память, процессорное время и т.п.), определяет, какие службы их используют в данный момент и какое время они могут продолжать их использовать. При невытесняющей многозадачности процессы работают без непосредственного контроля со стороны ОС. Ответственность за корректное поведение системы в целом возлагается на все приложения и их разработчиков. При невытесняющей многозадачности активный процесс выполняется до тех пор, пока он сам, по собственной инициативе, не отдаст управление операционной системе для того, чтобы та выбрала из очереди другой готовый к выполнению процесс.

**Поддержка многонитевости.** Важным свойством операционных систем является возможность распараллеливания вычислений в рамках одной задачи. Многонитевая ОС разделяет процессорное время не между задачами, а между их отдельными ветвями (нитьями).

**Многопроцессорная обработка.** Другим важным свойством ОС является отсутствие или наличие в ней средств поддержки многопроцессорной обработки – *мультипроцессирование*. Мультипроцессирование приводит к усложнению всех алгоритмов управления ресурсами.

Многопроцессорные ОС могут классифицироваться по способу организации вычислительного процесса в системе с многопроцессорной архитектурой: асимметричные ОС и симметричные ОС.

При использовании *асимметричной* многопроцессорности только один или несколько процессоров выполняют задачу поддержки ОС. Все другие процессоры заняты поддержкой дополнительных функций, таких как пользовательские приложения системы, служба сети, т.е. распределяют прикладные задачи по остальным процессорам. Системы, использующие асимметричную многопроцессорность, обычно сильно привязаны к конкретной аппаратной платформе, что противоречит принципу переносимости. В то же время асимметричную модель проще реализовать.

В *симметричной* модели многопроцессорности ОС и службы приложений распределяются по всем свободным процессорам. Иными словами, симметричная ОС полностью децентрализована и использует весь пул процессоров, разделяя их между системными и прикладными задачами.

Сегодня становится общепринятым введение в ОС функций поддержки многопроцессорной обработки данных. Такие функции имеются, в частности, в операционной системе Unix.

Заметим, что многозадачность и многопроцессорность – разные понятия, хотя многозадачность поддерживает многопроцессорность. Многозадачность появляется благодаря совместным усилиям микропроцессора и операционной системы. Например, можно запустить MS DOS (однозадачная ОС) на четырехпроцессорной системе, но все равно будет выполняться только одна задача. В то же время, если запустить ОС Unix (многозадачная ОС) на процессоре, не поддерживающем многозадачности, то процессор не позволит реализовать возможность многозадачности этой ОС. Только многозадачная ОС, работающая на одном или нескольких процессорах, поддерживающих многозадачность, – единственный способ достигнуть истинной многозадачности.

Отметим, однако, что в случае использования однопроцессорной системы микропроцессор должен распределять свое время между различными задачами. Высокая скорость, с которой процессор переключается между задачами, создает у пользователя иллюзию, что одновременно выполняется более одной задачи, т.е. иллюзию многозадачности.

### *Особенности областей использования ОС*

Многозадачные ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности:

- системы пакетной обработки (например, ОС ЕС);
- системы разделения времени (например, Unix);
- системы реального времени (например, OS9, QNX).

**Системы пакетной обработки** предназначались для решения задач в основном вычислительного характера, не требующих быстрого получения результатов. *Главной целью и критерием эффективности систем пакетной обработки является максимальная пропускная способность, то есть решение максимального числа задач в единицу времени.* Для достижения этой цели в системах пакетной обработки используется следующая схема функционирования: в начале работы формируется пакет заданий, каждое задание содержит требование к системным ресурсам; из этого пакета заданий формируется мультипрограммная смесь, то есть множество одновременно выполняемых задач. Для одновременного выполнения выбираются задачи, предъявляющие отличающиеся требования к ресурсам, так, чтобы обеспечивалась сбалансированная загрузка всех устройств вычислительной машины; так, например, в мультипрограммной смеси желательно одновременное присутствие вычислительных задач и задач с интенсивным вводом-выводом. Таким образом, выбор нового задания из пакета заданий зависит от внутренней ситуации, складывающейся в системе, то есть выбирается «выгодное» задание. Следовательно, в таких ОС невозможно гарантировать выполнение того или иного задания в течение определенного периода времени. В системах пакетной обработки переключение процессора с выполнения одной задачи на выполнение другой происходит только в случае, если активная задача сама отказывается от процессора, например, из-за необходимости выполнить операцию ввода-вывода. Поэтому одна задача может надолго занять процессор, что делает невозможным выполнение интерактивных задач.

**Системы разделения времени.** В системах разделения времени каждой задаче выделяется только квант процессорного времени, ни одна задача не занимает процессор надолго, и время ответа оказывается приемлемым. Если квант выбран достаточно небольшим, то у всех пользователей, одновременно работающих на одной и той же машине, складывается впечатление, что каждый из них единолично использует машину. *Критерием эффективности систем разделения времени является не максимальная пропускная способность, а удобство и эффективность работы пользователя.*

**Системы реального времени** применяются для управления различными техническими объектами, такими, например, как научная экспериментальная установка, или технологическими процессами, такими, как прокатный стан, агрегатыковки, штамповки и т.п. Во всех этих случаях существует предельно допустимое время, в течение которого должна быть выполнена та или иная программа, управляющая объектом, в противном случае может произойти авария: спутник выйдет из зоны видимости; экспериментальные данные, поступающие с датчиков, будут потеряны; качество проката и поковок не будет соответствовать стандартам. Таким образом, *критерием эффективности для систем реального времени является их способность выдерживать заранее заданные интервалы времени между запуском программы и получением результата (управляющего воздействия).* Это время называется временем реакции системы, а соответствующее свойство системы – реактивностью. Для этих систем мультипрограммная смесь представляет собой фиксированный набор заранее разработанных программ, а выбор программы на выполнение осуществляется исходя из текущего состояния объекта или в соответствии с расписанием плановых работ.

Некоторые операционные системы могут совмещать в себе свойства систем разных типов; например, часть задач может выполняться в режиме пакетной обработки, а часть – в режиме реального времени или в режиме разделения времени. В таких случаях режим пакетной обработки часто называют фоновым режимом.

### **Тенденции в структурном построении ОС**

Как уже отмечалось выше, для удовлетворения требований, предъявляемых к современной ОС, большое значение имеет ее структурное построение. Операционные системы прошли длительный путь развития от монолитных систем к хорошо структурированным модульным системам, способным к развитию, расширению и легкому переносу на новые платформы.

К базовым концепциям построения ОС относятся:

- способ построения ядра системы – монолитное ядро;
- микроядерный подход.

### **ОС с монолитным ядром**

Операционную систему в общем случае можно представить как программную систему, состоящую из **ядра**, являющегося резидентным в памяти ЭВМ и обеспечивающего непосредственный интерфейс с аппаратной средой, а также обработку запросов на обслуживание активных в данный момент вычис-

лительных процессов  $\Pi_m$ , на основе того или иного механизма приоритетов (рис. 2.3) [1].

Таким образом, ядро содержит ту часть ОС, которая непосредственно обеспечивает функционирование основных компонент аппаратной среды ЭВМ, обрабатывает и обслуживает наиболее массовые запросы процессов и обеспечивает непосредственный интерфейс с аппаратурой.

Получив запрос на обслуживание от  $\Pi_k$  – процесса, ядро обрабатывает и обслуживает его на основе некоторого механизма приоритетов. Ядро ОС не является монолитом и состоит из целого ряда отдельных программ (планировщика, обработчика запросов, администратора ОС, драйверов внешних устройств и т.д.), управляющих различными ресурсами ЭВМ и в первую очередь вычислительными, управления, планирования.

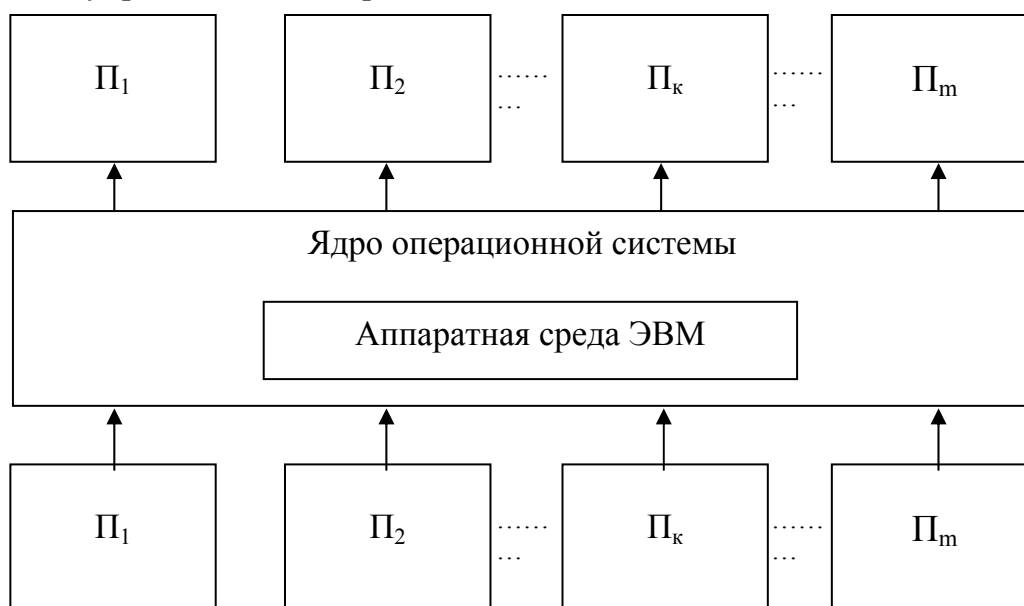


Рис. 2.3. Принципиальная организация операционной системы ЭВМ

Первые ОС, такие как MS DOS, использовали монолитное ядро, которое компонуется как одна программа, работающая в привилегированном режиме и использующая быстрые переходы с одной процедуры на другую, не требующие переключения из привилегированного режима в пользовательский и наоборот.

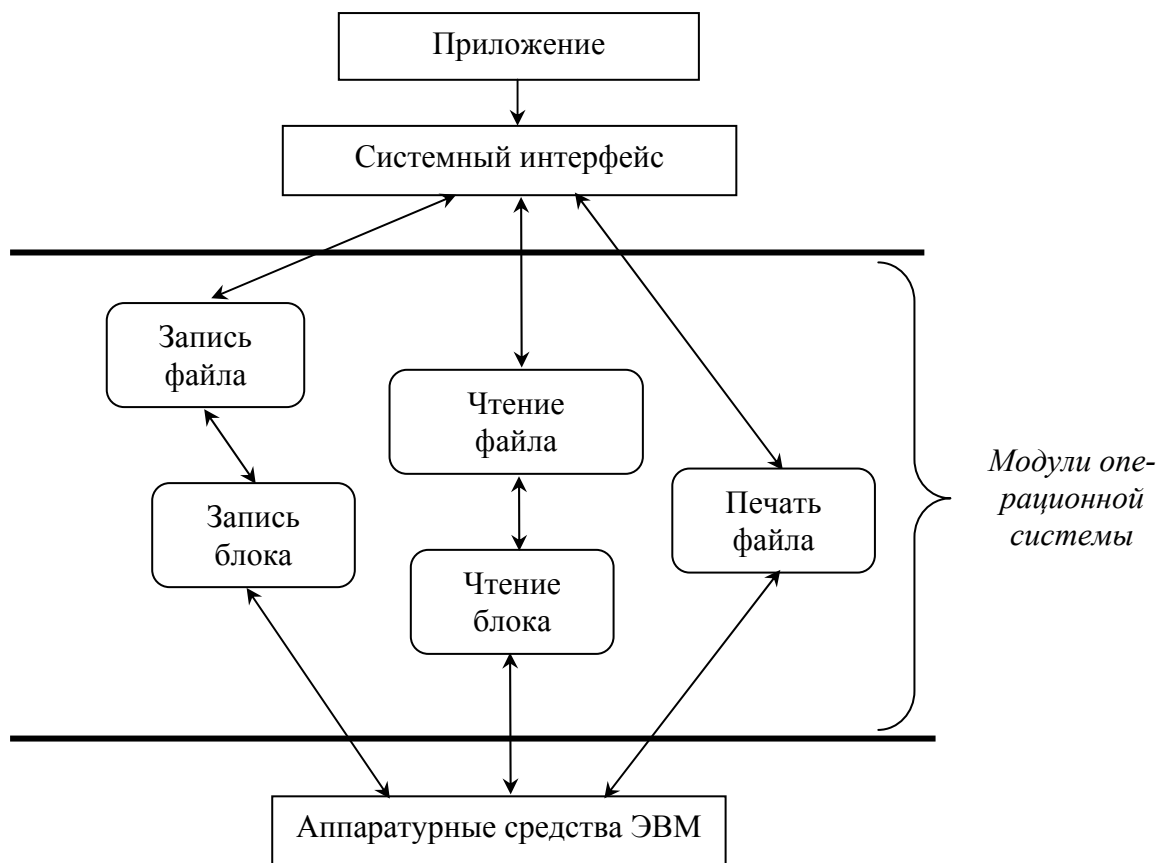
В общем случае «структура» монолитной системы представляет собой отсутствие структуры (рис. 2.4). ОС написана как набор процедур, каждая из которых может вызывать другие, когда ей это нужно. При использовании этой техники каждая процедура системы имеет хорошо определенный интерфейс в терминах параметров и результатов и каждая вольна вызвать любую другую для выполнения некоторой нужной для нее полезной работы.

Для построения монолитной системы необходимо скомпилировать все отдельные процедуры, а затем связать их вместе в единый объектный файл с помощью компоновщика (примерами могут служить ранние версии ядра Unix или Novell NetWare). Каждая процедура видит любую другую процедуру (в отличие

от структуры, содержащей модули, в которой большая часть информации является локальной для модуля и процедуры модуля можно вызвать только через специально определенные точки входа).

Однако даже такие монолитные системы могут быть немного структурированными. При обращении к системным вызовам, поддерживаемым ОС, параметры помещаются в строго определенные места, такие как регистры или стек, а затем выполняется специальная команда прерывания, известная как вызов ядра или вызов супервизора. Эта команда переключает машину из режима пользователя в режим ядра, называемый также режимом супервизора, и передает управление ОС. Затем ОС проверяет параметры вызова для того, чтобы определить, какой системный вызов должен быть выполнен. После этого ОС индексирует таблицу, содержащую ссылки на процедуры, и вызывает соответствующую процедуру. Такая организация ОС предполагает следующую структуру:

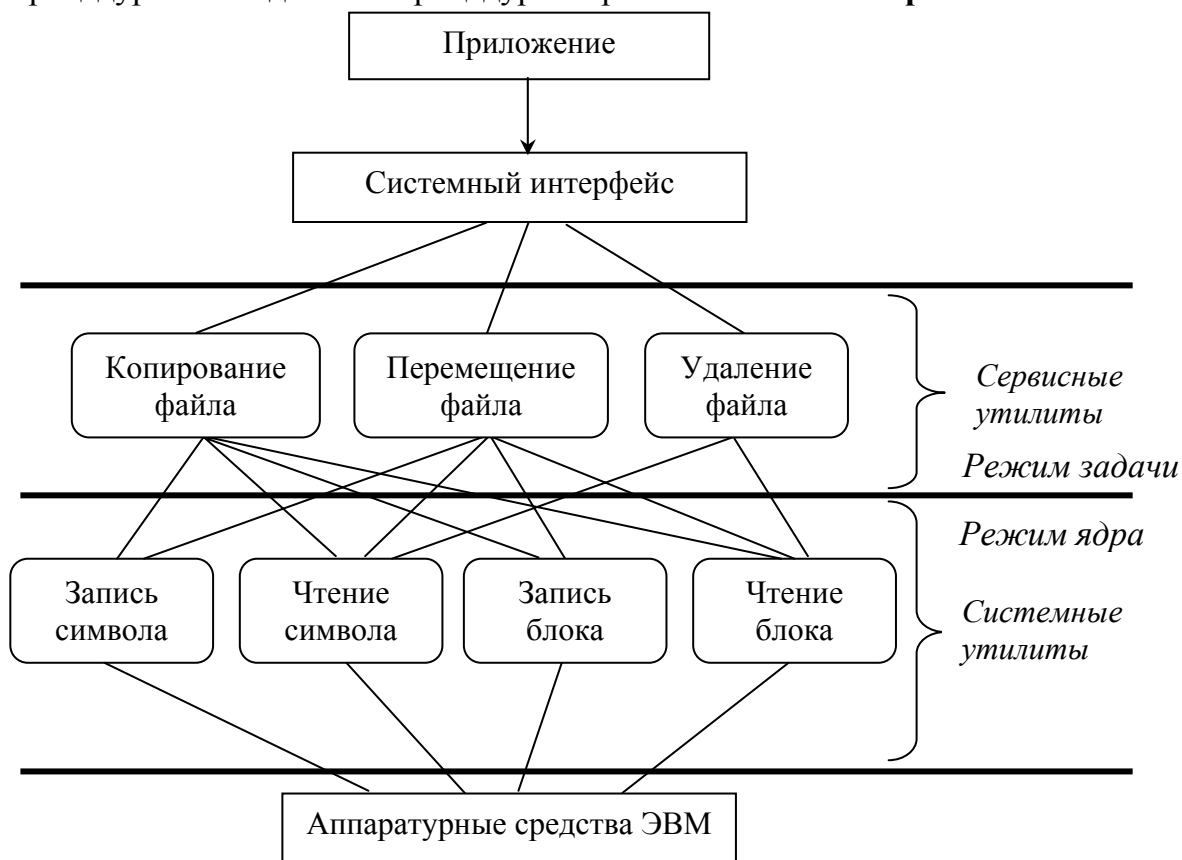
1. Главная программа, которая вызывает требуемые сервисные процедуры.
2. Набор сервисных процедур, реализующих системные вызовы.



**Рис. 2.4.** Монолитная структура операционной системы: все процессы работают последовательно, нет возможности планировать их выполнение

3. Набор утилит, обслуживающих сервисные процедуры.

В этой модели для каждого системного вызова имеется одна сервисная процедура. Утилиты выполняют функции, которые нужны нескольким сервисным процедурам. Это деление процедур на три слоя показано на **рис. 2.5**.



**Рис. 2.5.** Простая структуризация монолитной операционной системы: ядро осуществляет планирование и выполнение задач

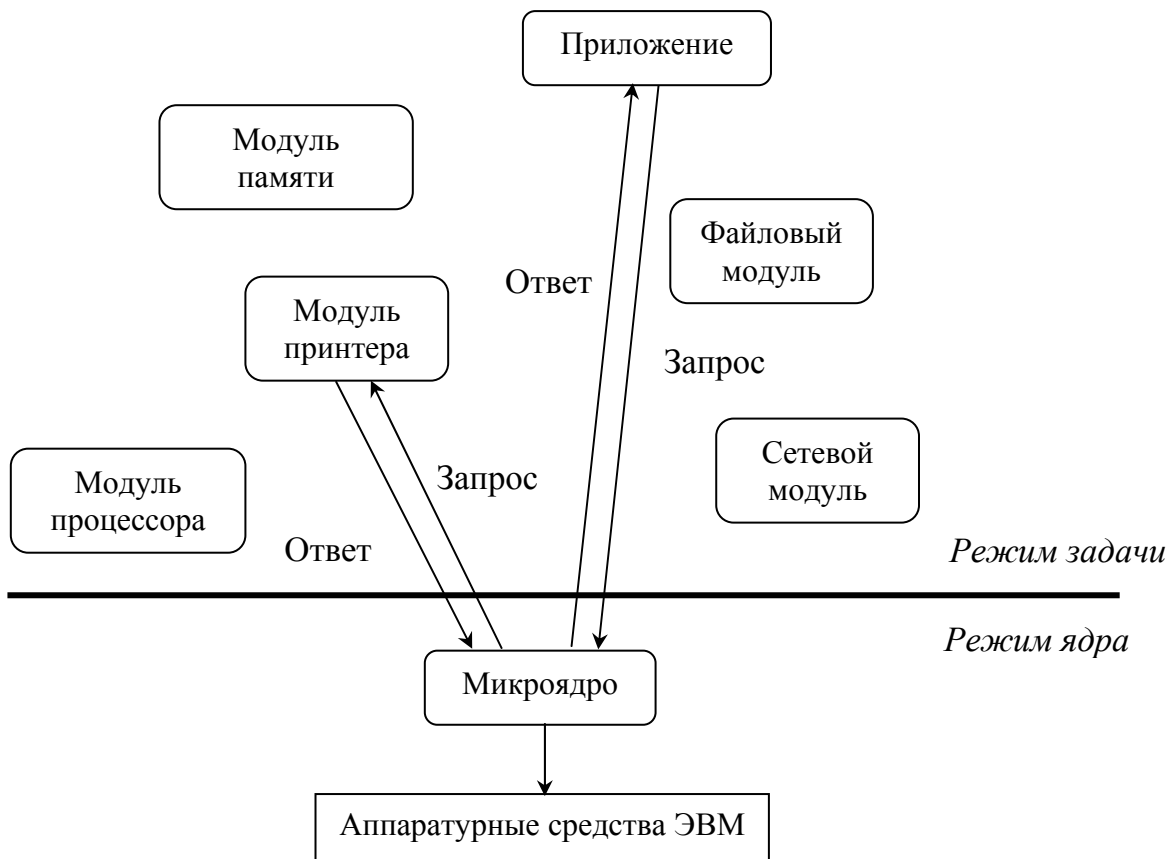
### ОС на основе микроядра

Альтернативой является построение **ОС на базе микроядра**, работающего также в привилегированном режиме и **выполняющего только минимум функций по управлению аппаратурой**, в то время как функции ОС более высокого уровня выполняют специализированные компоненты ОС, работающие в пользовательском режиме.

Подход с использованием микроядра заменил вертикальное распределение функций операционной системы на горизонтальное. Компоненты, лежащие выше микроядра, хотя и используют сообщения, пересылаемые через микроядро, взаимодействуют друг с другом непосредственно. Микроядро играет роль регулятора. Оно проверяет сообщения, пересылает их между модулями и предоставляет доступ к аппаратуре (**рис. 2.6**).



Архитектура «микроядро» означает, прежде всего, построение ОС по принципу модульности, а размер ядра играет не главную роль. При этом микроядро реализует базовые функции операционной системы, на которые опираются другие системные службы и приложения. Основная проблема при конструировании микроядерной ОС – это выделение функций системы, которые можно вывести из ядра. Функциональные возможности такой системы наращиваются не путем добавления новых функций в микроядро, а за счет подключения новых процессов, обеспечивающих дополнительный сервис. Поэтому поддержка файловой системы, сетевой обработки, графического интерфейса и других сервисов



**Рис. 2.6.** Структура модульной операционной системы:  
микроядро обладает минимальным набором системных утилит

не является функцией микроядра, как в других ОС. Эти сервисы обеспечиваются отдельными процессами. Лежащие вне микроядра компоненты, пользуясь средствами микроядра для обмена сообщениями, взаимодействуют непосредственно. Микроядро лишь проверяет корректность сообщений, организует их передачу и обеспечивает доступ к аппаратуре.

Целью создания ОС на основе микроядра является не желание сделать ее как можно меньше по размеру, а стремление обеспечить большую гибкость за счет модульной структуры. Небольшой размер микроядра является, несомненно, положительным, но все-таки побочным эффектом.

Микроядро реализует жизненно важные функции, лежащие в основе операционной системы. Это базис для менее существенных системных служб и приложений. Именно вопрос о том, какие из системных функций считать несущими

ществленными и, соответственно, не включать их в состав ядра, является предметом спора среди соперничающих сторонников идеи микроядра. В общем случае подсистемы, бывшие традиционно неотъемлемыми частями операционной системы – файловые системы, управление окнами и обеспечение безопасности – становятся периферийными модулями, взаимодействующими с ядром и друг с другом.

Главный принцип разделения работы между микроядром и окружающими его модулями – включать в микроядро только те функции, которым абсолютно необходимо исполняться в режиме супервизора и в привилегированном пространстве.

Высокая степень переносимости таких систем обусловлена тем, что весь машинно-зависимый код изолирован в микроядре, поэтому для переноса системы на новый процессор требуется меньше изменений и все они логически сгруппированы вместе. Технология микроядер является основой построения множественных прикладных сред, которые обеспечивают совместимость программ, написанных для разных ОС. Абстрагируя интерфейсы прикладных программ от расположенных ниже операционных систем, микроядра позволяют гарантировать, что вложения в прикладные программы не пропадут в течение нескольких лет, даже если будут сменяться операционные системы и процессоры.

Как мы уже отмечали, расширяемость также является одним из важных требований к современным операционным системам. Увеличивающаяся сложность монолитных операционных систем делала трудным, если вообще возможным, внесение изменений в ОС с гарантией надежности ее последующей работы. Ограниченный набор четко определенных интерфейсов микроядра открывает путь к упорядоченному росту и эволюции ОС.

Сегодня стало ясно, что имеется тенденция движения от монолитных систем к системам с использованием небольших ядер. Именно такой подход уже использовался компаниями QNX Software, OS9, в течение нескольких лет представляющими пользующиеся успехом операционные системы реального времени на основе микроядра.

### **2.2.2. Общая характеристика операционных систем общего назначения**

До недавнего времени вопрос о выборе операционной системы для персональных компьютеров вообще не ставился. Все пользователи находились в равных условиях – в среде MS DOS. Но с развитием аппаратных средств на сцену начали выходить операционные системы Windows NT/2000, OS/2, Unix – Linux.

*Рассмотрим кратко наиболее распространенные операционные системы.*

**MS DOS** – 16-разрядная диалоговая операционная система фирмы Microsoft появилась в 1981 г. Еще совсем недавно эта операционная система была установлена на подавляющем большинстве персональных компьютеров. Взаимодействие пользователя с операционной системой MS DOS построено по принципу диалога: пользователь набирает в командной строке на клавиатуре

нужную команду и нажимает клавишу Enter, после чего MS DOS выполняет введенную команду. С технической точки зрения Windows 3.x является операционной средой, а не системой, поскольку сама работает под управлением MS DOS.

**Microsoft Windows 95** (с 1995 года), **Windows 98** (с 1998 года), **Windows NT** (с 1993 года), **Windows 2000** – это многозадачные 32-разрядные операционные системы с графическим интерфейсом и расширенными системными возможностями. Windows 95 – первая полномасштабная операционная система семейства Windows, не требующая MS DOS. Windows 95/98 – интегрированная среда, обеспечивающая эффективный обмен текстовой, графической, звуковой и видеоинформацией между отдельными программами. В Windows 95/98, как и в других операционных системах, много действий самой операционной системы скрыты от пользователя, что, безусловно, упрощает общение пользователя с компьютером. Для работы достаточно наглядных операций с небольшими зрительскими образами файлов и папок.

В Windows 95/98 по сравнению с MS DOS для обозначения родственных понятий используется ряд новых терминов. В частности, вместо термина «программа» чаще используется термин *«приложение»*, который обозначает программу, выполняемую под управлением Windows. Понятие *«папка»* служит для обозначения хранилища документов и программ, структуру которых можно просмотреть с помощью проводника Windows или значка *Мой компьютер*. В предыдущей версии Windows и в MS DOS папки назывались каталогами.

В феврале 2000 года корпорация Windows представила на рынок семейство операционных систем, включающее в себя четыре продукта: ОС для настольных и мобильных компьютеров Windows 2000 Professional; серверную ОС для приложений начального уровня Windows 2000 Server; серверную ОС для приложений электронной коммерции и критически важных для бизнеса приложений Windows 2000 Advanced Server; серверную ОС для наиболее требовательных к надежности и масштабируемости решений Windows 2000 Datacenter Server.

**Операционная система Unix**, созданная корпорацией Bell Laboratory, и современная Unix-подобная операционная система **Linux**. Linux – это одна из современных операционных систем для персональных компьютеров и рабочих станций.

В России активно ведутся работы по локализации ОС Linux. Казалось бы, в нашей стране у бесплатной Linux есть неплохие шансы стать конкурентом ОС Windows, за которую приходится платить. Бесплатный продукт StarOffice и ряд других продуктов уже сегодня являются приемлемой альтернативой самым популярным программным продуктам Windows – Word, Excel Power Point. Тем не менее многие специалисты утверждают, что у Linux нет шансов в ближайшем будущем составить серьезную конкуренцию Windows на российском рынке. Несмотря на продолжающиеся споры специалистов по системам управления на тему «что лучше – Unix, Linux или Windows NT?», рынок однозначно сделал выбор в сторону последней. Решающим для быстрого роста популярности стала ее открытая архитектура и эффективные средства разработки приложений, что позволило многочисленным фирмам-разработчикам создавать программные продукты для решения широкого спектра задач.

Независимо от версии, общими для ОС Unix чертами являются:

- многопользовательский режим со средствами защиты данных от несанкционированного доступа;
- реализация мультипрограммной обработки в режиме разделения времени, основанная на использовании алгоритмов вытесняющей многозадачности (preemptive multitasking);
- использование механизмов виртуальной памяти и свопинга для повышения уровня мультипрограммирования;
- унификация операций ввода-вывода на основе расширенного использования понятия «файл»;
- иерархическая файловая система, образующая единое дерево каталогов независимо от количества физических устройств, используемых для размещения файлов;
- переносимость системы за счет написания ее основной части на языке C;
- разнообразные средства взаимодействия процессов, в том числе и через сеть;
- кэширование диска для уменьшения среднего времени доступа к файлам.

Данная операционная система реализует принцип открытых систем и широкие возможности по комплектованию в составе вычислительной системы разнородных технических и программных средств. Операционная система Unix основана, как и MS DOS, на работе с командной строкой, но, по сравнению с MS DOS, она значительно сложнее. Ядро этой операционной системы написано на языке высокого уровня (C и другие) и поэтому не привязано к конкретной платформе. Одним из наиболее существенных преимуществ данной ОС является даже не только ее переносимость на различные типы и классы ЭВМ, но и наличие в ней обширных библиотек и средств для разработки программного обеспечения, а также поддержка огромного множества прикладных программ, языков программирования. ОС Unix получила наибольшее распространение для суперкомпьютеров, рабочих станций и профессиональных персональных компьютеров, она чрезвычайно популярна в университетах, НИИ, компьютерных центрах и компьютерных сетях. Даже Internet уходит своими корнями в Unix. Среди специалистов существует мнение: «Когда программисту требуется защитить свой гарем персоналок, он призывает на помощь Unix». Однако система Unix, особенно для ПК, имеет ряд недостатков (высокую стоимость, недостаточно развитые средства диагностики, достаточно сложный интерфейс с пользователем и др.). Для удобства работы имеется графический интерфейс пользователя. Таким стандартом для графического пользовательского интерфейса, реализованного для всех Unix-систем, является X Window .

**Операционная система Solaris.** Solaris 2.x – это операционная система компании Sun, базирующаяся на Unix System V Release 4. Она включает:

- базовую операционную систему SunOS 5.x и систему сетевой поддержки ONC (Open Network Computing);
- набор вспомогательных утилит (диспетчер файлов, почту, печать, календарь и другие) DeskSet версии 3.x.

Компания Sun доработала исходный код Unix System V Release 4 в соответствии со своими потребностями. Новая ОС Solaris 2.x имеет несколько основных отличий от базовой операционной системы:

- реализована многонитевость;
- поддерживается симметричная многопроцессорная обработка;
- предусмотрен режим реального времени: допускаются прерывания процессов в системной фазе, что обеспечивает гарантированное время ответа на запросы.

Осенью 1994 года компания Sun Microsystems объявила о выпуске новой версии операционной системы Solaris для платформ SPARC и Intel x86 – Solaris2.4. Эта версия появилась в результате тщательного и долгого тестирования предыдущих версий. На сегодняшний день Solaris 2.4 является наиболее стабильной и качественной версией Solaris. Новое качество выражается не только в устранении всех замеченных в ходе тестирования недостатков, но и в более высокой производительности, чем у Solaris 2.x. В частности, увеличена средняя производительность при работе с СУБД, гораздо быстрее стали работать программы, реализующие пользовательский интерфейс. Важным свойством Solaris 2.4 является ее переносимость – программы, написанные для SPARC, могут выполняться на x86, и наоборот.

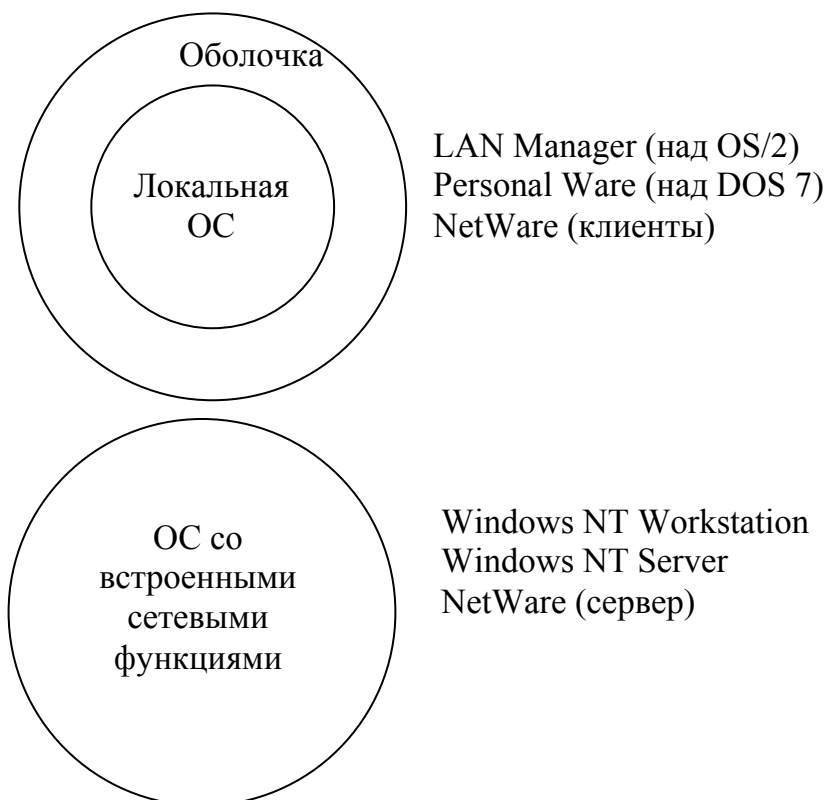
**Операционная система OS/2** (операционная система/2, иногда используется жаргон – полуось или ось пополам) разрабатывалась в содружестве корпорациями IBM и Microsoft с целью замены MS DOS и Windows для персональных компьютеров на основе системной прикладной архитектуры, ранее используемой для больших ЭВМ. В дальнейшем это содружество распалось, и в настоящее время корпорация IBM пытается представить OS/2 как конкурента ОС Windows. Хотя ОС OS/2 и эксплуатируется на ряде металлургических заводов, однако эта конкуренция явно не в пользу ОС OS/2, что обусловлено, в частности, недостаточностью программного обеспечения, разработанного для этой системы. OS/2 – это 32-разрядная многозадачная, однопользовательская, высоконадежная операционная система, обеспечивающая как текстовый, так и многооконный графический интерфейс пользователя. Отдельные версии имеют встроенный доступ в Internet.

**Сетевые операционные системы** – комплекс программ, обеспечивающий получение, обработку, передачу и хранение данных в сети.

Специфика ОС проявляется и в том, каким образом она реализует сетевые функции: распознавание и перенаправление в сеть запросов к удаленным ресурсам, передача сообщений по сети, выполнение удаленных запросов. При реализации сетевых функций возникает комплекс задач, связанных с распределенным характером хранения и обработки данных в сети: ведение справочной информации о всех доступных в сети ресурсах и серверах, адресация взаимодействующих процессов, обеспечение прозрачности доступа.

На характеристиках некоторых сетевых операционных систем мы остановимся позднее (при рассмотрении информационных сетевых технологий), а пока лишь отметим, что на практике сложилось несколько подходов к построению сетевых операционных систем (**рис. 2.7**).

Первые сетевые ОС представляли собой совокупность существующей локальной ОС и надстроенной над ней *сетевой оболочки*. При этом в локальную ОС встраивался минимум сетевых функций, необходимых для работы сетевой



**Рис. 2.7.** Варианты построения сетевых операционных систем

оболочки, которая выполняла основные сетевые функции. Примером такого подхода является использование на каждой машине сети операционной системы MS DOS (у которой, начиная с ее третьей версии, появились такие встроенные функции, как блокировка файлов и записей, необходимые для совместного доступа к файлам). Принцип построения сетевых ОС в виде сетевой оболочки над локальной ОС используется и в современных ОС, таких, например, как LANtastic или Personal Ware.

Однако более эффективным представляется путь разработки операционных систем, изначально предназначенных для работы в сети. Сетевые функции у ОС такого типа глубоко *встроены* в основные модули системы, что обеспечивает их логическую стройность, простоту эксплуатации и модификации, а также высокую производительность. Примером такой ОС является система Windows NT фирмы Microsoft, которая за счет встроенности сетевых средств обеспечивает более высокие показатели производительности и защищенности информации по сравнению с сетевой ОС LAN Manager той же фирмы (совместная разработка с IBM), являющейся надстройкой над локальной операционной системой OS/2.

### 2.2.3. Операционные оболочки

Как уже отмечалось, взаимодействие пользователя, в частности, с операционной системой MS DOS построено по принципу диалога: пользователь набирает на клавиатуре нужную команду и нажимает клавишу Enter, после чего MS DOS выполняет введенную команду. Такой способ взаимодействия не нагляден и недостаточно удобен. Действительно, для того чтобы, например, скопировать файл в другой каталог, надо набрать имя команды и имя каталога, а для этого надо помнить эти имена и не ошибиться в их наборе.

Кроме того, не все команды представляют пользователю возможность наглядного просмотра результатов своего выполнения. Куда проще «ткнуть» с помощью мыши в определенное место экрана, чтобы указать нужный файл, потом каталог, в который надо скопировать файл, а затем требуемое действие. Текстовые оболочки и позволяют работать с компьютером на таком наглядном уровне. Многие пользователи настолько привыкли к удобствам, предоставляемым своей любимой программой-оболочкой, что чувствуют себя без нее «не в своей тарелке».

*Операционные оболочки – это специальные программы, предназначенные для облегчения общения пользователя с командами операционной системы.*

Операционные оболочки имеют *текстовый и графический варианты интерфейса конечного пользователя.*

Наиболее популярны следующие виды *текстовых оболочек* ОС MS DOS:

- Norton Commander;
- Volkov Commander;
- Norton Navigator;
- Графическая операционная оболочка MS Windows для MS DOS и др.

*Графические оболочки*, т.е. набор средств для вывода изображений на экран и манипулирования ими, построения меню, окон на экране и т.д. Так, пользовательский интерфейс Windows 3.x использует графический режим работы монитора, т.е. является графическим. Основу графического интерфейса пользователя составляет организованная и хорошо продуманная система окон и других графических образов, использующих принцип «рабочего стола».

### 2.2.4. Сервисное программное обеспечение (утилиты)

Расширением базового программного обеспечения компьютера является набор сервисных, дополнительно устанавливаемых программ. Эти сервисные программы часто называют утилитами.

*Утилиты – программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров (диагностики, тестирования аппаратуры и программных средств, оптимизации использования дискового пространства, восстановления разрушенной на магнитном диске информации и т.д.).*

Эти программы можно классифицировать по функциональному признаку (см. рис. 2.2):

- *Программы диагностики работоспособности компьютера* позволяют проверить конфигурацию компьютера (количество памяти, ее использование, типы дисков и т.д.), а также проверить работоспособность устройств компьютера.

- *Антивирусные программы*, обеспечивающие защиту компьютера, обнаружение и восстановление зараженных файлов. Наиболее распространенными антивирусными программами являются Doctor Web, AntiViral Toolkit Pro (AVP), Aidstest – продукты двух отечественных фирм «Диалог-Наука» и «Лаборатория Касперского» и другие. Информацию по вирусам, программам, еженедельные обновления антивирусной базы можно получить из сети Internet.

- *Программы обслуживания дисков*, обеспечивающие проверку качества поверхности магнитного диска, сжатие дисков, дефрагментацию дисков, восстановление поверхности дисков, резервирование данных на внешних носителях и т.п. Среди таких программ широкое распространение получили Microsoft ScanDisk, входящий в состав Windows 95/98/2000 и NortonDiskDoctor, SpeedDisk из коллекции Norton Utilities и многие другие.

- *Программы архивирования данных (программы-упаковщики)*, которые обеспечивают процесс сжатия информации в файлах с целью уменьшения объема памяти для ее хранения. Применение программ-упаковщиков очень полезно при создании архива файлов, т.к. в большинстве случаев значительно удобнее хранить на дискетах предварительно сжатые программы. В настоящее время применяется несколько десятков программ-архиваторов, которые отличаются перечнем функций и параметрами работы. К числу наиболее популярных программ можно отнести: ARJ, PKZIP, RAR, WinZip, WinRAR и др. Обычно упаковка и распаковка файлов выполняются одной и той же программой, но в некоторых случаях это осуществляется разными программами (например, программа PKZIP производит упаковку файлов, а PKUNZIP – распаковку файлов).

- *Программы обслуживания сети*. К этой группе программных продуктов относятся средства резервного копирования в масштабах предприятия (IBM Amstar Distributed Storage Manager), сегментации, фильтрации и поиска неисправностей в сетях предприятий (Armor onsite Manager), программа, устраняющая барьер между сетями NetWare Windows NT (Microsoft NT File and Print Service for NetWare) и другие.

Разумеется, многообразие вспомогательных программ отнюдь не исчерпывается указанными выше типами программ. Рынок таких программ огромен и очень мобилен.

### 2.3. Пакеты прикладных программ

*Пакеты прикладных программ (ППП)* – комплекс взаимосвязанных программ, предназначенных для решения определенного круга задач из различных проблемных областей, снижающих трудоемкость и повышающих эффективность работы пользователя.



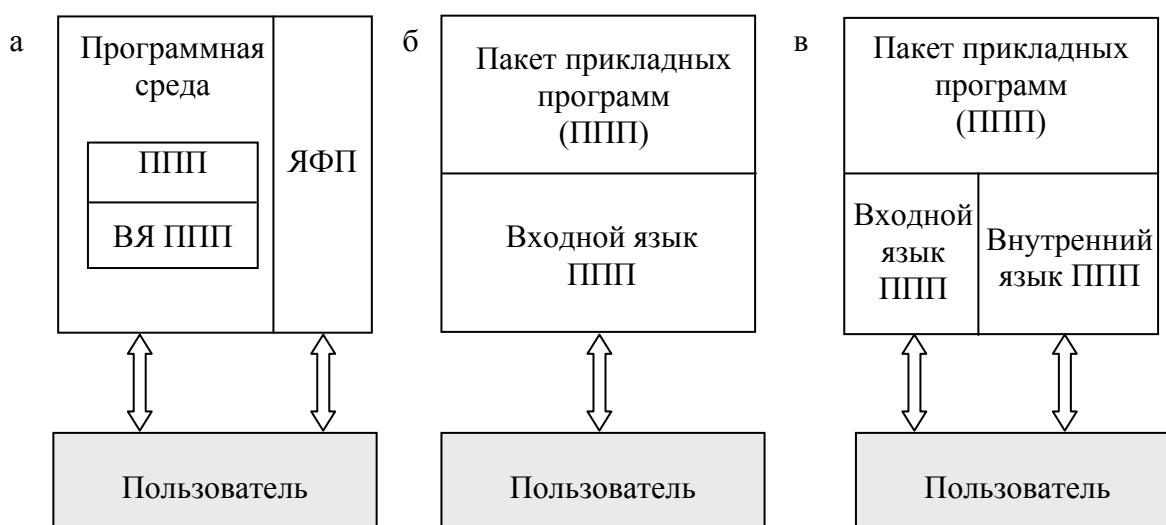
ППП создаются для решения наиболее массовых научно-технических, инженерных, экономических и многих других классов задач. Суть ППП состоит в максимальном упрощении интерфейса с ЭВМ проблемного пользователя.

Рассмотрим ППП несколько детальнее, основываясь на трех уровнях:

- 1) принцип организации ППП;
- 2) требования к профессиональной подготовке пользователя;
- 3) основные, современные группы ППП.

### 2.3.1. Принципы организации пакетов прикладных программ

Варианты формирования пакетов прикладных программ проиллюстрированы на **рис. 2.8** [1].



**Рис. 2.8.** Варианты организации пакетов прикладных программ

Одной из характерных черт современных ППП является использование **принципа синтеза**, т.е. (формирования) рабочих программ из подпрограмм на основе содержательного описания задачи на проблемно-ориентированном языке, близком к понятиям и терминам проблемной области пользователя. При таком подходе пользователь посредством специального языка формирования пакета (ЯФП) содержательно описывает отдельную задачу или класс задач, требующих решения, формируя программу генерации конкретного ППП из набора средств фиксированной или расширяемой программной среды (**рис. 2.8, а**). На основе данной программы генерируется ППП для конкретного приложения со своим входным языком (ВЯ) общения с пользователем. После создания ППП пользователь работает с ним в процессе решения своих задач.

Как правило, описанный принцип организации используется для *проблемно-ориентированных* ППП, когда решаемый ими круг задач достаточно четко определен и задачи связаны какими-либо характеристиками, например численными методами, статистическим анализом, моделированием в конкретной предметной области и т.д. Данная технология в общем случае предполагает наличие двух уровней пользователей – *системного*, генерирующего ППП с заданной предметной областью, и *проблемного*, использующего заданный ППП по-

средством его входного языка, близкого к понятиям предметной области или к естественному языку. В случае достаточно простой программной среды, узко ориентированного класса решаемых задач и простого языка формирования пакета (ЯФП) функции системного и проблемного пользователя может совмещать одно и то же лицо.

В отличие от рассмотренной организации ППП, обеспечивающей их гибкость и адаптируемость к предметной области и выдвигающей ряд требований к профессиональному уровню пользователя, для *массового пользователя* используется, как правило, *жесткая организация ППП*, не позволяющая ему модифицировать пакет (**рис. 2.8, б, в**).

Жесткая организация в сочетании с предметно-ориентированным *входным языком пакета (ВЯП)* (см. **рис. 2.8, б**) ориентирована на относительно небольшие по объему пакеты либо пакеты специальной направленности, но массового применения. Примером такой организации могут служить простые пакеты текстовых редакторов, специальные пакеты и т.п.

Такая организация имеет два основных исполнения:

- 1) высокий логический уровень *входного языка пакета (ВЯП)*, ориентированного на пользователя и предметную область, и *внутреннего языка пакета (ВЯ)*, позволяющего создавать в его среде *модули*, обеспечивающие функции расширения пакета (см. **рис. 2.8, в**);
- 2) интерфейс с пакетом обеспечивается только на уровне проблемно-ориентированного *входного языка пакета (ВЯП)* (см. **рис. 2.8, б**).

В обоих случаях практика создания входного языка пакета (ВЯП) для пакета использует два основных подхода:

- 1) создание языка на основе уже существующего языка высокого уровня, который, как правило, является и языком реализации пакета;
- 2) разработку оригинального входного языка.

При первом подходе существенно упрощается реализация входного языка, однако для непрофессионального пользователя возникают дополнительные сложности по освоению пакета.

Второй подход требует в ряде случаев значительно больших затрат на разработку входного языка, однако позволяет сделать его легким в освоении и использовании (язык меню, языки запросов, язык диалоговой графики и др.).

При этом спектр входного языка пакета (ВЯП) лежит в весьма широком диапазоне. Например, игровые пакеты имеют высокопонятийный графический интерфейс и не требуют особого освоения; пакеты текстовых редакторов также имеют развитый язык систем меню; проблемно-ориентированные пакеты (например, математические) обеспечены входным языком, ориентированным на знакомого с математическим языком пользователя.

Таким образом, согласно своей организации (**рис. 2.8**) пакеты допускают различные типы и уровни расширения со стороны пользователя: от возможности полной генерации пакета на конкретные условия применения (**рис. 2.8, а**) до отсутствия какой-либо возможности расширения пакета (**рис. 2.8, б**).

*Большинство современных ППП допускают расширения*, из которых для проблемного использования наиболее естественным способом является создание

библиотек, документов, программ, модулей, макросов на внутреннем языке пакета, решающих задачи в некоторой предметной области.

Так, в среде Microsoft Office можно осуществлять макропрограммирование с использованием макросов. Макропрограммирование – это разработка программ с использованием макросредств.

**Макрос** (сокращенное обозначение макрокоманд) – это способ автоматизации серии часто встречающихся операций. В простейшем случае макрос создается запоминанием ряда операций, которые затем воспроизводятся. Макрос записывается и редактируется на специальном языке программирования Visual Basic. Во время записи макроса создается программа на этом языке, содержащая необходимые команды.

**Visual Basic for Applications (VBA)** – развитая система визуального программирования для создания прикладных программных средств в среде Microsoft Office. Детальное описание макросов и команд Visual Basic можно прочитать, в частности, в меню Справка программ Word, Excel, Access пакета Microsoft Office, а также в книге [47].

Перспективным дальнейшим направлением развития интегрированных ППП являются программные среды типа оболочки ОС. По желанию пользователь сам определяет набор составляющих пакетов и собирает их в интегрированный ППП в своей операционной среде. Такая операционная оболочка является программным интерфейсом между выбранным пакетом и ОС, предоставляя пользователю не готовые интегрированные ППП, а средства для их создания.

### 2.3.2. Основные группы пакетов прикладных программ

ППП можно условно классифицировать по четырем основным группам:

- 1) общего назначения;
- 2) проблемно-ориентированные;
- 3) расширяющие функции ОС;
- 4) интегрированные.

**ППП общего назначения** – ориентированы на широкий круг пользователей в различных проблемных областях, позволяя автоматизировать наиболее часто используемые виды работ. К пакетам такого типа относятся обработка текстовой информации, деловой графики, электронных таблиц и т.д. При этом следует иметь в виду, что функции пакета одного типа могут перекрывать функции пакета другого типа. Однако каждый из этих пакетов все-таки в большей мере акцентирован на обработку информации определенного вида.

**Проблемно-ориентированные ППП** имеют относительно узкое применение, используя особые методы представления и обработки информации. Это пакеты для экспертных систем, математические пакеты, статистические, издательские системы, управления производством, работы бухгалтерии, обработки фотографий, финансовой деятельности, управления персоналом (кадрового учета), управления материальными запасами (учета материалов), учета и управления транспортными средствами и т.п.

**Расширяющие функции ОС пакеты** обеспечивают, в частности, сопряжение ЭВМ с унифицированными приборными интерфейсами, научными приборами и установками.

**Интегрированные ППП** объединяют основные функции ряда пакетов, как правило, общего назначения. В таких пакетах используются не только единый формат файлов, но и методы доступа к файлам отдельных пакетов, т.е. обеспечивается полная информационная совместимость составляющих совокупность пакетов. Современные интегрированные пакеты для массового применения включают такие основные функции, как текстовый редактор, электронные таблицы, деловую графику, системы управления базами данных и телекоммуникацию. Типичным средством данного класса является известный пакет MS Office.

В настоящее время количество ППП лавинообразно растет. Даже их краткий обзор стал практически невозможным. Наряду с этим многие пакеты пересекаются на функциональном уровне, даже если не иметь в виду интегрированных пакетов. Поэтому дать сколько-нибудь удовлетворительную большинству пользователей классификацию ППП не представляется возможным. Вместе с тем можно выделить ряд классов пакетов, наиболее широко используемых в ПК на металлургических заводах.

Кратко охарактеризуем каждую из указанных групп ППП (табл. 2.1), приводя конкретные примеры средств, ориентированных на IBM-совместимые ПК.

Таблица 2.1

### Основные группы пакетов прикладных программ

№ п/п	Назначение ППП
1	Обработка текстовой информации (текстовые редакторы, процессоры и др.)
2	Обработка графической информации
3	Обработка информации табличного типа (электронные таблицы)
4	Система управления базами данных (СУБД)
5	Специальные математические пакеты (средства численных и символьных вычислений)
6	Экспертные системы (ЭС), системы представления знаний и базы знаний
7	Коммуникационные пакеты для обеспечения сетевой работы компьютеров
8	Пакеты, реализующие CASE-технологии
9	Интегрированные пакеты общего назначения

**1. Обработка текстовой информации** различного типа и назначения составляет одну из наиболее массовых групп задач на ЭВМ. Пакеты данной группы являются одними из наиболее часто используемых на ПК и позволяют проводить набор основных процедур работы с текстовой информацией:

- создание и корректировка текстовых документов;
- компоновка текстовой и графической информации;
- форматирование выходных документов, предварительный их просмотр;
- синтаксический контроль и т.д.

Среди пакетов этой группы можно выделить три подгруппы:

- общего назначения (например, MS Word);
- специального назначения (например, News Master – осуществляет редактирование деловой и коммерческой информации, Easy Flow – используется для документирования технической документации, TEX);
- издательские системы (например, Page Maker, Quark XPress и др.).

**2. Графические пакеты** – предназначены для обработки различного рода и назначения графической и иной изобразительной информации (рисование, создание и использование библиотек рисунков различного назначения, черчение в двух и трех измерениях, создание иллюстраций и т.д.). Среди средств этой группы можно проводить довольно разветвленную классификацию, учитывая бурное развитие и многоаспектность использования графической информации. Наиболее доступными и популярными являются пакеты: Paint, Power Point, Freelance Graphics for Windows и целый ряд других, позволяющих проводить весьма разнообразную обработку графической информации для различных приложений. Профессионалам из редакций газет и журналов или рекламных агентств требуются более мощные редакторы типа Corel Draw или Adobe PhotoShop. Для автоматизации проектно-конструкторских работ фактически стандартом является пакет AutoCAD, представляющий собой мощную систему автоматизированного проектирования (САПР) для создания и редактирования сложных графических объектов, черчения, в первую очередь промышленных приложений. Отметим, что освоение этих средств требует от пользователя большого трудолюбия и настойчивости.

### **3. Обработка информации табличного типа (электронные таблицы)**

*Электронная таблица* – компьютерный эквивалент обычной двумерной таблицы, в клетках (ячейках) которой записаны данные различных типов: тексты, даты, формулы, числа и другая информация. Электронная таблица – одна из самых распространенных и мощных информационных технологий для профессиональной работы с данными. Для управления электронной таблицей созданы специальные программные продукты – *табличные процессоры*. Идея создания электронной таблицы возникла у студента Гарвардского университета (США) Дэна Бриклина в 1979 году. Выполняя скучные вычисления экономического характера с помощью бухгалтерской книги, он и его друг Боб Франкстон, который разбирался в программировании, разработали первую программу электронной таблицы, названную ими VisiCalc. Среди средств данной группы в настоящее время хорошо зарекомендовали себя такие пакеты, как Super Calc,

Quattro Pro, Excel, Lotus и др. В современных электронных таблицах Excel обеспечиваются такие основные операции, как:

- хранение в памяти и просмотр на экране дисплея таблиц большой размерности;
- размещение в клетках таблицы информации и операции (в том числе математические) над содержанием клеток;
- графическое отображение информации и многие другие.

Большинство электронных таблиц имеют средства создания графиков и диаграмм, средства их редактирования и включения их в нужное место. Кроме того, в них имеется большое число встроенных функций – математических, статистических, имеется возможность решать задачи условной и безусловной, дискретной и непрерывной оптимизации. Сервисные функции табличного процессора Excel позволяют проверять орфографию текста, осуществлять защиту данных от чтения или записи. Возможно создание диалоговых окон или обращение к динамическим библиотекам. Заметим, что в табличном процессоре Excel есть средство создания макросов – *Visual Basic for Applications*. Все табличные процессоры поддерживают формат базы данных и представляют собой удобные средства работы с ними. Как уже отмечалось ранее, функциональное наполнение пакетов этой группы, как правило, является составной компонентой интегрированных ППП общего назначения. Подробнее с методикой работы в среде Excel можно познакомиться в книге [40].

**4. Системы управления базами данных (СУБД).** СУБД позволяют хранить различного рода информацию, обеспечивая доступ к ней как локально, так и в сетевых режимах. В настоящее время данный класс средств достаточно многочислен, и только для IBM-совместимых ПК существует около 50 известных пакетов общего назначения, настольных СУБД, среди которых можно отметить такие, как FoxPro, Paradox, Access. Для управления большими базами данных и их эффективной обработки на серверах баз данных устанавливаются SQL Server, SQL Base, Oracle Server, Informix, SyBase и другие программные продукты. Средства данной группы мы будем обсуждать в детальнейшем, в разделе, посвященном СУБД, а пока лишь заметим, что после появления в 1992 году MS Access за последующие пять лет было продано более пяти миллионов лицензионных копий этой программы. Это дает основание считать ее сегодня самой популярной и наиболее доступной системой управления базами данных для персональных компьютеров. Access имеет средства программирования – макросы и язык Visual Basic, что позволяет «вдохнуть жизнь» в приложения.

**5. Специальные математические пакеты (средства численных и символьных вычислений)** ориентированы на достаточно массовое применение и обеспечивают решение задач в математике, технике, статистике и других областях. Потребность в математических расчетах по-прежнему велика в нашем обществе, идущем сквозь тернии к рыночной экономике. Одной из основных областей применения компьютеров и поныне являются математические и научно-технические расчеты. Само по себе появление компьютеров не упрощало математические расчеты, а лишь позволяло резко повысить скорость их выполнения и сложность решаемых задач. Пользователям компьютеров, прежде чем начинать такие расчеты, необходимо было изучить сами компьютеры, языки про-

граммирования и довольно сложные методы вычислений, применить и представить под свои цели программы для решения расчетных задач на языках программирования. Поневоле инженеру, физику, химику, металлургу и математику приходилось становиться программистом, порою довольно посредственным.

Такая порочная практика изменилась лишь после появления интегрированных математических программных систем для научно-технических расчетов, наиболее распространенные из них: MatLAB, MathCAD, Mathematica; Maple, Statistika, пакет оптимизации QSB и т.д. [18, 19].

Большое количество подобных разработок свидетельствует о значительном интересе к ним во всем мире. Математики, физики, ученые, инженеры из других смежных отраслей науки давно мечтали о математически ориентированном языке программирования для записи алгоритмов решения математических и научно-технических задач в наиболее удобной, компактной и доступной для понимания форме. Для этого они пытались приспособить различные языки программирования высокого уровня. Но их попытки так и не увенчались успехом – программы на этих языках, увы, ничем не напоминали привычные математические и физические символы и формулы, с которыми все привыкли работать и с помощью которых описываются решения математических задач.

Математические пакеты такого рода являются системами, в которых описание решения математических задач задается с помощью привычных математических формул и знаков. Такой же вид имеют и результаты вычислений.

Как правило, они имеют удобный пользовательский интерфейс (совокупность средств общения с пользователем). У этих систем, как правило, имеются и эффективные средства типовой научной графики, они просты в применении и интуитивно понятны. Одним словом, этим системы ориентированы на массового пользователя, от ученика начальных классов до академика. О системе с такой вычислительной мощностью, как последняя версия MathCAD 8.0 PRO и MapleV Release 5, еще десять лет назад не могли мечтать даже разработчики уникальной научной и космической аппаратуры.

Особый интерес представляют встроенные в систему электронные книги, содержащие справки и примеры применения системы по ряду разделов математики, механики, физики, электротехники. Справки содержат математические формулы и иллюстрации, можно выделить нужную справку – формулу или рисунок – и перенести ее в текст документа.

По ряду пакетов этой группы – MathCAD, Mathematica, Maple, Statistika – издан целый ряд книг различного уровня и назначения, рассчитанных на различный круг пользователей. В этих книгах приведены достаточно подробные описания этих популярных пакетов, особенности их эксплуатации и применения для решения различного рода задач математического характера.

Заметим, что математические пакеты – ценный инструмент, но лишь в квалифицированных руках, и он не избавляет, а, наоборот, усиливает необходимость изучения математики. Действительно, освоение математических пакетов может создать у пользователя иллюзию освоения самой математики. Однако следует помнить, что инструмент не заменяет компетентность. Никакие красочные меню не освобождают пользователя от понимания сути математических методов, реализованных в таких пакетах. Это своего рода искусственный интел-

лект, который только в умелых руках пользователя, обладающего естественным интеллектом, может принести существенную пользу.

Примеры использования специальных математических пакетов при математическом моделировании технологических процессов в металлургии мы рассмотрим позднее, в главе, посвященной новым информационным технологиям в металлургии.

**6. Экспертные системы (ЭС).** В настоящее время в обращении находится свыше 6000 ЭС и инструментальных средств для их построения, постоянно расширяется их рынок и ведутся интенсивные работы по совершенствованию их архитектуры и возможностям. При этом наибольший интерес представляют так называемые «оболочки» и «программные обстановки».

Под «оболочками» понимают «пустые» версии существующих экспертных систем, т.е. готовые экспертные системы без базы знаний. Достоинство оболочек в том, что они вообще не требуют работы программистов для создания готовой экспертной системы. Требуются только специалисты в предметной области и инженеры по знаниям для заполнения базы знаний. В настоящее время разработан широкий спектр оболочек экспертных систем. Перечислим наиболее известные и распространенные сейчас «оболочки» ЭС. Примером таких оболочек могут служить EMYCIN (Empty MYCIN – пустой MYCIN) и VP Expert, которые представляет собой незаполненные экспертные системы.

*Программные обстановки* для разработки экспертных систем представляют собой высокоуровневые системы поддержки автоматизированной разработки и создания проблемно-ориентированных ЭС. Они, как правило, содержат богатый спектр языков представления знаний. Разработчик экспертной системы, пользуясь набором разнообразных средств организации вывода в ЭС, может сконструировать свою проблемно-независимую оболочку. Наиболее известной программной обстановкой является система KEE. Система имеет большие возможности, обладает средствами тестирования и отладки знаний, прекрасным пользовательским интерфейсом и графикой. Система Nexpert Object является мощным средством разработки ЭС, способным конкурировать с такими системами, как Gold Work и KEE, и имеющим подобную организацию базы знаний. Система работает в среде MS Windows, имеет интерфейс с языками программирования, хорошие возможности работы с базами данных.

К сожалению, для отечественных пользователей этот класс пакетов малодоступен, т.к. стоимость их чрезвычайно высока и практически полностью отсутствуют русифицированные версии этих программных продуктов.

**7. Коммуникационные ППП** предназначены для организации взаимодействия пользователя с удаленными абонентами или информационными ресурсами сети. Так, в условиях развития глобальной информационной сети Internet, идеология которой сейчас используется в локальных сетях, появился класс программного обеспечения – *браузеры*, средства создания WWW страниц и др. Более подробно вопросы компьютерной сетевой обработки информации и функции используемых при этом пакетов мы рассмотрим позднее в разделе, посвященном компьютерным сетям.

**8. Пакеты, реализующие CASE (Computer Aided Software Engineering)–технологии** применяются при создании сложных информацион-



ных систем, обычно требующих коллективной разработки проекта, в котором участвуют различные специалисты: системные аналитики, проектировщики и программисты.

*Под CASE-технологией понимается совокупность средств автоматизации разработки информационной системы, включающей в себя методологию анализа предметной области, проектирования, программирования и эксплуатации информационных систем.*

Инструментальные средства CASE-технологии применяются на всех этапах жизненного цикла системы (от анализа и проектирования до внедрения и сопровождения), значительно упрощая решение возникающих задач.

CASE-технология позволяет отделить проектирование информационной системы от собственно программирования и отладки: разработчик системы занимается проектированием на более высоком уровне, не отвлекаясь на детали. Это позволяет не допускать ошибок уже на стадии проектирования и получить более совершенные программные продукты.

Коллективная работа над проектом предполагает обмен информацией, контроль выполнения задач, отслеживание изменений и версий, планирование, взаимодействие и управление. Фундаментом реализации подобных функций чаще всего служит общая база данных проекта, которую обычно называют *репозитарием*.

По существу, репозитарий – это информационный архив, где хранятся сведения о процессах, данных и связях объектов в разрабатываемом приложении.

В различных CASE-технологиях репозитарий реализуется по-разному и может содержать описания и модели данных, а также правила их обработки. Репозитарий является важнейшим компонентом набора инструментальных средств CASE и служит источником информации, необходимой для автоматизации построения проектируемой системы и генераций приложений.

Современные CASE-технологии успешно применяются для создания информационных систем различного класса. Они обычно имеют достаточно высокую стоимость и требуют длительного обучения и кардинальной реорганизации всего процесса создания информационных систем. Заметим, что большинство современных серьезных программных проектов осуществляются именно с их помощью. Пример использования CASE-технологии мы рассмотрим далее, в главе, посвященной проектированию баз данных.

**9. Интегрированные ППП** – набор нескольких программных продуктов, функционально дополняющих друг друга, поддерживающих единые информационные технологии, реализованные на общей вычислительной и операционной платформе. Организационно такие пакеты представляют собой либо логическое объединение пакетов под управлением некоторого координирующего их работу модуля, либо выполняются в виде интегрированной системы функций пакетов общего назначения. На заре эры персональных компьютеров каждое из приложений представляло собой независимую программу. Обычно пользователь покупал программу обработки текстов у одной компании, программу обработки электронных таблиц – у другой, а систему управления базой данных – у третьей. Вероятность того, что эти приложения имеют одинаковый интерфейс и функ-

ционируют на базе одинаковых принципов, была низкой, что затрудняло изучение каждой новой программы. Причина такой обособленности приложений отчасти заключалась в самой однозадачной операционной системе, в которой она работала, – MS DOS.

Современные процессоры и операционные системы обладают возможностями реальной многозадачности – они позволяют создавать среду, в которой одновременно могут функционировать несколько приложений, совместно используя микропроцессор. Многозадачность позволила создавать пакеты, которые состоят из программ, имеющих одинаковый интерфейс, и основаны на одних и тех же принципах работы.

Современные интегрированные пакеты для массового применения включают следующие основные функции:

- текстовый редактор;
- электронные таблицы;
- деловую графику;
- СУБД;
- телекоммуникацию.

Компоненты интегрированных пакетов могут работать изолированно друг от друга, но основные достоинства интегрированных пакетов проявляются при их разумном сочетании друг с другом. Пользователи интегрированных пакетов имеют унифицированный для различных компонентов интерфейс, тем самым обеспечивается относительная легкость процесса их освоения. Типичный пример интегрированного пакета, как мы уже отмечали, – MS Office, который включает в себя:

- Word – приложение для обработки текстов;
- Excel – программа обработки электронных таблиц;
- Access – программа обработки баз данных;
- Outlook – комбинация программ обработки электронной почты и менеджмента личной информации;
- PowerPoint, позволяющий создавать презентации для выставок, продаж, семинаров или для случаев, когда необходимо представлять информацию группе людей с применением графики, текста и диаграмм.

Имеются, конечно, и другие типы прикладных программ; например, имеется большое количество ППП, поддерживающих автоматизированные рабочие места (АРМ) самого различного назначения: для научных работников, банковских служащих, бухгалтеров, сотрудников налоговой службы, управления персоналом и т.д.

Так, интегрированный пакет Scientific Work Place (SWP) – «рабочее место ученого» удачно объединяет несколько полезных инструментов:

- подобный Word текстовый редактор;
- текстовый редактор научных документов с возможностью набора математических зависимостей в привычной для ученого форме;
- подмножество пакета Maple V, позволяющее выполнять аналитические преобразования и численные расчеты.

На рынке программных продуктов имеется масса и других программных средств, в частности, мультимедиа, офисных ППП: органайзеров (планировщиков), программ-переводчиков, средств проверки орфографии и распознавания текста и т.п. Ввиду обилия и разноплановости этих средств дать им какую-либо характеристику не представляется возможным.

В заключение рассмотрения пакетов общего назначения отметим, что для успешного ориентирования на рынке ППП, особенно общего назначения, очень полезно использовать услуги Internet.

## 2.4. Особенности программного обеспечения технологических процессов

Естественно, специфика промышленных применений в металлургии наложила свой отпечаток на используемое программное обеспечение. Основные требования к программному обеспечению сводятся к следующему:

- Первым требованием является *надежность программного обеспечения*. Действительно, одно дело, когда у вас «зависает» редактор текста в офисе, а другое дело, когда неправильно работает программа, управляющая прокатным станом, разливкой стали или загрузкой материалов в доменную печь.

- Вторым требованием является быстрое реагирование на какие-либо внешние события или изменения в параметрах управляемых процессов. Системы, работающие в соответствии с этим требованием, относятся к *системам реального времени*.

- Третьим требованием, часто предъявляемым к программному обеспечению систем управления, является *многозадачность*. Это требование проистекает из-за сложной и многоуровневой природы управляемых процессов в металлургии, когда необходимо одновременно реализовать сложные алгоритмы управления различными подсистемами реального объекта. Каждая задача выполняет свою долю работы по управлению объектом, и все они делят между собой ресурсы вычислительной системы в зависимости от своего приоритета и от внешних и внутренних событий, связанных с конкретной задачей.

Названным требованиям должны удовлетворять все уровни программного обеспечения (BIOS, ОС, прикладные программы).

Ведущие фирмы-поставщики ориентируются в основном на рынок офисных компьютеров и не применяют специальных мер для обеспечения требований индустриальных приложений. Основная опасность при обращении к BIOS – это возможность запрета прерываний на достаточно долгое время, в результате чего может быть пропущена важная информация от быстродействующих датчиков и телекоммуникационных устройств. В связи с этим имеются фирмы, предлагающие ориентированные на приложения «жесткого» реального времени, а с другой стороны, многие операционные системы минимизируют взаимодействие с BIOS.

Задачи реального времени составляют одну из сложнейших и крайне важных областей применения вычислительной техники. С уверенностью можно сказать, что ссылки на красивое сочетание «реальное время» стали часто исполь-

зовать в специализированной печати. С не меньшей уверенностью можно сказать, что смысл этого термина трактуется специалистами по-разному.

Говорят, что информационная система работает в реальном времени, если ее быстродействие адекватно скорости протекания физических процессов на объектах контроля или управления.

Система управления реального времени должна собрать данные, произвести их обработку в соответствии с заданными алгоритмами и выдать управляющие воздействия за такой промежуток времени, который обеспечивает успешное решение поставленных перед системой задач.

Для системы управления прокатным станом время реакции системы должно быть в пределах нескольких миллисекунд, а для систем контроля за окружающей средой металлургического предприятия – несколько минут. Но, тем не менее, оба эти примера – из области задач реального времени.

Следовательно, практически все системы промышленной автоматизации являются системами реального времени. Принадлежность системы к классу систем реального времени никак не связана с ее быстродействием и во многом определяется динамическими свойствами самого технологического процесса и требованиями к качеству управления технологическими параметрами. Интуитивно понятно, что быстродействие системы реального времени должно быть тем больше, чем выше скорость протекания процесса на объекте контроля и управления.

Задачи реального времени предъявляют такие требования к аппаратному и программному обеспечению, как надежность, своевременная реакция на внешние события, высокая пропускная способность передающей среды и т.д. Для выполнения этих требований имеются специальные операционные системы реального времени.

*Под операционной системой реального времени понимают систему, которая выполняет свои функции, а также отвечает на внешние асинхронные события за предсказуемый промежуток времени.*

Главное требование, предъявляемое к операционным системам общего назначения, заключается в том, что они должны обеспечить оптимальное разделение всех ресурсов между всеми процессами. Соответственно, не должно быть высокоприоритетных задач, которые использовали бы какой-либо ресурс системы столько, сколько необходимо. В системах же реального времени требуется обеспечить требуемый уровень сервиса за вполне определенное, ограниченное время, т.е. ОС реального времени должна быть предсказуемой. Правильная, но запоздалая реакция системы на внешние события может быть губительной.

Операционные системы реального времени, как правило, должны удовлетворять следующим требованиям:

- своевременность реакции системы на любые внешние события за априорно известный интервал времени;
- высокая пропускная способность программных интерфейсов, гарантирующая доставку необходимых объемов данных за детерминированное время;

- развитые механизмы синхронизации процессов, обеспечивающие разделение и совместное использование наиболее критических ресурсов;
- желательно, чтобы система была многозадачной и многопроцессорной, что дает возможность дефрагментировать задачу и запускать фрагменты задач на выполнение одновременно. Последнее обеспечивает максимальное использование всех ресурсов системы;
- диспетчеризация вытесняющих задач, основанная на приоритетах, которые гарантируют, что задача с самым высоким приоритетом будет сразу же вызвана на выполнение, когда она попадет в очередь активных процессов;
- непосредственное управление планированием со стороны пользовательских процессов для обеспечения выполнения соответствующей задачи.

Принято различать системы *«жесткого»* и *«мягкого»* реального времени.

Системой *«жесткого»* реального времени называется система, где неспособность обеспечить реакцию на какие-либо события в заданное время является отказом и ведет к невозможности решения поставленной задачи. Последствия таких отказов могут быть весьма печальными, если, например, вовремя не сработала система блокировки аварийного поднятия кислородной фурмы в сталеплавильном конвертере. Фактически в системах жесткого реального времени время реакции на любые внешние события не должно зависеть от числа выполняемых прикладных процессов, и оно заранее определено.

Системы *«мягкого»* реального времени проработаны теоретически пока далеко не полностью, поэтому дать точное определение для *«мягкого»* реального времени дать не представляется возможным. Иногда считают, что в системах *«мягкого»* реального времени реакция на внешние события находится в зависимости от числа загружаемых прикладных процессов. При этом максимально возможное время реакции заранее определено. Для простоты будем считать, что сюда относятся все системы реального времени, не попадающие в категорию *«жестких»*.

Системы реального времени можно разделить в 4 класса.

1-й класс – программирование на уровне микропроцессоров. При этом программы для программируемых микропроцессоров, встраиваемых в различные устройства, очень небольшие и обычно написаны на языке низкого уровня, типа ассемблера.

2-й класс – минимальное ядро системы реального времени. На более высоком уровне находятся системы реального времени, обеспечивающие минимальную среду исполнения. Предусмотрены лишь основные функции, а управление памятью и диспетчер часто недоступны. Ядро представляет собой набор программ, выполняющих типичные, необходимые для встроенных систем низкого уровня функции, такие, как операции с плавающей запятой и минимальный сервис ввода-вывода. Прикладные программы разрабатываются в инструментальной среде, а выполняются, как правило, на встроенных системах.

3-й класс – ядро системы реального времени и инструментальная среда. Этот класс систем обладает многими чертами ОС с полным сервисом. Разработ-

ка ведется в инструментальной среде, а исполнение – на целевых системах. Этот тип систем обеспечивает гораздо более высокий уровень сервиса для разработчика прикладной программы.

4-й класс – ОС с полным сервисом. Такие ОС могут быть применимы для любых приложений реального времени. Разработка и исполнение прикладных программ ведутся в рамках одной и той же системы.

Системы 2-го и 3-го классов принято относить к системам «жесткого» реального времени, 4-го класса – «мягкого». Очевидно, это можно объяснить тем, что в первом случае к системе предъявляются более жесткие требования по времени реакции и необходимому объему памяти, чем во втором.

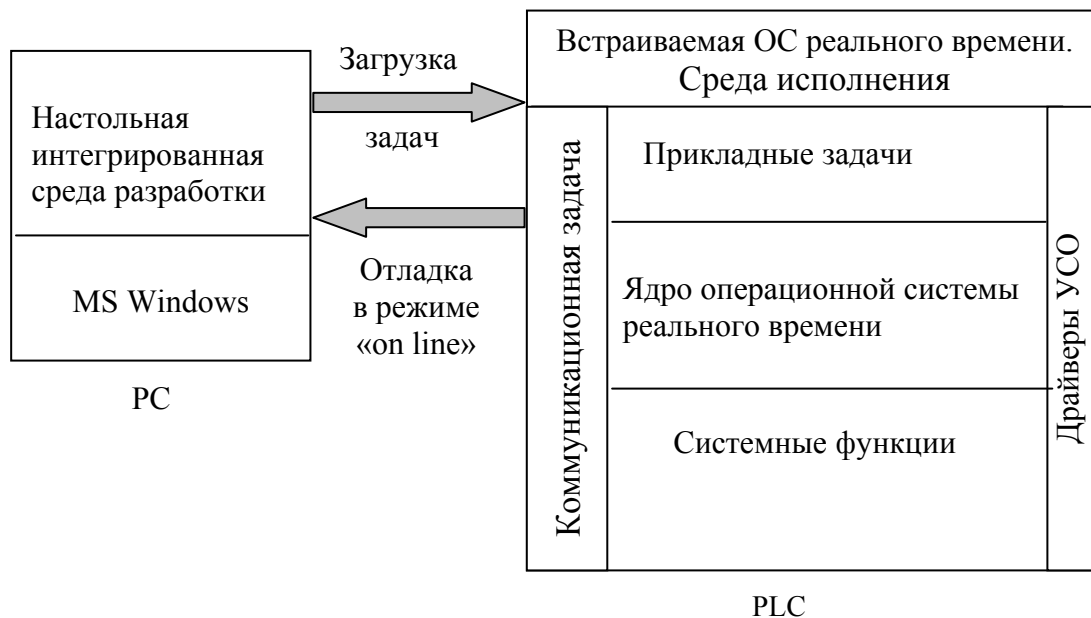
Как уже отмечалось, среда разработки и среда исполнения в системах реального времени обычно разделены. В связи с этим программное обеспечение систем контроля и управления технологическими процессами по применению разделяется на встраиваемое для системы исполнения и настольное – для системы разработки (рис. 2.9).

Система исполнения загружается в ПЗУ PLC при изготовлении контроллера или загружается пользователем. Чаще всего в качестве встраиваемого ПО используются операционные системы реального времени. Начинаются они обычно с маленького ядра и загрузчика программ, к которому как строительные блоки присоединяются системные службы и утилиты. Требования, предъявляемые к *среде исполнения* систем реального времени, следующие:

- небольшой объем памяти необходим для возможности ее встраивания;
- система должна быть полностью резидентна в памяти для того, чтобы избежать замещения страниц памяти или подкачки;
- система должна быть многозадачной – для обеспечения максимально эффективного использования всех ресурсов системы;
- микроядро ОС должно быть с приоритетом на обслуживание прерываний. Приоритет на прерывание означает, что готовый к запуску процесс, обладающий некоторым приоритетом, обязательно имеет преимущество в очереди по отношению к процессу с более низким приоритетом, быстро заменяет последний и поступает на выполнение. Ядро заканчивает любую сервисную работу, как только поступает задача с высшим приоритетом, что гарантирует предсказуемость системы;
- диспетчер с приоритетом дает возможность разработчику прикладных программ присвоить каждому загрузочному модулю приоритет. Присвоение приоритета используется для определения очередности запуска программ, готовых к исполнению. Диспетчеризация, в основу которой положен принцип присвоения приоритета, и наличие ядра с приоритетом на прерывание позволяют разработчику прикладной программы полностью контролировать систему. Если поступает событие с высшим приоритетом, то система прекращает обработку задачи с низшим приоритетом и отвечает на вновь поступивший запрос.

Основой любой среды исполнения в реальном времени является ядро или диспетчер. Ядро управляет аппаратными средствами целевого компьютера: цен-

тральным процессором, памятью и устройствами ввода/вывода, контролирует работу всех других систем и программных средств прикладного характера. В системе реального времени диспетчер занимает место между ядром операционной системы и прикладными программами, обеспечивает специальный сервис, необходимый для работы приложений реального времени.



**Рис. 2.9.** Взаимодействие сред разработки и исполнения

В ОС могут использоваться универсальные языки программирования. Они позволяют решать задачи программирования PLC, но их применение требует квалификации в теории и технологии программирования, знания тонкостей аппаратного обеспечения, поэтому использование традиционных языков программирования – не всегда лучшее решение. В связи с этим фирмы, производящие оборудование для построения информационных систем и систем автоматизации, стремятся сопровождать свою продукцию набором программных инструментов, упрощающих и ускоряющих процесс программирования, с помощью которых пользователь по определенным правилам и соглашениям мог бы описать логику работы PLC. Эти программные инструменты специально ориентированы на разработку программ управления внешними устройствами и контроллерами (см. подробнее главу 5).

В качестве технических средств среды разработки чаще всего используются PC, а в качестве операционной системы – ОС общего назначения, в частности MS Windows. Стоимость проекта любой системы реального времени определяется стоимостью программного обеспечения, поэтому все большее внимание уделяется среде разработки программ. Для разработки прикладных программ необходимы определенные средства: редакторы, компиляторы, компоновщики и отладчики. Так, символьные отладчики позволяют разработчику с удаленного терминала отлаживать исполняемую систему. Разработчики должны иметь интегрированные программные средства, охватывающие весь цикл разработки: анализ, проектирование, кодирование, отладку, тестирование, документирование

и поддержку задачи, основу которых составляют CASE-системы (Computer Aided Software Engineering).

В настоящее время существует достаточно большое число ОС, удовлетворяющих требованиям реального времени. Это широко известные ОС VxWorks, QNX, OS-9, RTKernel и многие другие.

**Операционная система OS-9** (фирма Microware Systems Corp., USA). Это многозадачная, многопользовательская, модульная, вытесняемая, Unix-подобная операционная система реального времени.

Основные свойства операционной системы:

- возможность размещения в ПЗУ как операционной системы, так и приложений пользователя;
- переносимость на основные микропроцессорные платформы Motorola, Intel, PowerPC и др.;
- детерминированное, полностью вытесняемое ядро с минимальным гарантированным временем реакции на прерывание 3 мкс;
- развитые сетевые средства (TCP/IP, Profibus, CAN, Ethernet и др.).

Основа операционной системы – модули, перемещаемые объекты в память OS-9. Все функциональные компоненты, включая микроядро, реализованы в виде отдельных модулей. OS-9 компактна и не требует больших ресурсов контроллеров, объем ядра в минимальном варианте составляет всего 24 кбайта.

Модульная структура OS-9 обеспечивает модульную технику программирования, позволяя пользователям легко наращивать и изменять конфигурацию OS-9. Наращивание не требует перекомпиляции всей операционной системы, необходима компиляция только того модуля, который обеспечивает дополнительные функции. Модульность позволяет OS-9 поддерживать общий набор прикладных программ для всего диапазона компьютеров, начиная от встроенных одноплатных компьютеров и заканчивая большими системами с разделением времени. Ядро OS-9 управляет памятью и процессорным временем, а также предлагает большой набор служебных системных функций. Многозадачное ядро OS-9 разрешает одновременное выполнение нескольких независимых программ с помощью механизма переключения задач и межпроцессорной связи. Все программы OS-9 выполняются как процессы, и каждый процесс с соответствующими правами доступа сможет осуществлять доступ к любым системным ресурсам, выполняя соответствующие системные вызовы. В дополнение к операционной системе OS-9 Microware представляет профессиональную среду программирования. Компиляторы языков высокого уровня, отладчики, средства разработки и связи обеспечивают исчерпывающую среду для разработки и поддержки прикладных программ.

**Операционная система QNX** (разработка фирмы QSSL, Канада) для IBM PC является одной из наиболее широко используемых при построении систем управления и сбора данных, прежде всего за счет того, что QNX гарантирует время реакции в пределах от нескольких десятков микросекунд до нескольких миллисекунд (в зависимости от быстродействия ПЭВМ и версии QNX). Кроме этого, высокая эффективность QNX в задачах управления и сбора данных в реальном времени обеспечивается такими свойствами, как наличие 32-разрядного ядра, многозадачность (до 250 задач на одном узле).



При проектировании ОС реального времени, как мы отмечали, все чаще используют идеологию микроядра, которая увеличивает надежность программного обеспечения и позволяет использовать только те компоненты операционной системы, которые необходимы в каждом конкретном случае.

Так, компактное ядро QNX/Neutrino имеет очень небольшой размер – менее 32 кбайт, выполняя несколько основных служб операционной системы:

- службу обмена сообщениями между потоками всей системы;
- службу синхронизации потоков;
- службу диспетчеризации потоков;
- службу управления таймерами и аппаратными прерываниями.

В то же время современные многозадачные и многопользовательские системы общего назначения, в частности Windows NT/2000 и OS/2, обеспечивают скорость отклика на внешние воздействия, сравнимую со специализированными ОС реального времени. Как показывает практика, эти ОС можно использовать только в задачах «мягкого реального времени». Заметим, что фирма VenturCom Co.(США) выпустила в 1999 году расширения реального времени для Windows NT – RTX (Real Time Extention). Они позволяют объединить все стандартные возможности Windows NT с требованиями реального времени и представляют средства для создания приложений, работающих фактически в режиме близком к жесткому реальному времени. RTX добавляет подсистему реального времени к стандартной Windows NT и обеспечивает тем самым специфическую для реального времени функциональность. Использование RTX в промышленности покажет жизнеспособность этого решения.

Заметим в заключение, что очень часто на выбор ОС влияют не только ее характеристики, но и ее стоимость, и наличие необходимого аппаратного обеспечения, и условия ее сопровождения.

## **2.5. Пакеты прикладных программ операторских станций технологических процессов (SCADA)**

### **2.5.1. Общая характеристика пакетов**

Особое место в создании информационных систем принадлежит программному обеспечению, предназначенному для эксплуатации на среднем и верхнем уровнях промышленных информационных систем (см. **рис. 1.2**), например пультов управления сложными агрегатами.

При создании АСУ ТП любой сложности всегда существовала тяжело решаемая проблема: как заставить программистов и технологов понимать друг друга. Надоело программистам разбираться с задвижками – клапанами, электродвигателями, и начали они изобретать всяческие языки «программирования без программирования», доступные для понимания простому инженеру-технологу. В результате этой деятельности появились программные пакеты для создания:

- интерфейса человек-машина (Man Machine Interface, **ММИ**), иными словами – того, что оператор почти все время видит на экране компьютера;

- программного обеспечения операторских станций технологических процессов (Supervisor Control And Data Acquisition, **SCADA**).

Граница между программным обеспечением класса SCADA и MMI весьма условна, поэтому многие системы SCADA могут работать в качестве интерфейса для систем нижнего уровня.

*Главными функциями таких SCADA-программ являются отображение технологического процесса в виде мнемосхем, сигнализация об аварийных ситуациях, обеспечение общего управления процессом со стороны оператора – технолога и т.п.*

Диспетчерское управление и сбор данных (SCADA) является основным и в настоящее время остается одним из наиболее перспективных методов автоматизированного управления сложными динамическими системами. Именно на принципах диспетчерского управления строятся крупные автоматизированные системы в металлургии.

Конечно, многие отделы АСУ крупных металлургических предприятий, как правило, имеют свои наработки в этой области. Примеры таких разработок отделов АСУ крупных металлургических заводов мы рассмотрим в дальнейшем. В то же время есть фирмы, специализирующиеся на разработке универсальных SCADA-программ, таких как Genesis, InTouch, FixDmacs, SIMATIC WinCC, iFIX, «САРГОН», RealFlex, Trace Mode, MIK\$Sys и многие другие.

SCADA – процесс сбора информации реального времени с удаленных точек (объектов) для обработки, анализа и возможного управления удаленными объектами. Требование обработки реального времени обусловлено необходимостью доставки (выдачи) всех необходимых сообщений и данных на центральный интерфейс диспетчера.

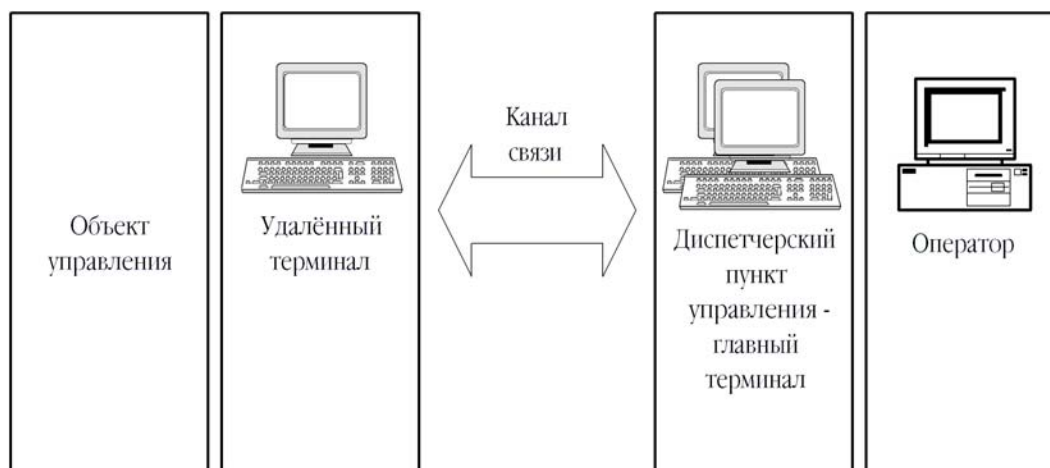
*Основные требования к SCADA-системам:*

- создание богатых возможностей для реализации графического интерфейса;
- обеспечение открытости, как с точки зрения подключения различного контроллерного оборудования, так и коммуникации с другими программами;
- обеспечение простоты разработки приложений;
- использование новых технологий.

Все современные SCADA-системы включают три основных структурных компонента (**рис. 2.10**).

*Удаленный терминал (Remote Terminal Unit, RTU) – осуществляющий обработку задачи (управление) в режиме реального времени. Спектр его воплощений широк – от простейших датчиков, осуществляющих съем информации с объекта, до специализированных многопроцессорных отказоустойчивых вычислительных комплексов, осуществляющих обработку информации и управление в режиме жесткого реального времени. Использование устройств низкоуровневой обработки информации позволяет снизить требования к пропускной способности каналов связи с центральным диспетчерским пунктом. Заметим, что число контролируемых удаленных данных в современной SCADA-системе может достигать 100 тысяч.*

*Диспетчерский пункт управления* – главный терминал (Master Terminal Unit, MTU) – осуществляет обработку данных и управление высокого уровня, как правило, в режиме мягкого реального времени. Одна из основных функций – обеспечение интерфейса между человеком-оператором и системой. В зависимости от конкретной системы диспетчерский пункт управления может быть реализован в самом разнообразном виде – от единичного компьютера с дополнительными устройствами подключения к каналам связи до больших вычислительных систем или объединенных в локальную сеть рабочих станций и серверов.



**Рис. 2.10.** Основные структурные компоненты SCADA-систем

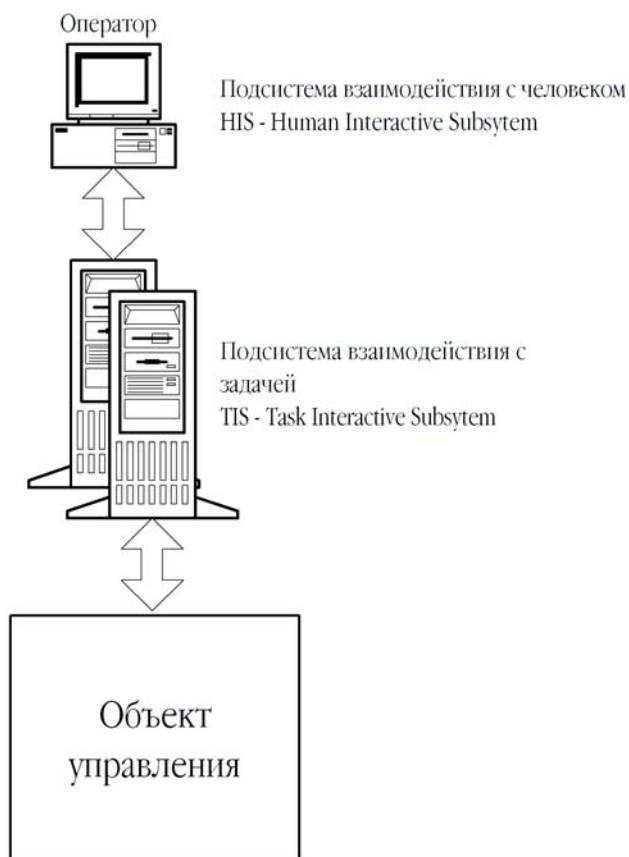
*Коммуникационная система – каналы связи (Communication System, CS)* необходима для передачи данных с удаленных точек (объектов, терминалов) на центральный интерфейс оператора-диспетчера и передачи сигналов управления на удаленный терминал (объект) в зависимости от конкретного исполнения.

Можно выделить четыре основных функциональных компонента систем диспетчерского управления и сбора данных (**рис. 2.11**):

- человек-оператор;
- компьютер взаимодействия с человеком;
- компьютер взаимодействия с задачей (объектом);
- задача (объект управления).

Поскольку конкуренция в области SCADA-систем достаточно высока и пакеты SCADA в общих чертах похожи друг на друга, мы сначала рассмотрим только общие их функции и затем дадим характеристику наиболее известных отечественных и зарубежных отечественных пакетов.

Для создания рабочего места оператора технологического процесса обычно необходимо реализовать следующий набор функций:



*Рис. 2.11. Функциональные компоненты SCADA-систем*

- органы управления различных типов;
- экранные формы отображения параметров процесса типа стрелочных, полосковых или цифровых индикаторов, а также сигнализирующие табло различной формы и содержания;
- возможность создания архивов аварий, событий и поведения переменных процесса во времени (так называемые тренды), а также полное или выборочное хранение параметров процесса через заданные промежутки времени постоянно или по условию;
- язык для реализации алгоритмов управления, математических и логических вычислений;
- средства документирования как самого алгоритма, так и технологического процесса;
- драйверы к обслуживанию нижнего уровня АСУ ТП;
- сетевые функции;
- средства защиты от несанкционированного доступа в систему;
- многооконный графический интерфейс и другие функции.

#### *Формирование статического изображения рабочего окна*

Это может быть фон, заголовки, мнемосхема технологического процесса и т.п. Для создания статического изображения, как правило, используются внешние графические редакторы, например Paint, а готовые изображения затем импортируются в пакет SCADA. Хотя некоторые пакеты имеют собственные средства рисования, все они содержат и средства импорта изображений.

### *Формирование динамических объектов рабочего окна*

Как правило, динамические объекты создаются при помощи специализированного графического редактора самого пакета по жестко заданному алгоритму или на основе набора библиотечных элементов с последующим присвоением параметров. В частности, для изображения полоскового индикатора нам нужно будет в простейшем случае изобразить прямоугольники, соответствующие начальному и конечному значению параметра, и задать эти значения. На этом же шаге динамическому объекту присваивается логическое имя, под которым он будет фигурировать в алгоритме управления. Одновременно путем ответов на вопросы меню или при заполнении соответствующего формуляра задается привязка логических имен динамических объектов к конкретному каналу ввода-вывода. В конце этого шага мы имеем набор необходимых нам динамических объектов, соответствующим образом размещенных на фоне статического изображения, и базу каналов ввода-вывода. Единственное, что остается сделать для получения работающей программы операторской станции, – описать взаимосвязи между логическими именами динамического объекта и алгоритмом функционирования системы.

### *Описание алгоритма отображения и управления*

Этот алгоритм выполняется в разных системах по-разному, хотя общие черты остаются. В простейшем случае при помощи обычного текстового редактора на языке BASIC записываются логические и математические формулы с использованием логических имен динамических объектов. В более сложных пакетах алгоритм может описываться при помощи языка функциональных блоков (ФБ). Причем исходные наборы ФБ включают в себя достаточно широкий набор: от простых фильтров и математических функций до PID-регуляторов. Как правило, в таких системах предусматривается возможность создания собственных ФБ, содержащих текст программы или формул на встроенном языке высокого уровня. На этом шаге процесс «программирования» заканчивается. Все, что остается сделать, – запустить полученную стратегию под управлением следующей неотъемлемой части всех пакетов SCADA-программы монитора.

Рассмотрим наиболее популярные программы SCADA-системы.

## **2.5.2. GENESIS (США)**

GENESIS – это один из самых популярных и наиболее продвинутых пакетов этого класса, успешно внедряемый не только за рубежом, но и в России. Первая версия этого пакета разработана фирмой Iconics (США) еще в 1986 году. С тех пор количество проданных копий пакета перевалило за 20 тысяч, а системы на его базе работают практически во всех странах мира. Последняя версия, GENESIS-32, работает под Windows 95/98/NT/2000 и позволяет осуществлять автоматизацию объектов различной сложности: от отдельного агрегата или установки до завода. Именно с использованием этого пакета в настоящее время осуществляется решение современных информационных задач при производстве алюминия, автоматизации прокатных станов. В России осуществлено к 2000 году не менее 350 инсталляций GENESIS-32 6.0. Так, развернутая в компании «Транснефть» автоматизированная система управления транспортировкой неф-

ти, базирующаяся на комплексе программ GENESIS–32, является крупнейшим в мире проектом SCADA на основе сети распределенных персональных компьютеров. Длина трубопроводов «Транснефти», перекачивающей 99,5 % нефти России, превышает 45 тыс. км. Система диспетчерского управления, в которой установлено 350 ПК с Windows NT/2000, 2000 программируемых логических контроллеров и 1,5 тыс. экранов операторов, обрабатывает данные, поступающие по 200 тыс. цифровых и аналоговых каналов. Среднее время ответа на запрос относительно состояния любого компонента инфраструктуры изменяется в пределах от 3 до 5 секунд.

GENESIS–32 является инструментальным средством, предназначенным для визуализации контролируемых технологических параметров и оперативного диспетчерского управления на верхнем уровне АСУ ТП. В состав GENESIS–32 также входят средства редактирования сценарных процедур, обеспечивающие возможность разработки части программного обеспечения средствами Windows Visual Basic.

В отличие от многих известных систем, имеющих либо собственный формат драйверов аппаратуры, либо встроенную поддержку аппаратуры ограниченного числа производителей, GENESIS–32 соответствует универсальному способу взаимодействия с аппаратными средствами любого производителя. Для фирм, занимающихся самостоятельным производством устройств сбора данных и управления, фирма Iconics предлагает комплект OPC, который позволяет создавать серверы OPC для собственных технических средств.

*OPC (OLE for Process Control) – это стандарт взаимодействия между промышленными компонентами системы сбора данных и управления (SCADA).* Через интерфейсы OPC одни приложения могут читать или записывать данные в другие приложения, обмениваться событиями, оповещать друг друга о нештатных ситуациях (тревогах), осуществлять доступ к данным, зарегистрированным в архивах. Эти приложения могут располагаться как на одном компьютере, так и быть распределенными по сети. Реализация OPC основана на объектной модели COM/DCOM фирмы Microsoft. COM (Component Object Model) – модель многокомпонентных объектов, позволяющая вызывать те или иные функции (методы) этих объектов так, как будто объекты находятся «рядом». Объект может находиться и в самом деле рядом в адресном пространстве приложения – тогда это просто COM. Если же объект находится в другой программе на том же компьютере или на другом узле сети, то это DCOM – Distributed (распределенная) COM. По сути DCOM – универсальная распределенная модель компонентного объекта.

Особый класс OPC-приложений представляют собой *OPC-серверы конкретных аппаратных устройств*. Они создают своего рода абстракцию аппаратуры, позволяя любому OPC-клиенту записывать и считывать данные с устройства. Устройство, для которого есть OPC-сервер, может использоваться вместе с любой современной SCADA-системой.

OPC-взаимодействие основано на схеме клиент-сервер. OPC-клиент, например SCADA, вызывая определенные функции объекта OPC-сервера, подписывается на получение определенных данных с определенной частотой. В свою очередь, OPC-сервер, опросив физическое устройство, вызывает известные функции клиента, уведомляя его о получении данных и вручая сами данные. Та-

ким образом, при OPC-взаимодействии используются как прямые (от клиента к серверу), так и обратные (от сервера к клиенту) вызовы. В настоящее время насчитывается более 1000 серверов OPC для аппаратуры ведущих мировых производителей.

*Фактически OPC сегодня признан в качестве стандартного механизма взаимодействия программного обеспечения АСУ ТП.*

GENESIS-32 включает в себя следующие приложения:

- *GraphWorX32*—система визуализации и диспетчерского управления технологических параметров.
- *TrendWorX 32*—система построения графиков и архивирования данных.
- *AlarmWorX32*—система обнаружения аварийных событий и оповещения персонала.

**GraphWorX32** реализует интерфейс человек-машина (ММІ). Основные характеристики GraphWorX32:

- многопоточное 32-разрядное приложение;
- мощные инструменты для создания экранных форм и динамических элементов отображения;
- встроенная среда редактирования сценарных процедур Visual Basic for Applications 6.0;
- средства разработки шаблонов экранных форм, содержащих наиболее часто используемые слои графических объектов;
- обширная библиотека элементов отображения, ориентированных на построение мнемосхемы промышленных объектов;
- возможность встраивания графиков TrendWorX32 и журналов событий и тревог AlarmWorX32.

GraphWorX32 имеет в своем составе полный набор средств рисования и анимации, объединенных в объектно-ориентированную среду разработки технологической графики. Проект состоит из совокупности экранных форм, каждая из которых хранится в отдельном файле в каталоге проекта. Разработка экранных форм выполняется с помощью инструментов рисования, встраивания графических изображений из метафайлов и файлов растровых изображений, а также путем использования обширной библиотеки встроенных стандартных символов. Каждый элемент отображения может быть связан с переменной процесса и отображать ее значение или состояние.

GraphWorX32 является полнофункциональным контейнером ActiveX и объектов OLE и позволяет вставлять в рабочую область окна экранной формы элементы управления ActiveX, разработанные фирмой Iconics, а также любых других производителей.

*ActiveX-технология* позволяет создавать экранные формы, состоящие из ранее разработанных компонентов, как из кубиков. Сегодня можно говорить о том, что современный процесс генерации конечного приложения (программы) все более напоминает не традиционный процесс разработки прикладного программного обеспечения, а процесс компонентной сборки. При этом соответственно изменяется и характер труда прикладного программиста. В настоящее время формируется открытый рынок ActiveX-компонентов и появляются фирмы,

специализирующиеся на выпуске подобной продукции. Это позволяет системному программисту использовать в рамках современной SCADA-системы дополнительные ActiveX-компоненты, разработанные самостоятельно или приобретенные у поставщиков. В GraphWorX32, кроме того, имеется возможность встраивания документов текстового процессора MS Word, рабочих листов и диаграмм MS Excel, а также видеороликов и других элементов управления OLE. Данная функциональная возможность GraphWorX32 является одним из наиболее уникальных свойств пакета, обеспечивающих универсальный механизм использования разработок в области технологической и деловой графики разных производителей в проектах, создаваемых средствами GENESIS-32.

Экранные формы, создаваемые с помощью GraphWorX32, могут быть насыщены графическими объектами и элементами отображения, форма и положение которых зависят от значений и состояний контролируемых параметров в прикладной области, а также от действий персонала, эксплуатирующего систему.

Любому элементу отображения и графическому объекту может быть поставлен в соответствие один или несколько способов динамизации (изменение размера по вертикальной и горизонтальной осям, изменение цвета, перемещение, вращение и др.) в зависимости от связанного с ним параметра.

GraphWorX32 содержит библиотеку символов технологической и деловой графики, которая позволяет значительно улучшить внешний вид экранных форм и создавать интуитивно понятные мнемосхемы технологических процессов. Одним из наиболее привлекательных качеств библиотеки символов является возможность ее расширения и дополнения пользователем путем создания собственных символов.

**TrendWorX32** является многооконным приложением, которое предназначено для выполнения следующих основных функций:

- представление контролируемых параметров в виде графики (трендов) различных типов в реальном масштабе времени;
- архивирование значений контролируемых параметров;
- вычисление статистических характеристик выборок значений контролируемых параметров;
- извлечение значений контролируемых параметров из архивов и представление в виде графиков различных типов;
- передача информации в базы данных (MS Access, MS SQL Server, Oracle и др.);
- обеспечение механизма извлечения информации из баз данных и передачи подсистемам построения графиков и визуализации;
- вывод графиков на печатающее устройство.

TrendWorX32 является клиентом OPC, поддерживающим технологии ActiveX и OLE, и может использоваться как совместно с другими компонентами GENESIS-32, так и с приложениями других производителей.

TrendWorX32 является многооконным контейнером ActiveX, который может запускаться как автономно, так и одновременно с другими компонентами GENESIS-32. Каждое окно TrendWorX32 содержит элементы управления, с помощью которых выполняется графическое представление неограниченного ко-



личества контролируемых параметров и внутренних переменных проекта. Поскольку TrendWorX32 является клиентом OPC, имеется возможность построения графиков значений параметров, сбор которых ведется на узлах глобальной сети Internet и локальной вычислительной сети предприятия.

*AlarmWorX32* является мультимедийным приложением, которое предназначено для выполнения следующих основных функций:

- голосовое оповещение персонала об обнаруженных аварийных ситуациях;
- рассылка электронных извещений об аварийных событиях посредством пейджинговой связи и электронной почты;
- оповещение персонала путем автоматического дозвона по коммутируемым каналам связи с передачей сообщений об аварийных событиях и приемом подтверждений восприятия от ответственных лиц;
- персональное планирование оповещения для задействования только дежурного персонала;
- анализ аварийных событий и действий ответственного персонала;
- объединение всех аварийных событий и подтверждений восприятия системных сообщений ответственным персоналом в сводки аварийных событий;
- отображение вспомогательной информации для аварийных событий, позволяющей локализовать и устранять причины аварии.

*AlarmWorX32* является клиентом OPC, поддерживающим технологии ActiveX и OLE, и может использоваться как совместно с другими компонентами GENESIS–32, так и с приложениями других производителей.

Для каждого технологического параметра, контролируемого системой, могут быть заданы условия, наступление которых воспринимается системой как аварийная ситуация. Технологические параметры, проверяемые на выполнение условий аварийных ситуаций, объединяются в группы, каждой из которых может назначаться свой приоритет, а также устанавливаться список лиц, ответственных за принятие мер по устранению причины и ликвидации последствий аварии.

*Системное администрирование и управление правами доступа.* Пакет GENESIS имеет мощные встроенные средства системного администрирования и управления правами доступа к информации, связанной с системой. Все лица, имеющие отношение к системе управления технологическим процессом, с помощью сервера системного администрирования вносятся в список допускаемых к системной информации. При этом имеется возможность создания групп ответственных лиц по занимаемым должностям и уровню ответственности. Наивысшим правом доступа обладает системный администратор. Остальной персонал, допущенный к системным функциям и системной информации, получает ограниченные права.

### 2.5.3. SIMATIC WinCC – модульная SCADA-система для мониторинга технологических процессов

В настоящее время этот программный продукт занимает первое место в Европе среди SCADA-систем. Он базируется на распространенных ОС Windows 95/98/NT.

В основной комплект WinCC входят следующие модули:

- *Alarm Logging (служба сообщений)* – предназначен для ввода сообщений о ходе контролируемого технологического процесса во время работы WinCC-приложения, подтверждения сообщений оператором и ведения архивов этих сообщений. Система состоит из двух частей – системы конфигурирования и системы исполнения. Система конфигурирования служит для создания таких параметров сообщений, как условия их возникновения, а также для задания параметров архива сообщений. Система исполнения предназначена для вывода сообщений на экран и их архивирования во время работы (служба сообщений), а также для ввода сообщений о ходе контролируемого технологического процесса во время работы WinCC-приложения.
- *User Administrator Text Library (администратор пользователей)* – предназначен для контроля прав доступа пользователей WinCC-приложения. Каждому пользователю в режиме конфигурирования WinCC-приложения или во время его выполнения могут быть присвоены определенные «уровни доступа». Как и все другие WinCC-редакторы, User Administrator состоит из двух частей – системы конфигурирования и системы исполнения. Система конфигурирования отвечает за создание новых пользователей, изменение паролей и определение уровней доступа. Назначение системы исполнения – контроль прав доступа, как во время выполнения приложения, так и во время его редактирования.
- *Text Library (поддержка различных языков)* – предназначен для создания библиотеки соответствий слов или символов. Это необходимо, когда текст может отображаться на двух или более языках.
- *Report Designer (встроенный генератор отчетов)* – предназначен для проектирования видов отчетов, которые могут содержать информацию в виде таблиц или графиков.
- *Global Scripts (подсистема обработки событий)* – позволяет делать проект динамичным в зависимости от требований.
- *Tag Logging (подсистема архивирования параметров)* – предназначена для создания оперативных и долговременных архивов. Совместно с редактором представляются средства для табличного и графического отображения значений в базе данных.
- *Graphics Designer (графический редактор)* – предназначен для создания мнемосхем и содержит богатую библиотеку стандартных графических объектов. При этом для каждого графического объекта определен набор свойств, которые могут изменяться во время исполнения приложения. Graphics Designer поддерживает технологию OLE, что

позволяет встраивать в мнемосхемы компоненты ActiveX/OLE, разработанные пользователем. В итоге пакет позволяет создавать графические объекты, обладающие самыми разнообразными функциями, а также организовывать связь между WinCC и другими Windows-приложениями.

Для облегчения задач выполнения WinCC постоянно пополняется новыми модулями. Так, программа Life beat Monitoring позволяет следить за состоянием соединением всех PLC, подключенных к системе, а также выдавать информацию об ОС компьютера, к которому подключен PLC, и отображать информацию с выбранного соединения.

Пакет WinCC широко используется в отечественной металлургии. Так, в настоящее время на базе WinCC реализованы и успешно функционируют современные информационные системы управления производством окатышей на Лебединском горно-обогатительном комбинате, выплавкой чугуна в доменных печах ОАО Западно-Сибирский металлургический комбинат (ЗСМК), стали в конвертерах на ОАО ММК и ОАО ЗСМК, на ряде предприятий цветной металлургии и др. На **рис. 2.12** представлен пример изображения мнемосхемы обжига окатышей на конвейерной машине Лебединского ГОКа. Экран монитора разделен на обзорное (вверху), рабочее (в середине) и клавиатурное поле (внизу). В поле обзора расположена строка сообщений, отображающая важнейшие сообщения, такие, например, как аварийные сигналы оповещения. Поле обзора служит также для быстрой ориентации в текущем состоянии управляемого процесса. Клавиатурное поле содержит функциональные клавиши, которые используются для перемещения по иерархии мнемосхем и активизации других функций (например, вызов окна «Расчет по математической модели процесса»). Рабочее поле служит для представления информации технологического процесса в соответствующих графических формах.

#### **2.5.4. BASEstar – комплект программного обеспечения для АСУ ТП в реальном времени фирмы DIGITAL (США)**

Фирма DIGITAL предлагает концепцию комплексной автоматизации, но, в отличие от описанной ранее (см. **рис.1.2**) пирамиды, предлагается не 5, а 4 уровня управления, где 3-й уровень по существу является совокупностью 3-го и 4-го уровней предыдущей модели (**рис. 2.13**). Именно с этой целью и был создан пакет BASEstar. Основная его функция – реализация уровня MES для интеграции АСУП и АСУТП. Кроме того, BASEstar может успешно выполнять и функции SCADA, предоставляя удобные средства для разработчика.

Комплект программного обеспечения BASEstar состоит из средств управления и сбора информации с УСО, средств разработки новых пакетов прикладных программ и средств интеграции уже существующих программ. BASEstar является по своей сути каркасом для автоматизации промышленных объектов, обеспечивает интеграцию АСУ ТП и АСУП с целью создания общей гетерогенной управляющей среды, а именно:

- получения достоверной информации в любой точке управления в реальном масштабе времени;

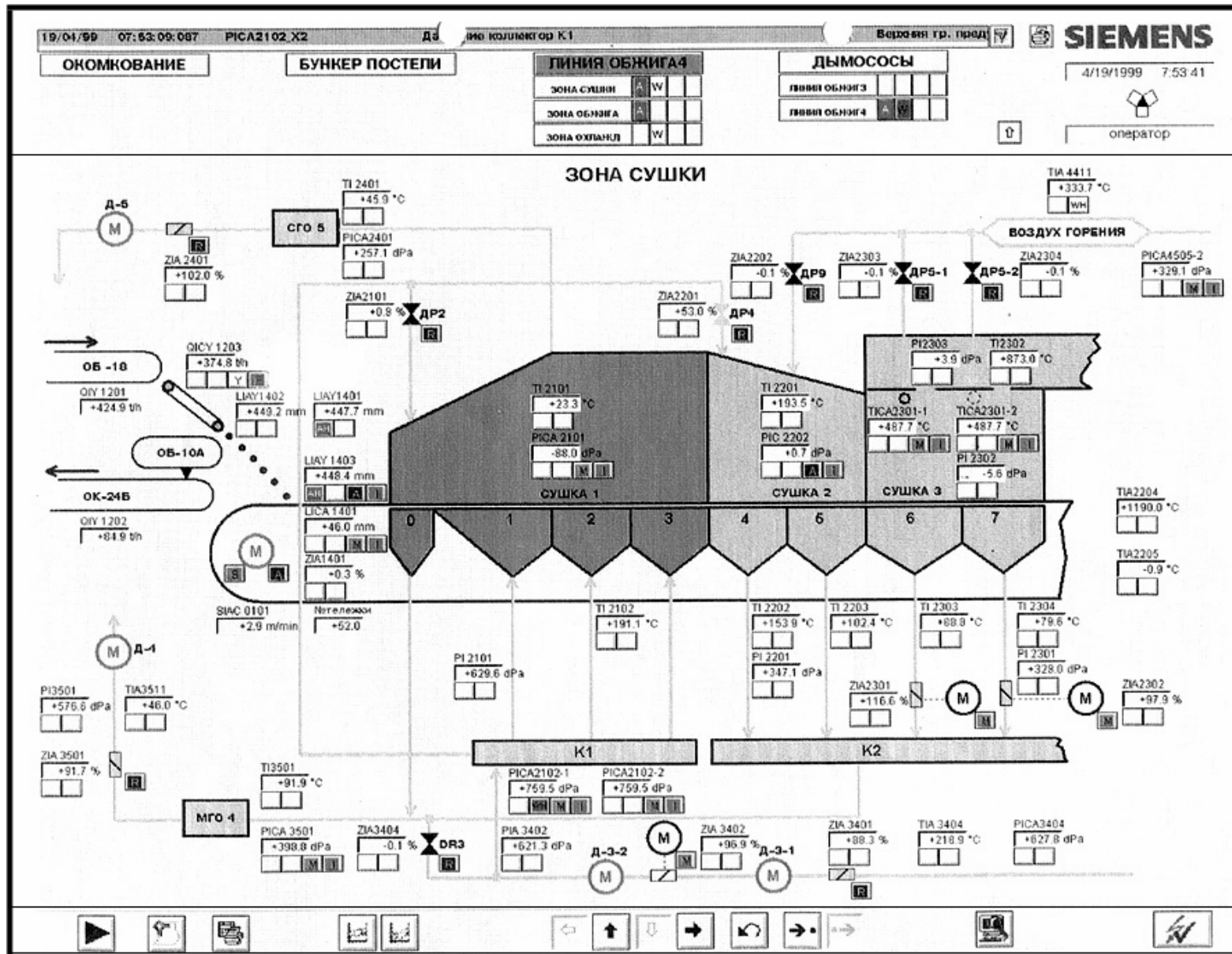


Рис. 2.12. Пример изображения мнемосхемы на экране монитора

- возможности подключения программных продуктов и оборудования различных производителей.

Основу этого комплекта составляет *пакет BASEstar Open (Server и Client)*, который является средством управления технологическими процессами и разработки приложений АСУ ТП, а также доступ к информации АСУ ТП с рабочих мест. Он поддерживает открытую сетевую архитектуру распределенного типа и может устанавливаться на операционные системы различного типа: Unix, Windows NT и др. Аппаратные средства могут быть реализованы на Intel-платформах различных производителей, а также на базе DEC (Alpha, VAX), HP (HP-UX).

Основные функции пакета BASEstar Open (Server и Client):

- сбор и обработка данных о технологических процессах и их передача различным узлам сети;
- автоматическая обработка аварийных сигналов и оповещение о них обслуживающего персонала;
- прямой обмен пакетами информации между различными узлами сети;
- моделирование производственного процесса и создание наиболее оптимальных конфигураций технологических цепочек, включая оперативное изменение процедур технологических процессов и схем управления при переходе на другой вид продукции.



Рис. 2.13. АСУ ТП BASEstar и ее место в автоматизации предприятия

**Пакет *BASEstar Graphics Enabler (BGE)*** – обеспечивает широкие возможности разработки мнемосхем и различных моделей технологических процессов, что позволяет производить полномасштабное управление технологическим процессом в режиме реального времени (SCADA-система). BGE обеспечивает легкую возможность создания графического (и/или табличного) отображения технологического процесса и его редактирования, используя обширную библиотеку элементов, объектов, исполнительных механизмов, а также фрагментов управления из различных отраслей промышленного производства. Наличие в библиотеке BGE большого количества компонентов управления (кнопок управления, измерительных приборов и т.п.) и уже готовых фрагментов позволяет пользователю создать структуру управления технологическим процессом путем компоновки из имеющихся объектов. При этом основным механизмом построения является процедура «drag and drop» (извлек из библиотеки и поместил в нужное место). Затем, используя простой язык команд, указывается, какие графические атрибуты объекта будут динамически управляться с помощью данных и событий в реальном времени, поступающих из приложений, работающих в системе *BASEstar*.

**Пакет *BASEstar Device Connect (DEComni)*** обеспечивает «прозрачный» доступ к любому объекту распределенной сети УСО, поддерживающих стандарт ISO/IEC 9506 (MMS – Manufacturing Message Specification), а **пакет *Device Access Software (DAS)*** – к всевозможным контролерам Allen Bradley, Siemens, Modicon, Telemecanique, April и многих других.

Кроме того, на базе *BASEstar* фирмой Digital разработан пакет *PDAS* (Process Data Acquisition System), предназначенный для интеграции АСУ ТП *BASEstar* с широко распространенной в мире автоматизированной системой управления производством SAP R/3 фирмы SAP-AG.

Как видно из этого перечня, пакет *BASEstar* содержит программное обеспечение для построения SCADA. Однако, помимо этого, в *BASEstar* имеются средства сопряжения с системами SCADA известных производителей, в частности InTouch, Genesis и др.

Комплект программного обеспечения *BASEstar* успешно используется на Оскольском электрометаллургическом комбинате при создании АСУ ТП машин непрерывного литья заготовок.

В заключение отметим, что определенным сдерживающим фактором к применению западных SCADA-систем являются большой объем работ по освоению, апробации, адаптации, опасения остаться с дорогостоящим пакетом один на один ввиду невозможности получения оперативной технологической поддержки на этапе освоения и внедрения, а также высокая стоимость, не гарантирующая немедленного получения положительного результата.

## **2.6. Пакеты прикладных программ управления производством**

### **2.6.1. Общая характеристика пакетов**

Чем крупнее современное металлургическое предприятие, тем дороже обходится неэффективное управление. Практика показывает, что в большинстве

предприятий при устаревшей информационной технологии или же при полном отсутствии таковой на любого руководителя буквально обрушивается поток информации, неструктурированной, несвязанной и зачастую неактуальной.

Перед руководством различных служб современного металлургического предприятия постоянно встает целый ряд неотложных вопросов, требующих прямого участия или координации. Времени на принятие решений становится все меньше, а цена ошибки возрастает. Выход – в современных подходах к управлению предприятием, основанных на использовании надежного и перспективного программного обеспечения – корпоративных информационных систем. В настоящее время на крупных металлургических предприятиях создано большое количество АРМов, автоматизирующих работу разных служб и отделов, внедряются современные информационные системы отдельных технологических объектов. Эти АРМы и информационные системы технологических объектов связаны между собой разного рода «перекачками» данных, однако они не составляют единую технологическую систему, реализующую современные принципы управления предприятием в условиях рыночных отношений в целом. Только при увязке решаемых задач с реальными потребностями и ориентацией на конечный результат возможно создание информационного пространства для управления предприятием.

Ядром каждой производственной системы являются воплощенные в ней рекомендации по управлению производством. Они представляют собой описание наиболее общих правил, по которым должно производиться планирование и контроль различных стадий производственного процесса: потребностей в сырье, закупок, загрузки мощностей, распределения ресурсов и пр. Исходным стандартом, появившимся в 70-х годах, был стандарт MRP (Material Requirements Planning), включивший только планирование материалов для производства. Этот стандарт был расширен до MRP-II (Manufacturing Resource Planning). MRP-II позволяет планировать все производственные ресурсы предприятия (сырье, материалы, оборудование и т.д.) Дальнейшим развитием стал стандарт ERP (Enterprise Resource Planning – Управление ресурсами предприятий), который позволил объединить все ресурсы предприятия; таким образом, добавилось управление заказами, финансами и т.д.

ERP-системы – это базовые технологические достижения экономики. В XXI веке они станут такими же обязательными элементами внутренней инфраструктуры предприятий, как компьютерные сети, телефонная связь или электропитание.

Современные ERP-системы, как правило, обладают следующими свойствами:

- интегрированность;
- гибкость и настраиваемость на особенности работы конкретного предприятия;
- наличие технологий внедрения.

*Интегрированность.* Все современные ERP-системы состоят из нескольких самостоятельных подсистем, которые охватывают все основные сферы деятельности предприятий: производство, сбыт, снабжение, финансы, склад, информационная система руководителя и другие. В целом различие между не-

сколькими крупнейшими системами по набору функций и алгоритмической реализации не слишком велика.

*Гибкость и настраиваемость на особенности работы конкретного предприятия.* Речь идет не только об инструментальных средствах, встроенных средствах программирования ERP-системы. Они имеются во всех ERP-системах, однако роль этих средств вспомогательная. Главное в этом свойстве – это настройка бизнес-процессов на сложившуюся на предприятии организационную структуру и должностные инструкции. Средства настройки нужны не только для того, чтобы привязать ERP-систему к условиям заказчика. Приходится считать, что мы живем в крайне динамичном мире. Представления руководства предприятия об оптимальной структуре и существовании бизнес-процессов изменяется. Если автоматизированная система управления не имеет средств для быстрой перестройки модели предприятия, то внедрение системы рискует никогда не закончиться. Средства настройки, входящие в состав ERP-систем, позволяют решить эту проблему.

*Наличие технологий внедрения.* Чтобы внедрение закончилось успехом и уложилось в отведенные для него сроки, требуется грамотное планирование и скоординированная работа всех участников процесса. Поставщики ERP-систем, которые специализируются на внедрении, должны иметь спланированный процесс и выдержать процедуру внедрения в соответствии с разработанными для этого стандартами. Компании рассматривают проект по внедрению как определенную последовательность этапов. Для каждого этапа прописано его содержание, набор документов, которые должны быть разработаны, решения, которые должны быть приняты.

В последнее время начинает получать распространение интегрированное программное обеспечение, основной задачей которого является оптимизация управления производством в целом, включая планирование запасов комплектующих, расписание технологических процессов и т.д. Системы, с одной стороны, взаимодействуют с системами SCADA, образуя единую систему управления производством, с другой стороны, они часто интегрируются в систему планирования и управления предприятием в целом и обеспечивают функционирование всех уровней автоматизации предприятия: Control, SCADA, MES, MRP, (ERP) (см. **рис.1.2**). Решение этой задачи на металлургическом предприятии можно осуществить либо разработкой собственного интегрированного пакета, либо закупкой уже готового интегрированного пакета. На мировом рынке управленческих информационных систем имеется относительно большой выбор таких пакетов: система SAP R/3, Oracle Applications, TRITON, Factory Suite, MTMS, M2, Галактика и другие. По состоянию на 2000 год две системы – SAP R/3 и TRITON (BAAN) – являются мировыми лидерами в данной области. За ERP-системой должны стоять тысячи человеко-лет и опыт тысяч внедрений на передовых предприятиях, независимость от отрасли внедрения, от компьютерных платформ и масса других ценных для заказчика свойств.

Рассмотрим лишь краткую характеристику некоторых из существующих ERP-систем.



### 2.6.2. MANUFACTURING TOTAL MANAGEMENT ASYSTEMS (MTMS)

MTMS – комплексная система управления всеми аспектами производственной и финансовой деятельности предприятий, которая может использоваться в самых различных отраслях промышленности.

MTMS включает свыше 2000 стандартных прикладных программ, каждая из которых рассчитана на работу в качестве самостоятельной программы или совместно с другими стандартными системами MTMS.

MTMS включает две основные части: *общесистемное программное обеспечение* и *прикладное программное обеспечение*, охватывающее все прикладные области управления производственного предприятия; обе части, общесистемная и прикладная, образуют единую интегрированную информационную систему.

*Общесистемное программное обеспечение включает:*

- общесистемную часть;
- интерфейс с другими прикладными системами;
- средства для адаптации системы к специфике пользователя;
- техническое программное обеспечение;
- базы данных.

Предусмотрена стандартная и расширенная конфигурации прикладного программного обеспечения.

*Стандартное прикладное программное обеспечение включает:*

- базовую часть (административный модуль);
- производственную часть (производственные данные, управление производством);
- управление запасами;
- финансовую деятельность и бухгалтерский учет;
- коммерческую часть;
- планирование производства;
- планирование ресурсов.

В расширенной конфигурации прикладного программного обеспечения в дополнение к стандартной конфигурации имеются блоки:

- контроля качества;
- контроля распределения;
- управления инженерными разработками;
- контроля присутствия персонала и т.д.

MTMS – это открытая система, работающая на различных платформах (ALPHA, IBM RS6000, IBM AS/400, SUN и др.), с различными операционными системами (Unix, SOLARIS и др.).

### 2.6.3. Программный комплекс Oracle Applications

Приложения Oracle Applications ориентированы в первую очередь на крупные промышленные предприятия со сложной организационной структурой,

территориально удаленными подразделениями и большой номенклатурой производства

Пакет бизнес-приложений Oracle Applications – это 55 интегрированных программных модулей, каждый из которых представляет полнофункциональное решение в областях:

- управления персоналом (приложения – заработная плата, персонал и др.);
- управления финансами (приложения – главная книга, основные средства, кредиторы и т.д.);
- управления материально-техническим снабжением и материальными потоками (приложения – планирование производства, материальных потоков, управления материальными запасами и т.д.);
- управления маркетингом (сбыт и маркетинг, банк данных Oracle Applications и т.д.);

Модульный подход Oracle Applications позволяет заказчику начать с минимального набора модулей и постепенно расширять его, дополняя базовую функциональность системы и избавляя от необходимости приобретения излишних в данный момент функциональных блоков. Корпорация Oracle является по сути дела единственным поставщиком, предлагающим не только прикладные решения управления предприятием, но и всю совокупность базовых программных продуктов и технологических решений, начиная с систем управления базами данных и средств их разработки и заканчивая специализированными продуктами для руководителей высшего звена – системами поддержки принятия управленческих решений (DSS, Decision Support Systems).

Приложения Oracle Applications – первая в мире полнофункциональная ERP система, целиком основанная на Internet-технологиях. Как мы увидим в дальнейшем, по своей природе Internet-архитектура представляет собой наиболее гибкую и перспективную среду для создания информационных систем масштаба крупного предприятия.

Пакет Oracle Applications соответствует принципам открытых информационных систем, что создает предпосылки того, что заказчик не окажется заложником технологической политики только этого поставщика, когда любое расширение базовых возможностей системы чревато финансовыми затратами и непредсказуемыми результатами.

Пакет Oracle Applications работает на платформах Sun Solaris, Intel Solaris, Intel NT, IBM AIX, Dec Unix и др.

#### **2.6.4. Программный комплекс управления предприятием IFS Applications**

Комплекс IFS Applications (разработчик – шведская компания IFS, Industrial & Financial Systems) охватывает основные сферы деятельности предприятия вне зависимости от его профиля. IFS Applications адаптирован для российских условий. Комплекс представляет из себя набор модулей, при этом функционально некоторая совокупность модулей рассматривается как система.

Состав модулей каждой системы дает первичное представление о ее функциях. Перечислим назначение основных модулей этой системы.

**Модуль IFS Финансы** – управление финансовыми потоками и бухгалтерский учет. Основные пользователи – работники бухгалтерий и плановых подразделений.

**Модуль IFS Поставки** – управление закупками, продажами, движением товарных потоков. Основные пользователи – работники подразделений снабжения и сбыта, складов, плановики

**Модуль IFS Производство** – управление производственными запасами. Координация работ с системой IFS Поставки. Основные пользователи – управленческий, инженерный, технический персонал производственных участков.

**Модуль IFS Технолог** – регистрация, учет и контроль конструкторско-технологической спецификации на продукцию. Тесная координация с системой IFS Производство. Основные пользователи – персонал технологических служб.

**Модуль IFS ППР** – планово-предупредительные и текущие ремонты. Учет ресурсов, материалов, времени при планировании и проведении ППР. Тесная связь практически со всеми IFS-системами.

**Модуль IFS Маркетинг** – мощный инструмент совершенствования деятельности маркетинговых служб. Основные пользователи – персонал маркетинговых служб.

**Модуль IFS Персонал** – решает разнообразные функции планирования и учета деятельности персонала фирм. Максимальная автоматизация табельного учета. Связь с системой IFS Финансы по учету затрат на проект.

### **2.6.5. Интегрированное семейство прикладных программ TRITON (ТРАЙТОН)**

Интегрированное семейство прикладных программных и программно-инструментальных средств – TRITON (ТРАЙТОН) фирмы BAAN (США, Нидерланды) – обеспечивает комплексное решение задач производственных, плановых, финансовых, снабженческих, сбытовых подразделений промышленных предприятий, холдингов, транснациональных корпораций.

Технически TRITON/BAAN реализована в соответствии со стандартами открытых систем, функционирует в среде Unix, Windows NT/2000, может быть портирована на любую распространенную аппаратную платформу (в частности – IBM, SUN и др.) и СУБД (в частности – ORACLE, Informix, Sybase и др.). Структура семейства TRITON/BAAN представлена на **рис. 2.14**, включает ряд подсистем по планированию ресурсов производства – Производство, Сбыт и материально-техническое снабжение, Сервис, Финансы и т.п.

Заметим, что подсистема «Предприятие» берет основные данные из всех пакетов TRITON и использует эти данные для оценки деловой активности предприятия или компании. При этом руководители и лица, принимающие решения, имеют легкий и быстрый доступ к этой информации. И, что наиболее важно, информация поступает в доступной форме, что делает возможным быстро и точно принимать оперативные решения.

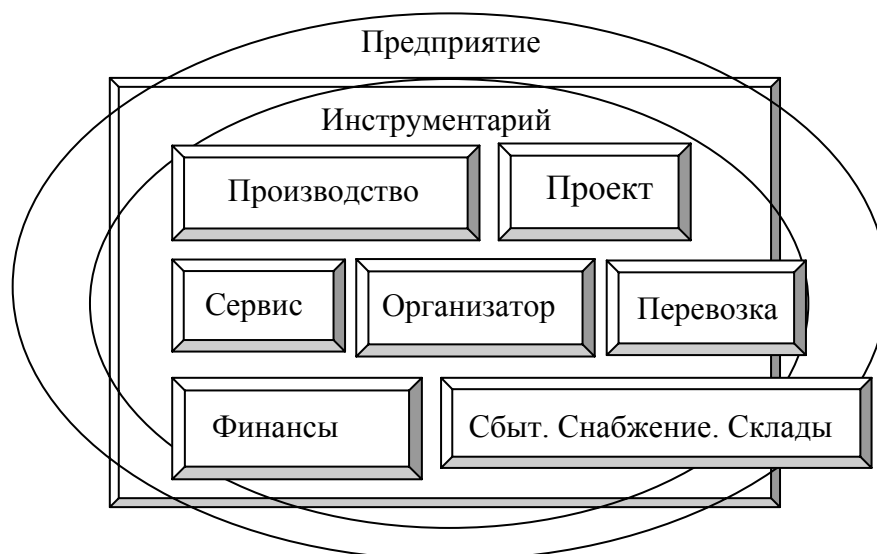


Рис. 2.14. Структура комплекса TRITON/BAAN

### 2.6.6. Интегрированный программный пакет Factory Suite 2000

Интегрированный программный пакет Factory Suite 2000 (заводской набор Wonderware) фирмы Wonderware (США) олицетворяет смелую идею интегрированной системы промышленной информационной системы. Это первый набор программ, создающий постоянный источник поступающих в реальном времени данных, которыми могут пользоваться все работники предприятия: операторы, инженеры и руководители среднего и высшего звена. Компоненты Factory Suite 2000 «покрывают» три уровня автоматизации предприятия: Control, SCADA, MES (ERP). Большое значение имеет то, что набор программ Factory Suite 2000 тесно связан с системой Microsoft. Это означает, что его программы интегрируются с серверами Microsoft SQL Server, Windows NT Server. Factory Suite 2000 работает под Windows NT/2000, а программа операторского интерфейса и клиентские модули работают также и под Windows 95/98.

**In Control** – система программирования контроллеров. Поддерживает широкий набор устройств ввода-вывода информации различных датчиков через устоявшиеся интерфейсы и открытые промышленные локальные сети. Обеспечивает режим реального времени (время реакции до 1 мс).

**InTouch** – наиболее известный пока компонент Factory Suite – является одной из самых популярных в мире программ MMI/SCADA (только в 1995 году установлено около 20 тыс. пакетов). Это набор мощных и гибких средств разработки операторских интерфейсов для создания АСУ ТП дискретных и непрерывных производств, распределенных систем управления – полнофункциональная SCADA-система. InTouch позволяет следить за работой предприятия, наблюдая за процессами, графически отображенными на экранах в реальном масштабе времени, допускает неограниченное количество динамических элементов отображения в каждом окне. Можно создавать собственные сценарные функции на языке С и т.п. InTouch является источником информации о ходе технологиче-

ского процесса для двух MES-систем, входящих в Factory Suite – Scout и InTrack. Следует, однако, отметить, что весьма популярный за рубежом пакет InTouch имеет недостаточную техническую поддержку дистрибьюторами в России.

**Industrial SQL Server** – мощная система управления базами данных реального времени. Собирает и хранит историю производственного процесса, позволяет работать при этом с сотнями устройств ввода-вывода и управления, а также с узлами InTouch и InControl. Обеспечивает высокое быстродействие – запись 40000 параметров процесса в секунду на сервере Dual Pentium Pro 200. Частота записи параметров – до 1 мс.

**InBatch** – мощная гибкая система управления процессами дозирования и смешения. Она специально разработана для моделирования и автоматизации дозирования и смешения материалов. С помощью InBatch пользователи в металлургической, химической промышленности, промышленности строительных материалов могут моделировать свои процессы, осуществлять их имитацию, сопоставлять с реальными данными и управлять реальными процессами.

**InTrack** – автоматизированная система управления производством. Объектно-ориентированная система InTrack позволяет моделировать процессы, контролировать материально-технические запасы, автоматизировать сбор данных, определять очередность работы для составления графиков и т.д. Содержит графическую среду моделирования, встроенный язык, интерфейсы с СУБД.

При выборе того или иного интегрированного пакета управления производством обычно используют группы различных критериев и ограничений, в частности:

- показатели эффективности операционных систем и СУБД (возможность поддержки распределенных рабочих мест, возможность неограниченного роста числа пользователей и т.п.);
- характеристики пакета в целом (открытость архитектуры, возможность поэтапного внедрения, удобство интерфейса пользователя, наличие русской версии, стоимость внедрения и т.п.);
- требования к аппаратной платформе (возможность поддержки типовых аппаратных средств, независимость от сетевых решений);
- авторитет фирмы – производителя (наличие опыта и результаты внедрения в России, наличие центров поддержки и т.п.).

Заметим в заключение, что реализация этих пакетов зачастую требует коренного преобразования существующей управленческой технологии металлургического предприятия, реструктуризации и реорганизации служб, замены и подготовки персонала.

## 2.7. Контрольные вопросы

1. Что такое компьютер? Какие основные компоненты он в себя включает?
2. Перечислите основные функции процессора. С какой целью в компьютер устанавливаются сопроцессоры?
3. В чем заключаются особенности использования внешней и внутренней памяти?
4. На какие классы подразделяют программное обеспечение компьютера?

5. Что такое системное программное обеспечение, какие функции оно выполняет и на какие классы условно подразделяется?
6. Дайте определение операционной системе. Чем вызвана необходимость использования операционных систем?
7. Раскройте смысл выражений: «однопользовательский и многопользовательский режим», «вытесняющая и невытесняющая многозадачность», «многопроцессорная обработка». Какие требования предъявляются к современным операционным системам?
8. Перечислите критерии эффективности работы следующих систем: пакетной обработки, разделения времени и реального времени. В чем заключается особенность функционирования этих систем и для решения каких задач они используются ?
9. Сформулируйте назначение и назовите основные функции операционных оболочек компьютера. Приведите примеры известных программ-оболочек. В чем состоят их принципиальные отличия от операционных систем?
10. Для каких целей используется сервисное программное обеспечение (утилиты)? Приведите примеры известных программ-утилит.
11. Сформулируйте назначение, перечислите виды и основные функции прикладного программного обеспечения компьютера. На какие группы подразделяют прикладные программы?
12. В чем заключаются особенности программного обеспечения технологических процессов?

---

ГЛАВА 3. ОСНОВЫ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ	3-2
3.1. Традиционный подход к организации данных	3-2
3.2. Система баз данных	3-6
3.2.1. Данные	3-7
3.2.2. Аппаратное обеспечение	3-9
3.2.3. Программное обеспечение	3-9
3.2.4. Пользователи	3-10
3.3. Преимущества и недостатки современного подхода к организации данных	3-11
3.4. Классификация систем баз данных	3-13
3.5. Архитектура клиент/сервер	3-16
3.5.1. Компоненты приложений клиент/сервер	3-17
3.5.2. Разделение клиента и сервера	3-19
3.5.3. Преимущества и перспективы системы клиент/сервер	3-22
3.6. Общие понятия реляционного подхода к организации данных	3-22
3.6.1. Базовые понятия реляционных баз данных	3-25
3.6.2. Общая характеристика реляционной модели данных	3-29
3.6.3. Манипулирование данными с помощью языка запросов SQL	3-31
3.7. Основы проектирования реляционных баз данных	3-33
3.7.1. Основные требования при проектировании БД	3-33
3.7.2. Основы классической методологии проектирования БД	3-34
3.7.3. Основные этапы проектирования базы данных	3-37
3.7.4. Обеспечение свойств БД в процессе проектирования	3-40
3.8. Проектирование реляционных баз данных с использованием принципов нормализации	3-43
3.8.1. Первая нормальная форма	3-44
3.8.2. Вторая нормальная форма	3-45
3.8.3. Третья нормальная форма	3-47
3.9. Семантическое моделирование данных. Диаграммы «сущность–связь»	3-48
3.9.1. Основные понятия	3-48
3.9.2. Методология IDEF1	3-51
3.10. Информационное моделирование с помощью CASE-средства ERwin	3-55
3.10.1. Общая характеристика программы ERwin	3-55
3.10.2. Этапы построения информационной модели в ERwin	3-57
3.11. Проектирование базы данных доменного производства	3-59
3.11.1. Концептуальное и логическое проектирование	3-59
3.11.2. Физическая реализация информационной модели	3-66
3.12. Контрольные вопросы	3-72

## Глава 3. ОСНОВЫ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Основными функциями любой информационной системы являются хранение данных и их обработка. *Под термином «данные» мы будем понимать любую информацию об объектах окружающего мира, представленную в формализованном виде, пригодном для ее передачи, хранения и обработки при помощи некоторого процесса.* Для обеспечения процесса хранения и обработки данных средствами вычислительной техники во все современные информационные системы входит специальная программа, а точнее, комплекс программных средств, называемых системой управления базами данных (СУБД). Эта система обеспечивает создание, сопровождение<sup>1</sup> и использование специальных хранилищ данных – баз данных (БД), которые являются современной формой организации хранения и доступа к информации.

В данном разделе мы познакомимся с основными принципами организации данных, дадим классификацию баз данных по ряду признаков, рассмотрим особенности клиент/серверной архитектуры построения информационных систем. Далее мы перейдем к рассмотрению основ моделирования данных и характеристик реляционной модели организации данных, принятой в настоящее время на вооружение большинством фирм-разработчиков коммерческих версий СУБД. Завершающей частью данного раздела будет характеристика современных методов и средств проектирования баз данных.

### 3.1. Традиционный подход к организации данных

Как организовать хранение данных в компьютере, чтобы наилучшим образом, быстро и надежно извлекать и обрабатывать нужную информацию? Простейший подход состоит в разработке для отдельной прикладной задачи одной или нескольких компьютерных программ, называемых *прикладными программами*. Напомним, что программы, выполняемые под управлением операционной системы, в частности Windows, называют *приложениями*. Для каждого приложения во внешней памяти компьютера создается один или несколько файлов для размещения, хранения и извлечения из них информации о каком-нибудь элементе предметной области. Процедура размещения и извлечения информации в файле зависит от его типа, который в общем случае подразделяется на две категории:

- файл последовательного доступа;
- файл произвольного доступа.

Файл последовательного доступа представляет собой последовательность записей данных в виде строк произвольной длины, разделенных запятыми или специальными символами, обозначающими переход на новую строку. Особен-

---

<sup>1</sup> Сопровождение (ведение, поддержка) данных – термин, объединяющий действия по добавлению, удалению или изменению хранимых в компьютере данных.



ностью данного типа файла является отсутствие возможности упорядочить хранящиеся записи, размещение и извлечение записей в такой файл производится построчно в определенной последовательности. Файл произвольного доступа состоит из записей фиксированной длины, которая указывается при его создании. Все записи в таком файле упорядочены, каждая имеет свой номер, что позволяет быстро переместиться на любую запись, минуя предыдущие.

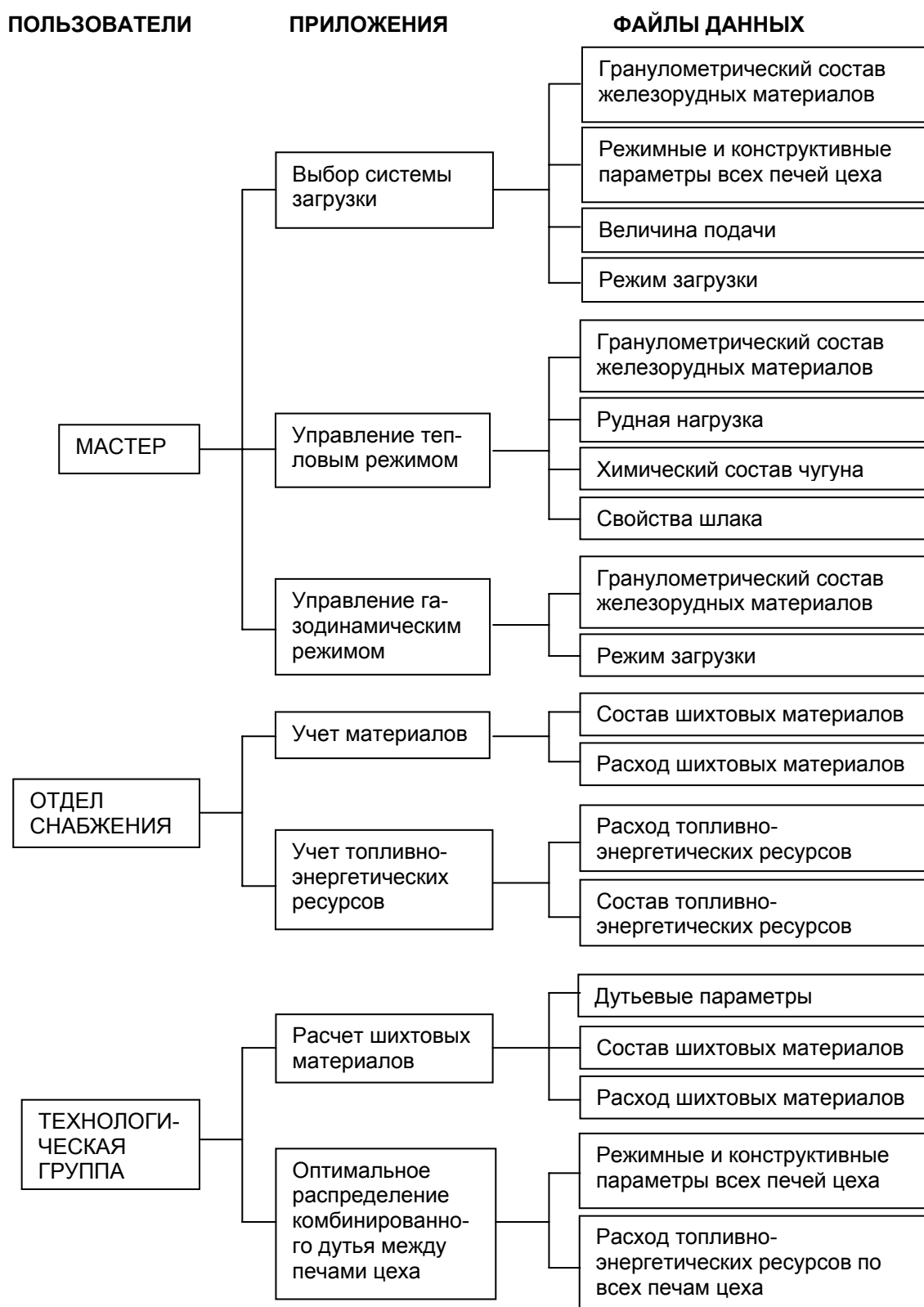
К примеру, если рассматривать в качестве предметной области доменное производство (**рис. 3.1**), то фрагмент такой информационной системы будет содержать следующие приложения:

- приложение для мастера печи, осуществляющего выбор системы загрузки, управление тепловым режимом конкретной печи и т.п.;
- приложение для отдела снабжения, решающего задачи по бесперебойному обеспечению работы доменного цеха материалами и топливно-энергетическими ресурсами;
- приложение для главного технолога, решающего задачи по анализу технологии доменной плавки, например оптимальному распределению материалов и топливно-энергетических ресурсов между несколькими печами цеха;
- приложение для технического управления предприятием, решающего задачи управления технологией доменного производства, анализа перспективных решений, планирования работы цеха и т.п.

Каждое приложение (прикладная программа) использует для обработки данные, расположенные в одном или нескольких файлах. Чтобы приложение правильно считывало данные из файлов, последние должны быть организованы по определенным правилам, которые «понятны» приложению. Другими словами, при разработке прикладной программы необходимо продумать и реализовать хранение данных таким образом, чтобы осуществлять доступ к ним наиболее эффективным способом. Критерием эффективности в большинстве случаев является скорость доступа к данным, которая зависит от множества факторов, в частности типа накопителя, способа доступа к записям файла, размера файла и др. В этом случае говорят, что логика организации файлов данных встроена в само приложение, т.е. в код его программы.

Такой подход, который в литературе принято называть *традиционным*, был характерен для начальных этапов развития информационных систем, но может встречаться и сейчас, когда специалист в области, не связанной с вычислительной техникой непосредственно, создает свою персональную информационную систему, пользуясь известными ему инструментами. Каковы *недостатки* такой организации данных?

**Избыточность данных.** Некоторые элементы данных неизбежно используются во многих приложениях. Поэтому они записываются в несколько файлов, т.е. одни и те же данные хранятся в разных местах. Примером является файл данных, содержащий информацию о характеристиках шихтовых материалов (см. **рис. 3.1**). Такое положение называют избыточностью данных. Это, в свою очередь, делает проблематичным обеспечение непротиворечивости информации, поскольку избыточность данных требует наличия нескольких процедур ввода и своевременного обновления, как правило, разными пользователями.



*Рис. 3.1. Традиционный подход к организации данных*

**Проблемы непротиворечивости данных.** Одной из причин нарушения непротиворечивости данных является их избыточность, что связано, как уже отмечалось, с хранением одной и той же информации в нескольких местах. При появлении новой информации, необходимой пользователям различных струк-

турных подразделений, ее надо ввести в разных местах. Зачастую по разным причинам выполнить это не удастся или ввод осуществляется с искажениями. В результате об одном и том же объекте предметной области в разных местах хранится различная информация.

**Ограниченная доступность данных.** В современных условиях, когда исключительно важна оперативность управления, лицо с соответствующими правами доступа должно иметь возможность получить данные за приемлемый отрезок времени. Если же данные разбросаны по нескольким файлам, доступность данных ограничена.

**Сложности в организации и управлении.** Жесткая зависимость между данными и использующими их программами создает серьезные проблемы в ведении данных и делает использование их менее гибким. Разработчики прикладных программ, написанных, например, на языках программирования Basic, Pascal или C, размещают нужные им данные в файлах, организуя их наиболее удобным для себя образом с целью достижения максимальной эффективности работы программы (повышение скорости доступа к данным, оптимизация расположения данных во внешней памяти, реализация усиленных средств защиты данных и др.). Система программирования накладывает на структуру загрузочного модуля прикладной программы специфичную для этой системы логику обработки данных. Одни и те же данные могут иметь в разных приложениях совершенно разную организацию (разную длину и последовательность размещения записей, разные форматы одних и тех же полей и т.п.). Обобщить такие данные очень трудно. Например, любое изменение структуры или способа доступа к файлу данных, производимое одним из разработчиков, приводит к необходимости изменения другими разработчиками тех прикладных программ, которые используют записи этого файла. Кроме того, из-за избыточности данных в файлах трудно реализовать новые изменения данных во всей предметной области.

Дополнительные трудности создания информационных систем на базе файловых систем проявились в следующем:

- недостаточность средств защиты хранимых данных;
- низкопроизводительная работа в многопользовательской среде;
- отсутствие процедур восстановления данных после возникновения отказов;
- отсутствие средств манипулирования данными;
- высокая стоимость программирования и сопровождения;
- негибкость к изменениям и др.

Попытки интеграции, объединения данных до появления технологии систем баз данных наталкивались на ряд трудностей. Информационные системы ориентированы, главным образом, на хранение, выбор и модификацию постоянно существующей информации о предметной области. Понятно, что эта информация каким-то образом должна быть упорядочена, т.е. структурирована. Логическая структура информации зачастую очень сложна, и хотя структуры организации данных различны в разных информационных системах, между ними часто бывает много общего. Стремление выделить и обобщить общую часть информационных систем, ответственную за управление сложно структурированными

данными, явилось, пожалуй, главной побудительной причиной создания систем управления базами данных (СУБД).

### 3.2. Система баз данных

В основе системы баз данных лежит принцип отделения данных, точнее, структуры данных, от процедурных предложений (управляющих команд) прикладных программ. Это позволяет накапливать, обрабатывать и манипулировать данными посредством специальных хранилищ – баз данных. **База данных (БД)** – это именованная совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, не зависящая от прикладных программ. Необходимо отметить, что база данных является информационной моделью, которая отображает состояние объектов и их отношений в рассматриваемой предметной области. Существуют различные модели организации данных, наиболее распространенной из которых является реляционная модель. Сопровождение баз данных, а также обеспечение взаимосвязи данных с прикладными программами осуществляется посредством специальной программы, точнее, комплекса программных средств – системы управления базами данными (СУБД). **Система управления базами данных (СУБД)** – комплекс программ и языковых средств, предназначенных для создания, сопровождения и использования баз данных.

Все работы по обслуживанию СУБД осуществляет человек – администратор базы данных (АБД). **Прикладные программы (приложения)**, осуществляю-

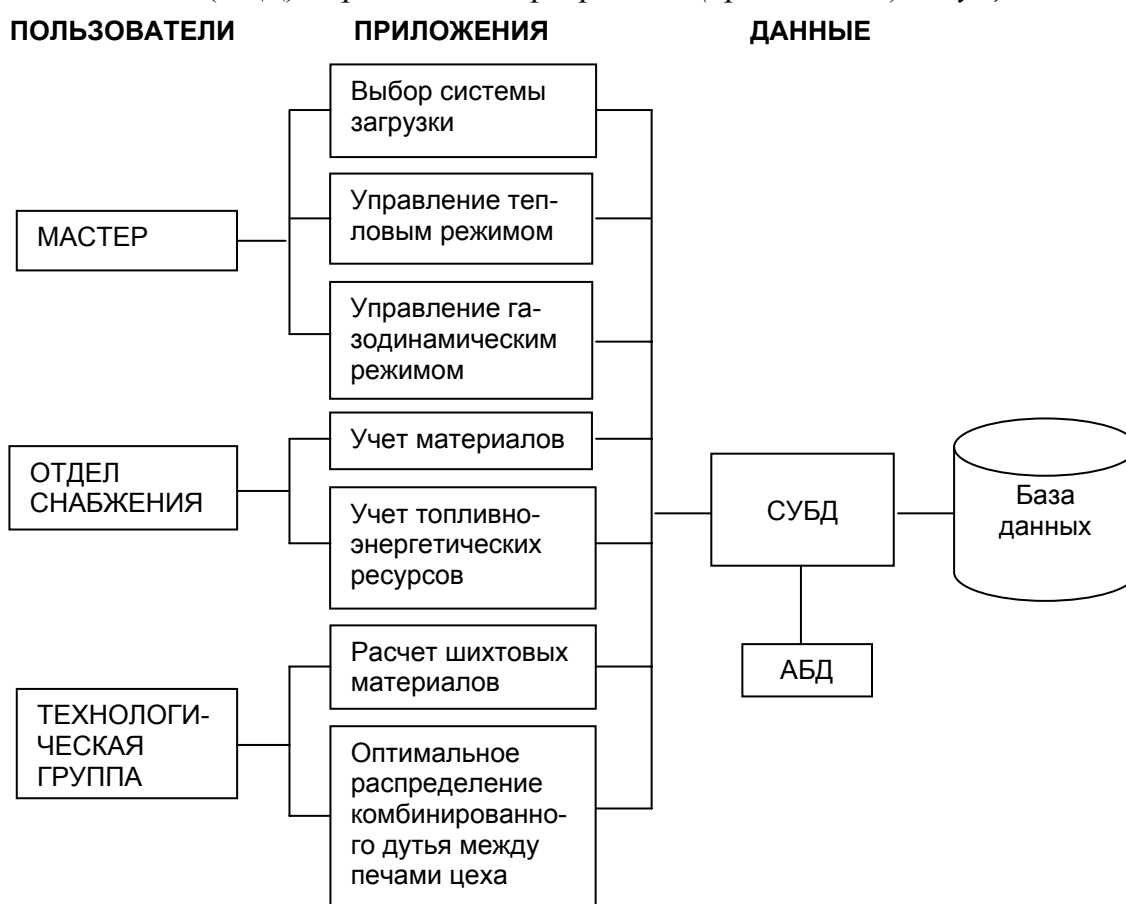


Рис. 3.2. Организация системы баз данных

щие доступ к базе данных, могут в общем случае добавлять, выбирать, копировать, корректировать, а также удалять данные, однако при этом они не могут изменять метод (модель), при помощи которого эти данные созданы. На **рис. 3.2** представлен фрагмент информационной системы доменного производства, основанной на системе баз данных.

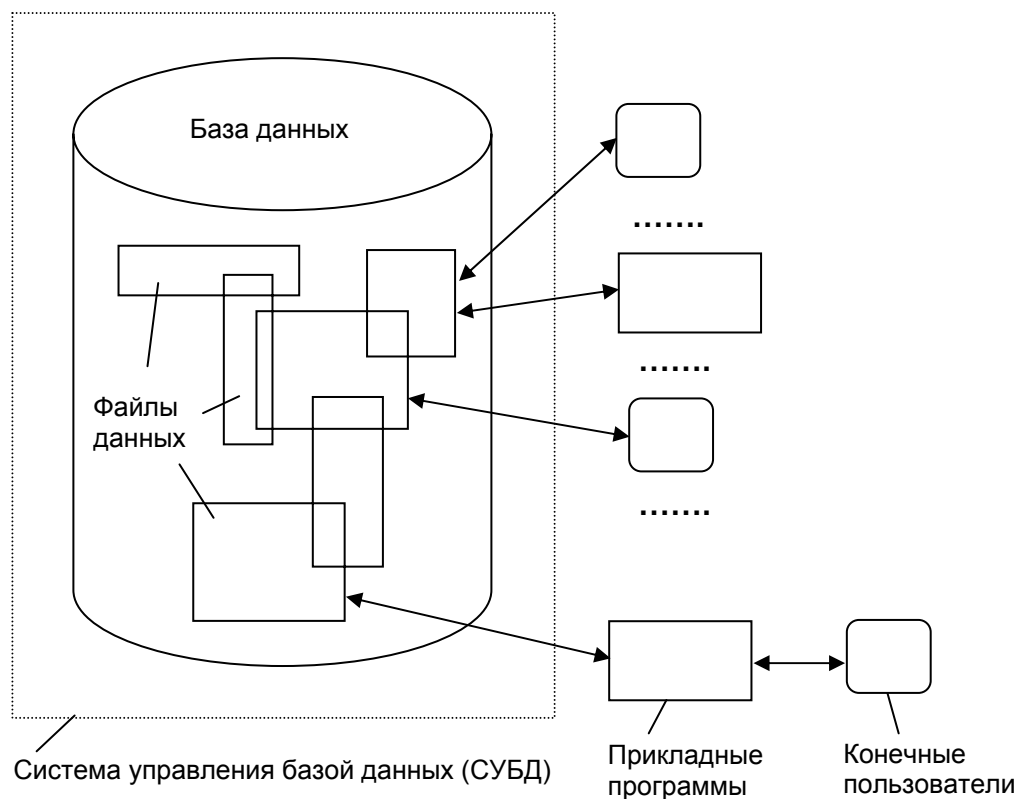
Очевидно, что для помещения информации в БД, ее хранения, манипулирования данными, обеспечения различным пользователям доступа к БД необходима система – **система баз данных**. На **рис. 3.3** показана упрощенная схема системы баз данных, которая содержит четыре главных компонента системы, а именно:

- данные;
- аппаратное обеспечение;
- программное обеспечение;
- конечные пользователи, прикладные программы.

Рассмотрим эти компоненты подробнее.

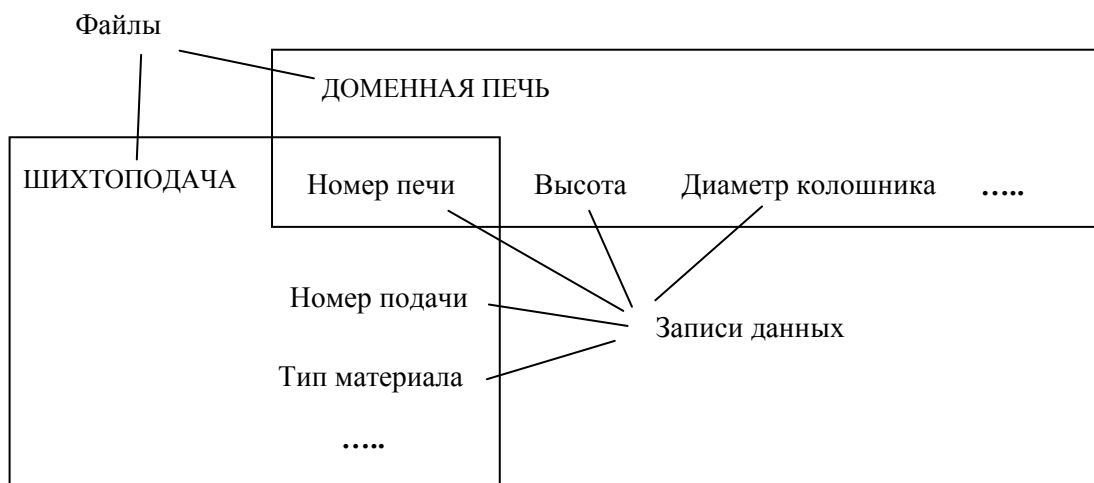
### 3.2.1. Данные

В начале раздела было дано определение термина «данные» как формализованного представления информации из какой-либо предметной области. Особенностью организации данных в БД является то, что они являются *интегрированными* и *общими*. Эти два аспекта, интеграция и разрешение общего доступа, представляют собой наиболее важное преимущество использования системы баз данных.



**Рис. 3.3.** Упрощенная схема системы баз данных

Под понятием *интегрированные* данные подразумевается возможность представить БД как объединение нескольких отдельных файлов данных, полностью или частично не перекрывающихся. Например, база данных может содер-



*Рис. 3.4. Фрагмент организации данных в базе данных*

жать файл «ДОМЕННАЯ ПЕЧЬ», включающий записи о номере печи, геометрических характеристиках ее конструктивных элементов, количестве чугуновых леек и др., а также файл «ШИХТОПОДАЧА», в котором регистрируется информация о загружаемых в эту печь материалах (рис. 3.4). Допустим, что для контроля процесса загрузки нам потребуется узнать некоторые геометрические характеристики доменной печи. Очевидно, что нет необходимости дополнительно включать такую информацию в файл «ШИХТОПОДАЧА», поскольку ее всегда можно получить путем обращения к файлу «ДОМЕННАЯ ПЕЧЬ».

Одним из следствий интегрированности базы данных является то, что любой конкретный пользователь обычно имеет отношение к какой-то небольшой части всей БД. Иначе говоря, такая база данных может восприниматься различными пользователями по-разному, т.е. иметь несколько назначений, соответствующих различным внешним представлениям о хранимых данных. *В этом состоит основное различие между базой данных и файлом: несколько назначений может иметь и файл, но соответствует при этом он только одному представлению.*

Под понятием *общие* данные подразумевается возможность использования отдельных областей данных в БД несколькими различными пользователями, т.е. каждый из этих пользователей может иметь доступ к одной и той же области данных (даже в одно и то же время), причем различные пользователи могут использовать эти данные для разных целей. В приведенном выше примере информация из файла «ШИХТОПОДАЧА» может совместно использоваться, разделяться: мастером печи – для контроля процесса загрузки, экономическим отделом – для подсчета количества и стоимости загруженных в печь материалов за определенный срок, отделом снабжения – для подсчета статистических показателей расхода определенного материала с целью планирования предстоящих его запасов и т.п.

### **3.2.2. Аппаратное обеспечение**

Аппаратное обеспечение системы представляет собой комплекс физических устройств, на которых существует база данных. Оно состоит из одного или нескольких компьютеров, дисководов, мониторов, принтеров, магнитофонов, соединительных кабелей и других вспомогательных и соединительных устройств.

Для обработки данных в базах данных могут использоваться универсальные компьютеры, мини-компьютеры или персональные компьютеры. Универсальные компьютеры и мини-компьютеры традиционно используются в качестве основы для поддержания доступа многих пользователей к общей базе данных. Персональные компьютеры часто используются для баз данных, с которыми работает один пользователь.

Основным способом хранения баз данных является хранение на дисках, поскольку к ним возможен произвольный доступ, без которого работать с базой данных невозможно. Настольные компьютеры, мониторы и принтеры используются для ввода и извлечения информации из базы данных. Ленты являются недорогим средством быстрого создания резервных копий хранящихся на дисках данных.

Успешная работа современных информационных систем сильно зависит от совершенства технологии аппаратного обеспечения. Для поддержания и управления огромным количеством данных, хранящихся в базе данных, требуется очень большой объем оперативной памяти и дисков. Кроме того, в системе с большим числом пользователей для того, чтобы за приемлемое время извлекать нужную им информацию, необходимо быстро выполнять огромное количество обращений к данным, для чего требуется высокое быстродействие компьютеров, сетей и периферийных устройств. К счастью, за время развития баз данных мощность компьютерного оборудования заметно выросла, а стоимость значительно снизилась. Благодаря этому стало возможным широкое распространение информационных систем, использующих базы данных.

### **3.2.3. Программное обеспечение**

Между собственно физической базой данных, т.е. данными, которые в действительности сохранены во внешней памяти компьютера, и конечными пользователями системы располагается уровень программного обеспечения – система управления базами данных. СУБД располагает средствами добавления, выборки, обновления и удаления данных, а также другими компонентами, обеспечивающими настройку системы базы данных, тестирование, выполнение вспомогательных функций, таких как восстановление баз данных, сбор статистики о функционировании системы и др.

Языковые средства СУБД являются важнейшей компонентой системы баз данных, поскольку в конечном счете они обеспечивают интерфейс пользователей разных категорий с базой данных. Современные СУБД обладают достаточно широким спектром языковых средств, среди которых выделяют языки описания данных, языки манипулирования данными, языки запросов и др.

С помощью языка описания данных (ЯОД) определяется схема, или структура организации данных, в виде объектов и связей между ними. В основу той или иной структуры положена концептуальная модель данных, отражающая в базе данных наиболее существенные с точки зрения пользователей особенности предметной области. Более подробно концептуальная модель будет рассмотрена позднее при обсуждении этапов проектирования баз данных.

Языки манипулирования данными (ЯМД) определяют допустимые операции, которые можно выполнить на основе выбранной структуры данных, и разделяются на две большие группы: процедурные и непроцедурные. При использовании *процедурных языков* требуется указать, какие действия и над какими объектами базы данных необходимо выполнить, чтобы получить результат. Как правило, такие языки ориентированы на обработку отдельных записей данных. В *непроцедурных языках* указывается, что надо получить в результате, а не как этого достичь. Они основаны на реляционном исчислении и ориентированы на операции сразу над множеством записей.

Большинство операций по обработке данных может быть выполнено с помощью языка запросов. *Запрос* – входное сообщение к информационной системе на выделение и изменение информационных ресурсов, содержащее условие на поиск, выборку и обработку данных.

В зависимости от возможностей языка запросов и системы хранения данных в тексте запроса могут быть указаны дополнительные действия, такие как упорядочение найденных данных, выполнение элементарных вычислений и др. В настоящее время основным языком запросов, общения с базами данных является язык **SQL (Structured Query Language)** – язык структурированных запросов. Язык является непроцедурным языком и не содержит имеющихся в обычных языках программирования операторов управления, организации подпрограмм, ввода-вывода и т.п. Формулирование запроса к базе данных на проведение определенных операций над данными производится на естественном для пользователя языке (английском).

Таким образом, основной функцией, выполняемой СУБД, является предоставление пользователю базы данных возможности работать с ней, не вникая в детали на уровне аппаратного обеспечения.

### 3.2.4. Пользователи

Всех пользователей можно условно разделить на три большие группы:

1. **Прикладные программисты**, которые отвечают за написание прикладных программ, использующих БД. Для этих целей применяются языки программирования приложений Pascal, C, Visual Basic и др. Прикладные программы выполняют над данными все стандартные операции: выборку, обновление или удаление существующей информации, а также вставку новой информации. Все эти функции выполняются через соответствующий запрос к СУБД.
2. **Конечные пользователи**, которые работают с системами баз данных непосредственно через рабочую станцию или терминал. В общем случае пользователь может получить доступ к БД, используя командный интерфейс (интегрированный интерфейс программного обеспечения самой СУБД) или пользо-



вательский интерфейс прикладной программы. *Командный интерфейс*, т.е. язык запросов, требует хорошего знания внутренней структуры организации данных, объектов и связей между ними, а также опыта работы с информационной системой. Поэтому общение конечного пользователя с информационной системой в большинстве случаев осуществляется средствами *пользовательского интерфейса* прикладной программы. В этом случае он в явном виде не использует команд-запросов на извлечение данных, а для работы с базой данных выбирает необходимые команды меню или заполняет поля в специальных формах. Такие пользовательские интерфейсы, основанные на меню и формах, разрабатываются прикладными программистами и облегчают работу с базами данных тем, кто не имеет достаточного опыта работы с информационными системами. Однако пользовательский интерфейс по сравнению с командным интерфейсом имеет ограниченные возможности.

3. **Администраторы данных (АД)**, на которых возлагается вся ответственность за сохранность данных. Основные функции администрирования заключаются в том, чтобы принимать решение, какие данные необходимо вносить в базу данных, а также обеспечивать поддержание порядка при обслуживании данных и использовании их после занесения в базу данных. Например, администратор должен указывать, кто, при каких условиях, над какими данными и какие операции может выполнять. Другими словами, основная функция администратора данных – обеспечивать *безопасность данных*. Администратор данных – не «владелец» базы данных, а ее «хранитель». С усложнением предметной области усложняется также процесс формирования информации и принятия решений. В результате расширяется спектр функций администрирования. Очень важно, чтобы администратор данных работал как управляющий, а не как специалист по техническим вопросам, хотя он, конечно, должен иметь хорошее представление о возможностях систем баз данных на техническом уровне. *Технический* специалист, ответственный за реализацию решений администратора данных, – это **администратор базы данных (АБД)**. Администратор базы данных, в отличие от администратора данных, должен быть *профессиональным специалистом* в области информационных технологий, как правило, системным программистом по образованию. Работа АБД заключается в создании самих баз данных и техническом контроле, необходимом для осуществления решений АД.

### 3.3. Преимущества и недостатки современного подхода к организации данных

*Преимущества* централизованного подхода к управлению данными заключаются в реализации следующих возможностей.

- **Сокращение избыточности** данных (одинаковой информации)  
Как уже было отмечено, в системах, не использующих базы данных, каждое приложение использует свои файлы для размещения, хранения и извлечения из них информации. Имея систему баз данных, эти файлы можно объединить с устранением избыточности. Надо отметить, что не всегда следует стремиться-

ся *полностью* устранять избыточность данных. Иногда веские причины (например, повышение отказоустойчивости информационной системы) требуют наличия нескольких копий хранимых данных. Однако такая избыточность должна строго контролироваться, т.е. учитываться в СУБД путем использования процедуры *множественного обновления* – если обновление вносится в одну запись, то оно автоматически должно распространяться на все остальные.

- *Устранение противоречивости* данных

Очевидно, что если какой-либо факт из предметной области представлен в БД одной записью, т.е. при отсутствии избыточности, то противоречий возникнуть не может. Если не удалять избыточность, а *контролировать* ее средствами СУБД, например с помощью процедуры множественного обновления, тогда противоречий также можно избежать.

- *Общий доступ* к данным

Общий доступ к данным означает возможность доступа к ним нескольких существующих приложений базы данных и разработки новых приложений для работы с этими же данными. Другими словами, новые приложения могут получить доступ к тем же данным, и при этом нет необходимости в создании новых данных.

- *Соблюдение стандартов*

Благодаря централизованному управлению, АБД может обеспечивать представление данных в определенных стандартах. Стандартизация представления данных наиболее важна для обмена и перенесения данных между различными системами баз данных. Кроме того, стандарты определения данных и документации важны для совместного использования данных на предприятии, а также для их правильного понимания.

- *Введение ограничений для обеспечения безопасности*

Благодаря полному контролю над базой данных, АБД может обеспечить доступ к БД через определенные каналы, а следовательно, может определить правила безопасности, которые будут проверяться при попытке доступа к уязвимым данным. Для разных типов доступа (выборки, вставки, удаления и т.д.) и разных частей БД (таблиц, строк, полей и т.д.) можно установить различные правила. Однако надо иметь в виду, что при отсутствии таких правил безопасность данных подвергается большему риску, чем в обычной, разрозненной файловой системе при традиционном подходе к организации данных. Следовательно, централизованная природа системы организации данных *требует* наличия повышенной системы безопасности.

- *Обеспечение целостности* данных, т.е. правильности и точности данных в базе данных

Задача целостности заключается в обеспечении правильности и точности данных в БД. Противоречие между двумя записями, представляющими один «факт», является примером недостатка целостности. Эта проблема может возникнуть лишь при наличии избыточности в хранимых данных. Но даже если избыточность отсутствует, база данных может содержать неправильную информацию. Например, сотрудник может иметь в соответствии с хранимыми данными 400 рабочих часов в неделю вместо 40 или принадлежать к не-

существующему отделу №5. Централизованное управление БД позволяет в какой-то мере избежать подобных проблем. Для этого АД определяет, а АБД реализует правила целостности, которые применяются при любой попытке проделать какую-либо операцию обновления информации в базе данных (вставки, удаления и т.д.). Введение правил целостности для работы в многопользовательской БД более важно, поскольку к такой базе данных есть общий доступ. При отсутствии должного контроля один пользователь может некорректно обновить данные, от чего пострадают другие пользователи.

Необходимо отметить, что интеграция данных какой-либо предметной области может иметь и ряд *недостатков*:

- пользователи *теряют право единоличного владения* данными, что может привести к снижению ответственности и, как следствие, к уменьшению достоверности данных;
- *повышается вероятность нарушений защиты данных*, если не предусмотрен соответствующий механизм разграничения прав доступа к базе данных;
- *ставится под угрозу секретность хранимой информации*, поскольку в интегрированной базе данных легче накапливать сведения и затем выдавать их лицу или организации, доступ которых к ней не разрешен.

Поэтому, если система управления базами данных не гарантирует соответствующий уровень непротиворечивости, защиты и секретности, ее применение может породить новые проблемы.

В связи с этим можно сформулировать ряд *требований*, которым должна удовлетворять современная СУБД:

- эффективное выполнение различных функций предметной области;
- минимизация и контроль избыточности хранимых данных;
- предоставление для принятия решений непротиворечивой (согласованной) информации;
- обеспечение возможности одновременного доступа к базе данных нескольких уполномоченных пользователей;
- обеспечение управления безопасностью;
- простая физическая реорганизация, т.е. изменение структуры данных в базе данных;
- возможность централизованного управления базой данных.

### 3.4. Классификация систем баз данных

По характеру использования системы баз данных делят на однопользовательские (персональные) и многопользовательские.

**Однопользовательские системы** – это системы, в которых в одно и то же время к базе данных может получить доступ не более одного пользователя, так называемые базы данных с локальным доступом.

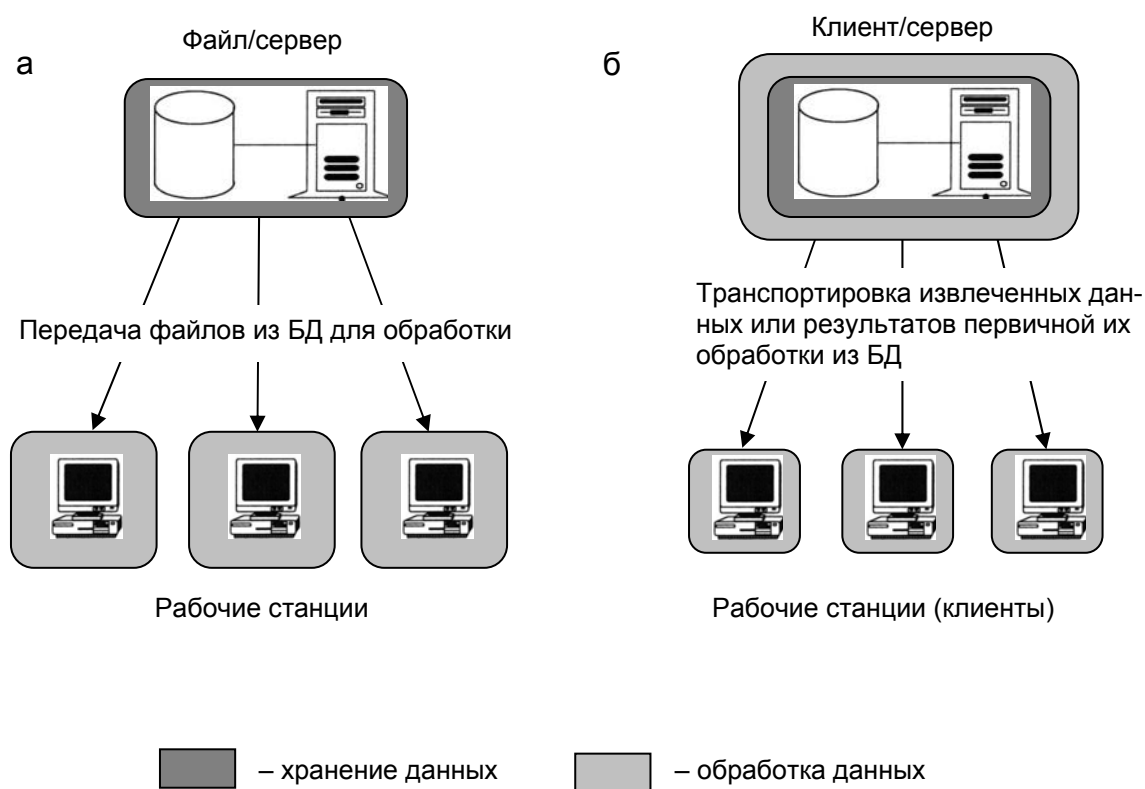
**Многопользовательские системы** – это системы, в которых к базе данных могут получить доступ одновременно несколько пользователей, так называемые базы данных с удаленным (сетевым) доступом. Различия между многопользовательской и однопользовательской системами проявляется в их внутренней структуре построения, различии используемых аппаратно-технических

средств и потому практически не видны конечному пользователю. Основная задача многопользовательских систем – позволить каждому отдельному пользователю работать с системой как с однопользовательской.

Современные многопользовательские системы централизованных баз данных с удаленным (сетевым) доступом предполагают различные архитектуры построения подобных систем:

- файл/сервер;
- клиент/сервер.

Архитектура построения систем БД типа *файл/сервер* предполагает выделение одного из компьютеров сети в качестве центрального (сервер файлов). На таком компьютере хранится совместно используемая централизованная БД. Все другие компьютеры сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в большинстве случаев и производится обработка данных. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать также на рабочих станциях локальные БД, которые используются ими монополюно. Концепция файл/сервер условно отображена на **рис. 3.5, а**.



**Рис. 3.5.** Схема обработки информации в БД при принципе файл/сервер (а) и клиент/сервер (б)

В концепции *клиент/сервер* подразумевается, что помимо хранения централизованной базы данных центральный компьютер – сервер базы данных –

должен обеспечивать выполнение некоторого объема обработки данных. Запрос на данные, выдаваемый клиентом (приложением с рабочей станции), порождает поиск, извлечение, а в некоторых случаях даже первичную обработку данных на сервере. Извлеченные данные или результаты их первичной обработки (но не файлы) передаются по сети от сервера к клиенту. Спецификой архитектуры клиент/сервер является использование специального языка запросов SQL. Концепция клиент/сервер условно отображена на **рис. 3.5, б**. В дальнейшем мы рассмотрим особенности данной архитектуры более подробно.

По **технологии обработки** данных системы баз данных подразделяются на централизованные и распределенные.

**В централизованной системе** база данных физически хранится в памяти одного компьютера. Если он является компонентом компьютерной сети, возможен распределенный, т.е. из различных узлов сети, доступ к такой базе. Такой способ использования баз данных часто применяют в локальных сетях.

**Распределенная система баз данных** состоит из нескольких, возможно, пересекающихся или даже дублирующих друг друга частей БД, хранимых в различных компьютерах-серверах, которые в общем случае могут быть удалены географически друг от друга на значительные расстояния, т.е. территориально распределены. Связь между различными компьютерами осуществляется с помощью программного обеспечения для управления компьютерной сетью, основанного на архитектуре клиент/сервер. Такое программное обеспечение, выполняющее сетевые функции, часто является встроенным компонентом операционной системы компьютера, например, Windows NT.

Преимущество распределенной обработки данных проявляется в соответствии практическим условиям работы многих предприятий и организаций. К примеру, в практике работы металлургического комбината, использующего единую информационную систему, обработка данных выполняется таким образом, что полный набор всех данных не сохраняется на одном компьютере, а распределяется на отдельных компьютерах, для пользователей же организован доступ к данным нескольких компьютеров. В частности, данные о работе доменного цеха хранятся на одном сервере, агломерационного цеха – на другом, рудообогащительной фабрики – на третьем и т.д. При этом серверы, как правило, расположены на территории соответствующих цехов. Пользователь (клиент) из управления комбинатом, составляющий сводку о работе предприятия в целом, может получить доступ к любому количеству цеховых серверов одновременно, т.е. за один запрос можно получить комбинированные данные двух или более серверов. В этом случае серверы рассматриваются клиентом как один с логической точки зрения, и пользователь может не знать, на каком именно компьютере какая часть данных содержится.

В общем случае компьютеры клиентов могут иметь свои собственные сохраняемые данные, а компьютер сервера может иметь свои собственные приложения, формирующие запросы. В этом случае каждый компьютер будет выступать в роли сервера для одних пользователей и в роли клиента для других (**рис. 3.6**). Это пример полной распределенной системы баз данных. Полная поддержка для распределенных баз данных означает, что отдельное приложение может «прозрачно» обрабатывать данные, распределенные на множестве раз-

личных баз данных, управление которыми осуществляют разные СУБД, работающие на многочисленных компьютерах с различными операционными системами, соединенных вместе коммуникационными сетями. Термин «прозрачно» означает, что приложение выполняет обработку данных с логической точки зрения, как будто управление данными полностью осуществляется одной СУБД, работающей на отдельном компьютере.

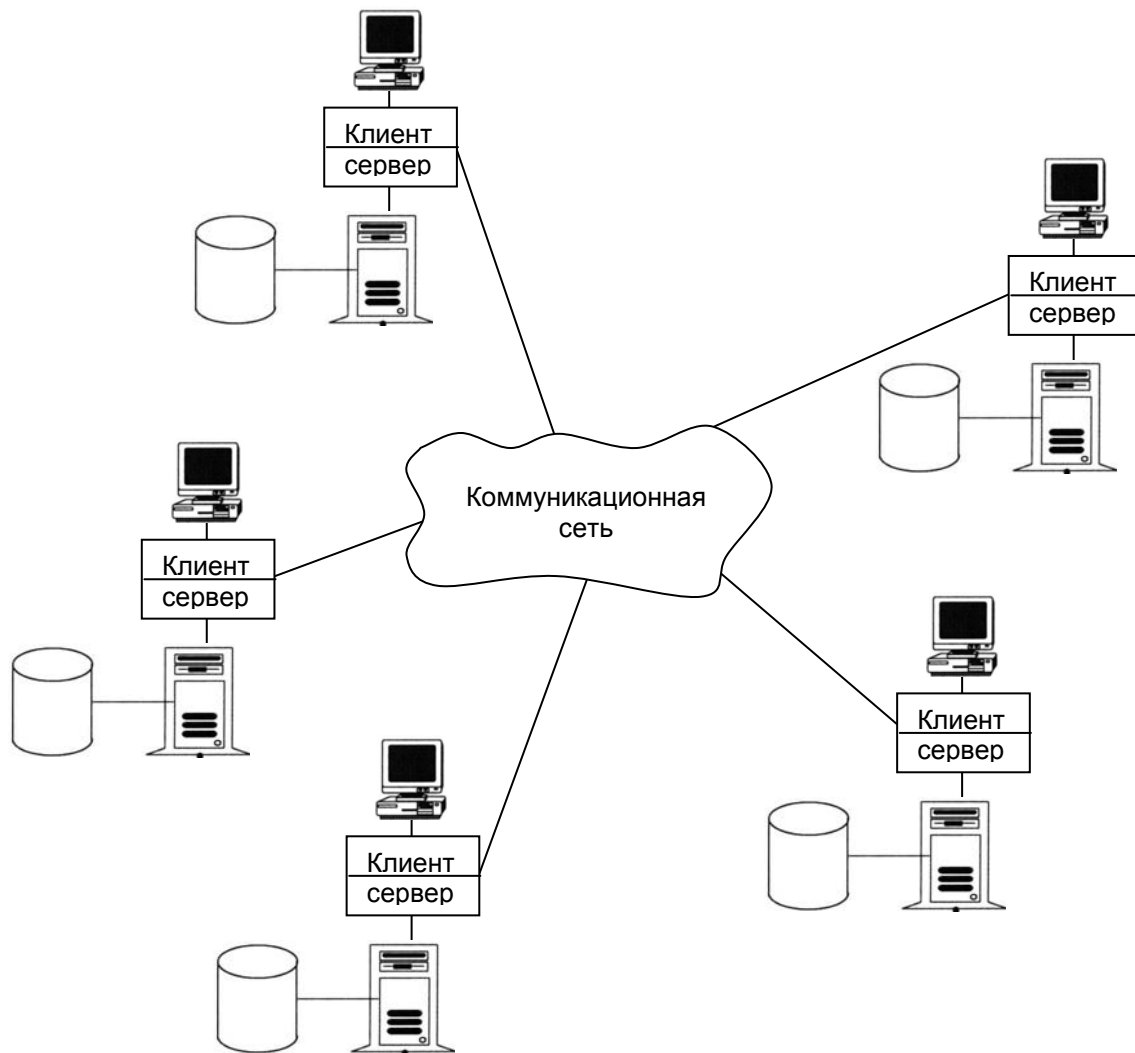


Рис. 3.6. Полная распределенная система обработки БД

### 3.5. Архитектура клиент/сервер

Архитектура клиент/сервер в той или иной форме стала общераспространенной при общении с компьютерной информационной системой. Приложение, разработанное в архитектуре клиент/сервер, можно определить как состоящее из *нескольких частей других приложений*, выполняемых на различных компьютерных системах. Одно из основных назначений архитектуры клиент/сервер – достижение высокоэффективного использования ресурсов компьютера – процессора, оперативной и внешней памяти, периферийных устройств – путем разбиения

имеющейся задачи на более мелкие части (процессы) и возможность распределения работы по выполнению задачи между двумя или более доступными компьютерами. Правильное построение приложения клиент/сервер предполагает, что процессы протекают именно там, где наиболее эффективно выполняется работа, специфичная для каждого из них.

### 3.5.1. Компоненты приложений клиент/сервер

Ключ к успешному выполнению программы клиент/сервер – баланс ее различных компонентов. Можно реализовать очень простую и недорогую архитектуру, эффективную для небольшого числа пользователей и/или простых приложений. Однако для поддержки большого числа пользователей или очень сложного приложения необходима более сложная архитектура.

Для реализации компьютерного приложения с архитектурой клиент/сервер в первую очередь необходимо понимание функций различных уровней, обычно существующих внутри этого приложения. На **рис. 3.7** показаны некоторые потенциальные уровни любого приложения, которое функционирует в общем случае на одном компьютере.



**Рис. 3.7.** Уровни компьютерного приложения

Крайним слева на этом рисунке изображен конечный пользователь-клиент данной архитектуры – тот, кто запрашивает услуги приложения. Общение пользователя с приложением происходит через **пользовательский интерфейс**, который обеспечивает непосредственный ввод данных, запросов к системе на извлечение данных, а также представление информации в удобном для пользователя виде.

При продвижении по схеме слева направо каждый из уровней, расположенных левее, можно рассматривать в качестве клиента для сервера, формируемого уровнями, расположенными правее. Так, пользователь приложения через свой интерфейс запрашивает выполнение **прикладной задачи**, например ввод данных о химическом анализе пробы чугуна, расчет израсходованного за смену кокса в доменном цехе или оформление квартального отчета. Для системы баз

данных прикладная задача представляет собой последовательность команд, обрабатываемых управляющей программой, как правило, операционной системой, и требующей для этого выделения ресурсов компьютера – процессора, оперативной и внешней памяти и др.

Прикладные задачи приложения должны соответствовать **бизнес-правилам**, или **коммерческим ограничениям**, которые позволяют установить характерные для прикладной (предметной) области ограничения на вводимые данные. С их помощью производится *логическая* проверка вводимой информации, контроль за правильностью и полнотой заполнения базы данных. В общем случае такие ограничения могут быть произвольной сложности и включать в себя запросы на проверку информации из нескольких таблиц базы данных, в том числе и расположенных на других компьютерах. Например, оператор не может ввести данные химического анализа кокса, если предварительно не была введена информация о дате взятия соответствующей пробы на анализ.

Требования предметной области (бизнес-правила) подчиняются **правилам целостности данных**. Целостность, как уже было отмечено, – это состояние данных, когда они сохраняют свое информационное содержание и однозначность при *физических* модификациях базы данных, т.е. замене, добавлении, удалении данных, а также в условиях случайных воздействий (сбой системы). Например, данные о результатах химического анализа может вводить и корректировать только пользователь с соответствующими правами доступа к базе данных, при этом все вводимые величины должны выражаться двузначным неотрицательным числом.

Отметим принципиальное отличие бизнес-правил от правил целостности. Правила целостности используют для описания (при проектировании структуры данных) и поддержания (в процессе функционирования информационной системы) точности и корректности данных, хранящихся в базе данных. В соответствии с концепцией централизованного хранения и управления данными, которая, как уже было отмечено, заложена в основу создания большинства современных информационных систем, база данных используется для обслуживания многих прикладных программ из нескольких предметных областей. В каждой предметной области существуют свои, специфические правила, которые учитываются при проектировании базы данных в виде правил целостности. Уровень бизнес-правил позволяет выполнить логическую проверку вводимых данных на соответствие требованиям предметной области, сформировав соответствующий запрос на следующий уровень – проверку правил целостности данных. Важно подчеркнуть, что на следующий уровень посылается запрос в виде какой-либо логической функции, но не сами данные прикладной программы.

Например, база данных доменного цеха обслуживает множество прикладных программ, в том числе для мастера печи, технологической группы и экономистов. Очевидно, что задачи, решаемые с помощью прикладных программ, для всех групп специалистов различны. Каждая прикладная программа отвечает за правильный ввод и корректировку своей части информации, вводимой в общую базу данных. Информация по загружаемым шихтовым материалам формируется в базе данных от нескольких клиентов: мастер печи вводит количество загружаемых в печь материалов, технологи – данные по химическому анализу, эконо-



мисты – стоимостные характеристики. Очевидно, что при вводе и корректировке соответствующей информации возникают свои специфические требования. Например:

- число, отражающее количество загружаемых шихтовых материалов, не должно превышать определенную величину, соответствующую общему количеству материалов в бункерах. Поэтому на следующий уровень отправляется запрос о суммарном количестве шихтовых материалов в бункерах;
- данные по химическому анализу материалов должны вводиться только определенным пользователем. На следующий уровень посылается запрос на проверку прав зарегистрировавшегося пользователя вводить указанную информацию;
- данные о стоимости шихтовых материалов не должны превышать величины, указанной в приходном ордере для партии этих материалов. На следующий уровень посылается запрос о проверке этого ограничения.

В случае соответствия данных правилам целостности осуществляется **доступ к базе данных**, который использует устройство **доступа к файлам** данных, управляемое средствами операционной системы. Среди функций, обеспечиваемых этими уровнями, можно отметить следующие:

- разграничение прав доступа к базе данных пользователей (чтение, изменение или удаление данных);
- возможность одновременной работы нескольких пользователей с базой данных (многопользовательский доступ).

Отметим, что именно благодаря *программному* разделению системы на несколько уровней, выполняющих свои специфические функции, появилась возможность *физической* обработки данных на разных компьютерах (клиенте и сервере), т.е. распределенной обработки. Эта возможность настолько важна, что термин «клиент/сервер» стал применяться главным образом в случае, если сервер и клиент находятся на разных компьютерах.

### 3.5.2. Разделение клиента и сервера

Напомним, что если все уровни приложения выполняются на одном компьютере, значит, это однопользовательское приложение. Для создания архитектуры клиент/сервер необходимо определить сферы действия клиента и сервера по границе любого из уровней приложения. Самое простое разделение клиент/сервер можно получить, выбрав ближайшую границу к любому концу данной архитектуры.

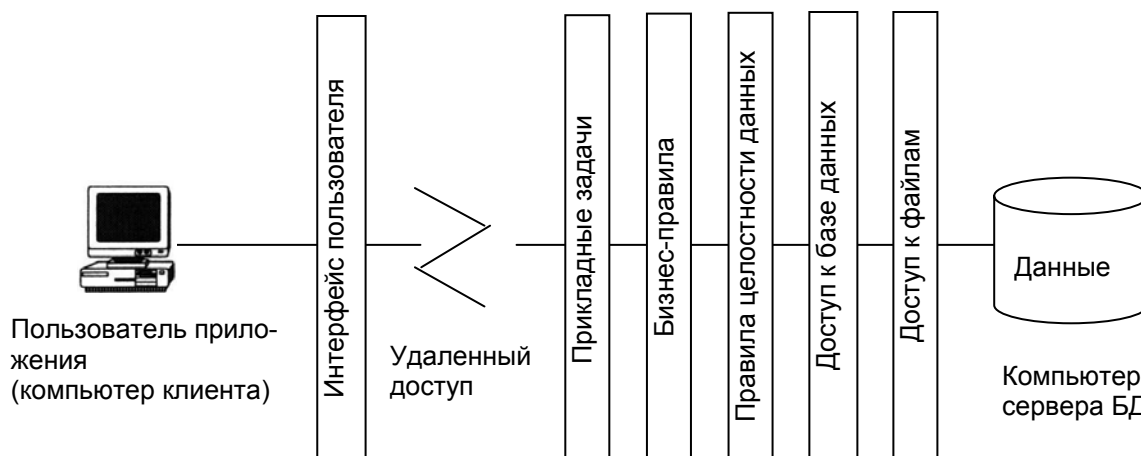
#### *Выполнение основной работы на сервере*

Один из простых способов разделения функций клиента и сервера – поручить клиенту только интерфейс пользователя, а остальное предоставить серверу (рис. 3.8).

Здесь в системе пользователя размещена часть приложения, ответственная

за просмотр данных, ввод с клавиатуры или с помощью мыши. Остальная часть выполняется на большой распределенной системе.

Иногда компьютер клиента представляет собой не более чем "интеллектуальный" терминал данных. В клиентской части приложения выполняется, как максимум, проверка правильности ввода простых данных.



*Рис. 3.8. Размещение большей части задач на сервере*

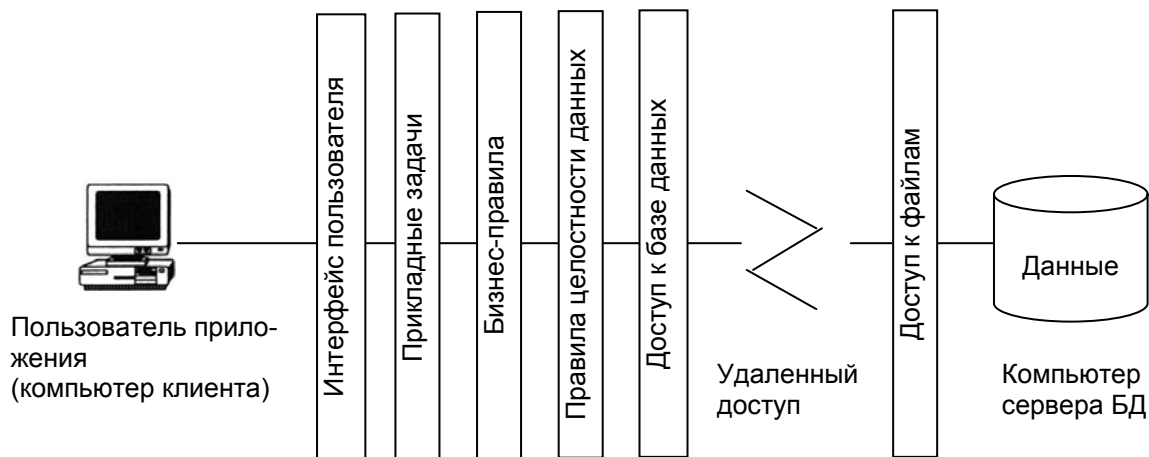
Такая архитектура подобна архитектуре большинства диалоговых сервисных систем. Компьютер пользователя осуществляет связь и обеспечивает обзор программного обеспечения (например, систему навигации и просмотра информации в сети Web), а основную работу выполняют компьютеры сети, с которыми соединен пользователь. В этом случае, несмотря на то что клиент обеспечивает только представление и перемещение данных, его компьютер должен быть достаточно мощным, чтобы быстро оперировать со сложными потоками графических данных, идущими с сервера.

Большинство профессионалов в области компьютерной технологии полагают, что такой тип простого разграничения не определяет правильность приложения клиент/сервер. И все-таки, если программное обеспечение компьютера или рабочей станции клиента выполняет важную работу по обеспечению пользователя усиленным графическим интерфейсом и по отправлению запросов на сервер с помощью простого потока символьных данных, такой вид приложения соответствует духу вычислений типа клиент/сервер.

Преимущество такой архитектуры – все данные и большинство прикладных логических средств сосредоточены в одном месте. Большинство изменений или усовершенствований приложения можно выполнить на центральном сервере. Недостаток – центральный сервер должен быть очень мощным, если доступ к данному приложению необходим многим пользователям. Поскольку связь между клиентом и сервером может потребовать работы с большими объемами графической информации, для получения достаточно быстрых ответов на запросы может понадобиться очень дорогая высокоскоростная компьютерная сеть, связывающая клиента с сервером.

### Выполнение основного объема работы у клиента

Противоположный описанному в предыдущем разделе вариант – разделение приложения таким образом, что сервер работает только как простое хранилище данных, или файловый сервер (**рис. 3.9**). Все представления данных и прикладные логические средства размещены на клиентском компьютере.

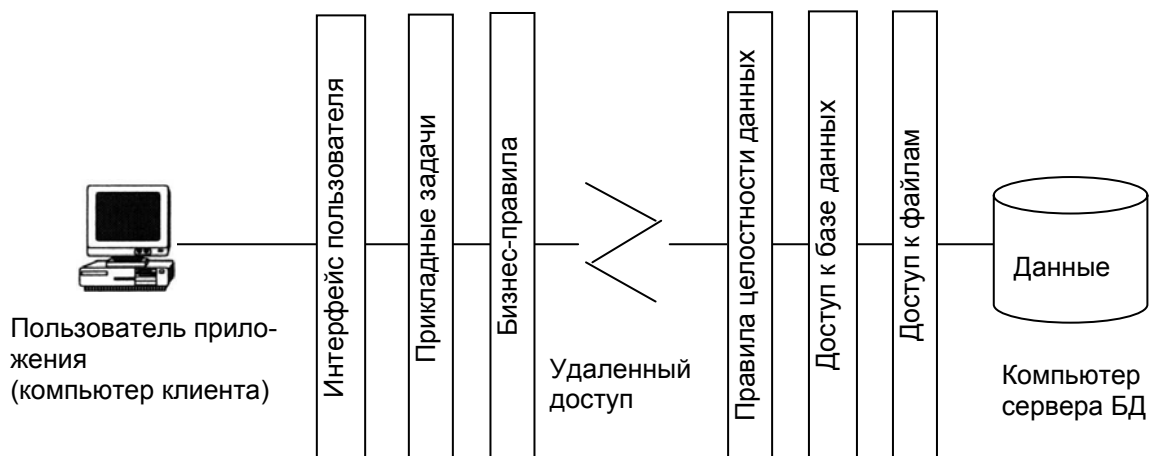


**Рис. 3.9.** Размещение большей части задач на клиенте

Основное достоинство такой архитектуры – ее простота. Для совместного использования данных системой клиент/сервер достаточно переместить файлы данных на сервер управления файлами. Однако эта архитектура требует достаточно сложного аппаратного обеспечения клиента, и, поскольку большинство файловых серверов осуществляет только первичное распределение данных на уровне файлов, она предусматривает только доступ к данным или относительно малое число совместно работающих пользователей.

### Достижение баланса

Для использования преимуществ архитектуры клиент/сервер в полной мере нужно так сконструировать систему приложения, чтобы можно было достичь баланса рабочей загрузки компьютерных систем клиента и сервера (**рис. 3.10**). Для этого в первую очередь следует реализовать сервер базы данных, который



**Рис. 3.10.** Сбалансированная нагрузка клиента и сервера

способен на любую работу по выполнению запросов путем прямого доступа к памяти для хранения данных. В сеть должны возвращаться только данные, необходимые для выполнения конкретной задачи (это часто самый "медленный" компонент архитектуры клиент/сервер).

Сервер базы данных может также централизованно реализовывать все правила целостности и защиты данных. При этом целостность данных обеспечивается независимо от целостности данных клиентов.

Реализация сбалансированной архитектуры требует дополнительной работы и планирования, но в результате достигаются высокая степень целостности данных и возможность манипулировать более сложными рабочими загрузками. Недостаток ее в том, что ни клиент, ни сервер не являются простыми реализациями.

### **3.5.3. Преимущества и перспективы системы клиент/сервер**

В настоящее время применение компьютерных приложений с архитектурой клиент/сервер – лучший способ использования вычислительных ресурсов. Эта архитектура дает следующие преимущества:

- рабочая загрузка естественным образом распределена на множестве компьютеров;
- пользователи могут легко совместно пользоваться данными;
- чувствительные к повреждениям данные можно надежно защитить в централизованном порядке;
- использование нескольких малых компьютеров, предназначенных для решения отдельных задач, улучшает показатель "стоимость / эффективность" по сравнению с применением одного мощного компьютера.

В то же время такая архитектура имеет некоторые неудобства, которые приходится учитывать при проектировании и администрировании:

- проектировщик должен определить, какие задачи следует выполнять клиенту, а какие – серверу;
- проектировщик должен выбрать для решения каждого типа задач подходящее аппаратное обеспечение;
- переходы к новым версиям программного обеспечения клиента должны контролироваться очень строго;
- изменения конструкции базы данных сервера могут повлиять на всех клиентов;
- сетевая топология часто очень сложна;
- контроль производительности и необходимые регулировки могут оказаться немного сложнее, чем при централизованной архитектуре.

### **3.6. Общие понятия реляционного подхода к организации данных**

Современная технология реализации баз данных, как правило, основана на реляционной модели, которая, в свою очередь, является некоторой абстрактной теорией данных. Принципы реляционной модели были изначально заложены в 1969 году доктором Е.Ф.Коддом, в то время исследователем, работавшим в

корпорации IBM. В конце 1968 года Кодд, математик по образованию, впервые осознал, что математические дисциплины можно использовать, чтобы привести в область управления базами данных строгие принципы и точность. Именно таких качеств и не доставало этой области в то время. С того времени эти идеи стали общепринятыми и оказали весьма существенное влияние на все аспекты технологии баз данных. Первые реляционные программные продукты начали появляться в конце 70-х – начале 80-х годов. С тех пор почти все созданные коммерческие СУБД основываются на реляционном подходе.

Что же понимается под реляционной системой? В самом широком смысле реляционная система – это система, основанная на следующих принципах:

- данные воспринимаются пользователем как таблицы;
- в распоряжении пользователя имеются операторы манипулирования данными (например, для выборки данных), которые генерируют новые таблицы из старых.

Причина, по которой такие системы называют реляционными, заключается в том, что английский термин «relation» (реляция, отношение), по существу, просто математическое название для таблицы. Поэтому на практике в большинстве случаев термины «отношение» и «таблица» принято считать синонимами. Определение «реляционные системы» требует, чтобы база данных *воспринималась пользователем* только в виде таблиц. Таблицы в реляционной системе являются *логическими*, а не физическими структурами. В действительности на физическом уровне система баз данных может использовать любую или все применяемые структуры памяти – последовательные файлы, индексирование, цепочки указателей, сжатие хранимых данных и т.п. – лишь бы обеспечивалась возможность отображать эти структуры в таблицы на логическом уровне. Другими словами, таблицы представляют *абстракцию* способа физического хранения данных, в которой множество деталей на уровне памяти – размещение, последовательность хранимых записей, кодировка хранимых данных, хранимые структуры доступа, такие как индексы, и т.д. – *скрыто от пользователей*.

На **рис. 3.11** показан простой пример реляционной базы данных отделов и сотрудников.

Среди операторов манипулирования данными есть как минимум следующие:

- SELECT, предназначенный для извлечения определенных строк из таблицы;
- PROJECT, предназначенный для извлечения определенных столбцов из таблицы;
- JOIN, предназначенный для соединения двух таблиц на основе общих значений в общих столбцах.

Некоторые примеры использования операций SELECT, PROJECT И JOIN показаны на **рис. 3.12**.

Необходимо отметить важный момент, очевидный из **рис. 3.12**: результат каждой из трех операций – это еще одна таблица. Это реляционное свойство **замкнутости**. Оно имеет очень большое значение главным образом потому, что *результатом операции является объект того же рода, что и объект, над которым производилась операция* – а именно таблица. Это значит, что над

Отдел		
Номер отдела	Наименование	Бюджет
1	Технический	200
2	Научный	250
3	Проектный	300

Сотрудники			
Номер сотрудника	Имя сотрудника	Номер отдела	Зарплата
14252	Королев	1	800
18453	Луценко	1	820
15265	Городинский	2	750
48452	Борисов	2	800

*Рис. 3.11. Фрагмент базы данных отделов и сотрудников*

SELECT (RESTRICT): Выборка строк из таблицы "Отделы", где бюджет более 230  
 Результат:

Номер отдела	Наименование	Бюджет
2	Научный	250
3	Проектный	300

PROJECT: Извлечение столбцов "Наименование" и "Бюджет" из таблицы "Отделы"  
 Результат:

Наименование	Бюджет
Технический	200
Научный	250
Проектный	300

JOIN: соединение таблиц "Отделы" и "Сотрудники" на основе столбца "Номер отдела"  
 Результат:

Номер отдела	Наименование	Бюджет	Номер сотрудника	Имя сотрудника	Зарплата
1	Технический	200	14252	Королев	800
1	Технический	200	18453	Луценко	820
2	Научный	250	15265	Городинский	750
2	Научный	250	48452	Борисов	800

*Рис. 3.12. Примеры операций SELECT, PROJECT и JOIN*

результатом операции можно вновь проделать какую-либо другую операцию. Например, можно выбрать столбцы из соединенной таблицы или соединить два результата операции SELECT и т.д. Другими словами, можно использовать вложенные выражения, т.е. выражения, в которых операнды<sup>1</sup> представлены выражениями, а не простыми именами таблиц.

Другой момент, который также проиллюстрирован на **рис. 3.12**, заключается в том, что операции применяются *сразу ко всему множеству строк*, а не к отдельной строке за один раз, т.е. операндами и результатами являются не отдельные строки, а целые таблицы, которые содержат *множество* строк. Такая возможность обработки множества – главная отличительная характеристика реляционных систем; нереляционные системы, как правило, за одно обращение выполняют обработку на уровне одной строки или записи.

### 3.6.1. Базовые понятия реляционных баз данных

Основными понятиями реляционных баз данных являются:

- тип данных;
- домен;
- атрибут;
- кортеж;
- отношение;
- первичный ключ.

Покажем смысл этих понятий на примере отношения СОТРУДНИКИ, содержащего информацию о сотрудниках некоторой организации (**рис. 3.13**).

#### Тип данных

Понятие тип данных в реляционной модели данных полностью адекватно понятию типа данных в языках программирования. Обычно в современных реляционных БД допускается хранение символьных, числовых данных, битовых строк, специализированных числовых данных (таких как «деньги»), а также специальных данных (дата, время, временной интервал). Достаточно активно развивается подход к расширению возможностей реляционных систем абстрактными типами данных. В нашем примере мы имеем дело с данными четырех типов: целые числа, строковые переменные, «деньги» и переменные типа даты.

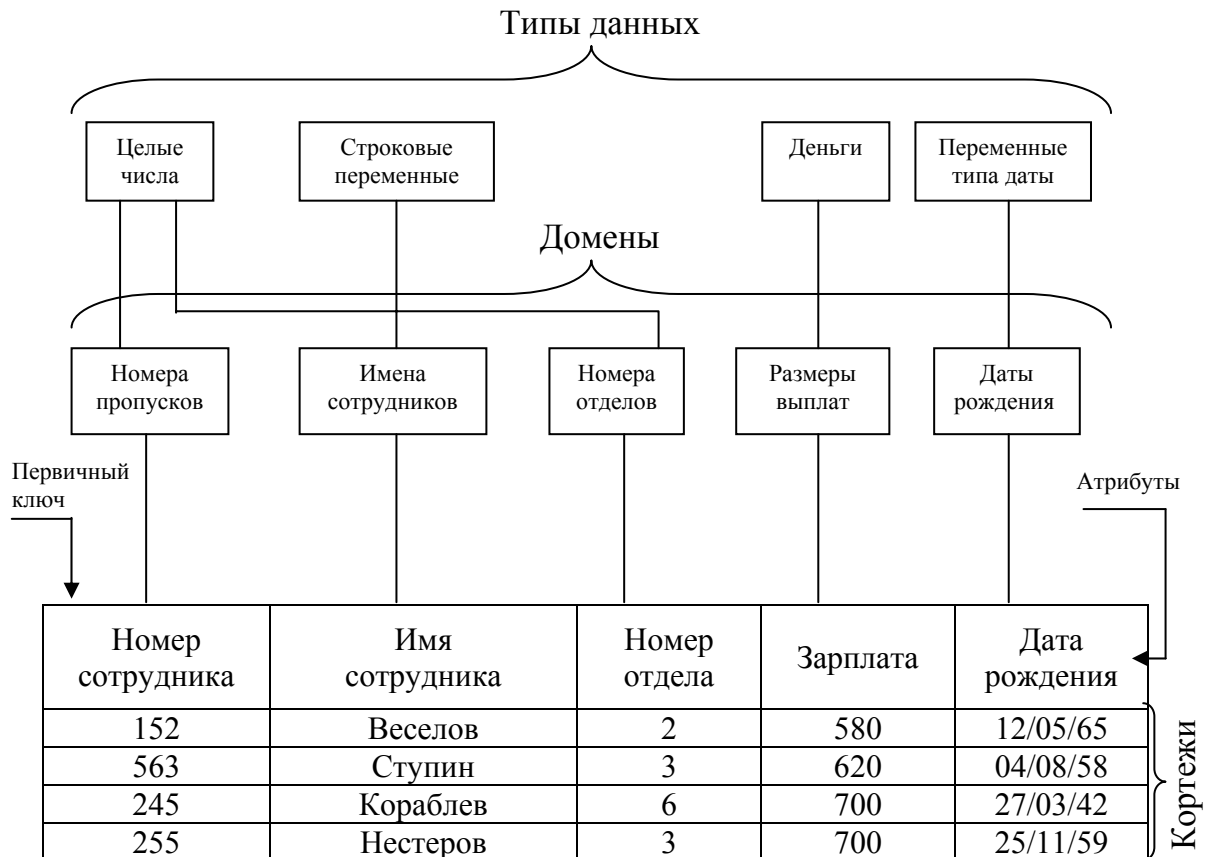
#### Домен

Понятие домена более специфично для баз данных, хотя и имеет некоторые аналогии с подтипами в некоторых языках программирования. В самом общем виде *домен* определяется заданием некоторого базового типа данных, к которому относятся элементы домена, и произвольного логического выражения, применяемого к элементу типа данных. Если вычисление этого логического вы-

---

<sup>1</sup> Операнд – часть выражения, над которой производится операция, аргумент операции.

ражения дает результат "истина", то элемент данных является элементом домена.



**Рис. 3.13.** К пояснению основных терминов реляционных баз данных

Наиболее правильной трактовкой понятия домена является понимание домена как допустимого потенциального множества значений данного типа. Например, домен "Имена сотрудников" в нашем примере определен на базовом типе строковых переменных, но в число его значений могут входить только те строки, которые могут изображать имя (в частности, такие строки не могут начинаться с мягкого знака).

Следует отметить также смысловую (семантическую) нагрузку понятия домена: данные считаются сравнимыми только в том случае, когда они относятся к *одному домену*. В этом и состоит основное назначение доменов: домены ограничивают сравнения. Так, в нашем примере значения доменов "Номера сотрудников" и "Номера отделов" относятся к типу целых чисел, но не являются сравнимыми.

### Кортеж

**Кортеж**, соответствующий данной схеме отношения, – это множество пар {имя атрибута; значение}, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения. Другими словами, кортеж – это



строка отношения (таблицы) реляционной модели данных, представляющая собой последовательность значений атрибутов.

### **Схема отношения, схема базы данных**

**Схема отношения** – это именованное множество пар {имя атрибута; имя домена (или типа)}. Степень, или «арность», схемы отношения характеризует мощность этого множества. Степень отношения СОТРУДНИКИ равна пяти, то есть оно является 5-арным. Если все атрибуты одного отношения определены на разных доменах, разумно использовать для именования атрибутов имена соответствующих доменов (не забывая, конечно, о том, что это является всего лишь удобным способом именования и не устраняет различия между понятиями домена и атрибута).

Схема БД (в структурном смысле) – это набор именованных схем отношений.

### **Отношение**

**Отношение** – это множество кортежей (строк таблицы), соответствующих одной схеме отношения. Иногда схему отношения называют *заголовком отношения*, а отношение как набор кортежей – *телом отношения*. На самом деле, понятие схемы отношения ближе всего к понятию структурного типа данных в языках программирования. Было бы вполне логично разрешать отдельно определять схему отношения, а затем одно или несколько отношений с данной схемой. Однако в реляционных базах данных это не принято. Имя схемы отношения в таких базах данных всегда совпадает с именем соответствующего отношения-экземпляра. В классических реляционных базах данных после определения схемы базы данных изменяются только отношения-экземпляры. В них могут появляться новые и удаляться или модифицироваться существующие кортежи. Однако во многих реализациях допускается и изменение схемы базы данных: определение новых и изменение существующих схем отношения. Это принято называть эволюцией схемы базы данных.

Обычное житейское представление отношения – таблица, заголовком которой является схема отношения, столбцами – атрибуты отношения, а строками – кортежи отношения-экземпляра; в этом случае имена атрибутов именуют столбцы этой таблицы. Поэтому иногда говорят «столбец таблицы», имея в виду «атрибут отношения». Этой терминологии придерживаются в большинстве коммерческих реляционных СУБД. В **табл. 3.1** представлен обзор некоторых реляционных терминов и даны неформальные эквиваленты (соответствия) этим терминам.

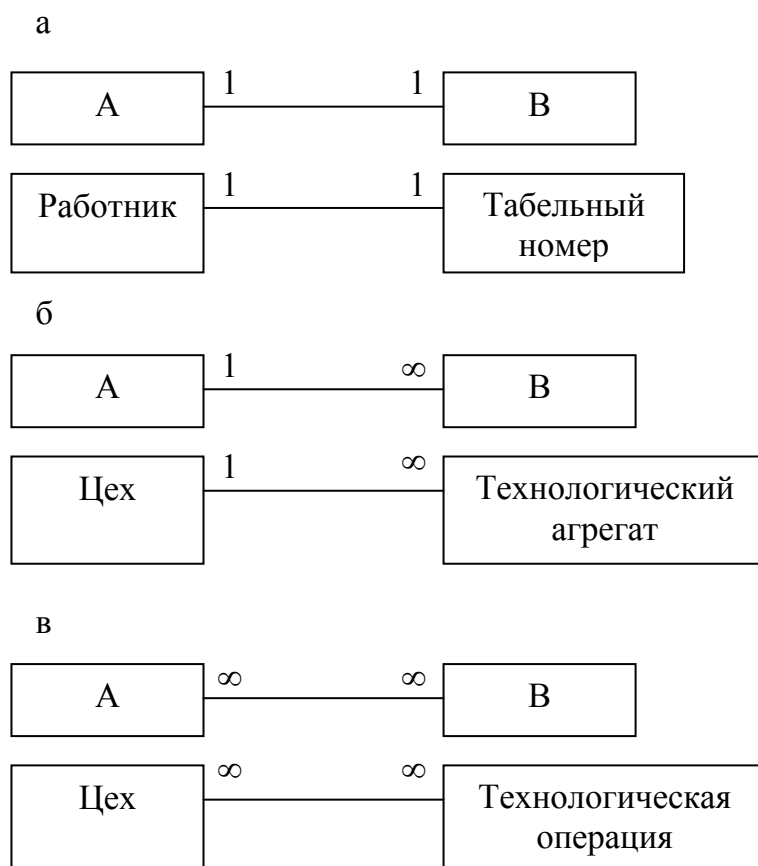
Как видно из **табл. 3.1**, основные структурные понятия реляционной модели данных (если не считать понятия домена) имеют очень простую интуитивную интерпретацию, хотя в теории реляционных БД все они определяются абсолютно формально и точно.

Таблица 3.1

**Неформальные определения реляционных объектов данных**

Формальный реляционный термин	Неформальный эквивалент
Отношение	Таблица
Кортеж	Строка или запись таблицы
Кардинальное число	Количество строк
Атрибут	Столбец или поле
Степень (арность)	Количество столбцов
Первичный ключ	Уникальный идентификатор
Домен	Общая совокупность допустимых значений данных в столбце

Учитывая вышесказанное, можно определить реляционную базу данных как набор отношений, связанных между собой. Связь в данном случае – это ассоциирование двух или более отношений. База данных, не имеющая связей между отношениями, имеет очень простую структуру и в полной мере реляционной называться не может. Одно из основных требований к организации реляционной базы данных – это обеспечение возможности поиска одних кортежей по значениям других, для чего необходимо установить между ними связи. Существуют следующие основные виды связей:



**Рис. 3.14.** Примеры различных видов связей: а – один к одному; б – один ко многим; в – многие ко многим

- один к одному;
- один ко многим;
- многие ко многим.

Связь «один к одному» предполагает, что в каждый момент времени каждому элементу (кортежу) отношения А соответствует 0 или 1 элементов (кортежей) отношения В. Например, за каждым работником предприятия закреплен один и только один табельный номер (**рис. 3.14, а**).

Связь «один ко многим» состоит в том, что в каждый момент времени каждому элементу (кортежу) отношения А соответствует несколько элементов (кортежей) отношения В (**рис. 3.14, б**). В качестве примера можно привести связь между отношениями «Цех» и «Технологический агрегат»: в одном цехе расположено несколько технологических агрегатов, каждый агрегат установлен только в одном цехе.

Связь «многие ко многим» предполагает, что в каждый момент времени множеству элементов отношения А соответствует множество элементов отношения В. К сожалению, этот тип связи в реляционных СУБД непосредственно не поддерживается. На **рис. 3.14, в** приведен пример такого типа связи для отношений «Изделие» и «Технологическая операция»: каждое изделие проходит множество технологических операций, в то же время одна технологическая операция выполняется для множества изделий.

### **3.6.2. Общая характеристика реляционной модели данных**

Когда в предыдущих разделах мы говорили об основных понятиях реляционных баз данных, мы не опирались на какую-либо конкретную реализацию. Эти рассуждения в равной степени относились к любой системе, при построении которой использовался реляционный подход к организации данных. Другими словами, мы использовали понятия так называемой реляционной модели данных. Модель данных описывает некоторый набор родовых понятий и признаков, которыми должны обладать все конкретные СУБД и управляемые ими базы данных, если они основываются на этой модели. Наличие модели данных позволяет сравнивать конкретные реализации, используя один общий язык.

Наиболее распространенная трактовка реляционной модели данных принадлежит К.Дж.Дейту, известному американскому специалисту в области реляционных баз данных: реляционная модель состоит из трех частей, описывающих разные аспекты реляционного подхода:

- структурной части;
- манипуляционной части;
- целостной части.

В **структурной** части модели фиксируется, что единственной структурой данных, используемой в реляционных БД, является нормализованное отношение. Процедуру нормализации данных рассмотрим позже.

В **манипуляционной** части модели утверждаются два фундаментальных механизма манипулирования реляционными БД – реляционная алгебра и реляционное исчисление. Первый механизм базируется в основном на классической теории множеств (с некоторыми уточнениями), а второй – на классическом ло-

гическом аппарате исчисления предикатов первого порядка. Далее мы рассмотрим эти механизмы более подробно, а пока лишь заметим, что основной функцией манипуляционной части реляционной модели является обеспечение меры реляционности любого конкретного языка реляционных БД: язык называется реляционным, если он обладает не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционное исчисление.

В **целостной** части реляционной модели данных фиксируются два базовых требования целостности, которые должны поддерживаться в любой реляционной СУБД. Первое требование называется требованием **целостности сущностей**. Объекту или сущности реального мира в реляционных БД соответствуют кортежи отношений. Конкретно требование состоит в том, что любой кортеж любого отношения отличим от любого другого кортежа этого отношения, т.е., другими словами, любое отношение должно обладать первичным ключом – уникальным идентификатором. Как мы видели в предыдущем разделе, это требование автоматически удовлетворяется, если в системе не нарушаются базовые свойства отношений.

Второе требование называется требованием **целостности по ссылкам** и является несколько более сложным. Очевидно, что при соблюдении нормализованности отношений (набора условий для формирования реляционной модели данных) сложные сущности реального мира представляются в реляционной БД в виде нескольких кортежей нескольких отношений. Например, представим, что нам требуется показать в реляционной базе данных сущность ОТДЕЛ с атрибутами ОТД\_НОМЕР (номер отдела), ОТД\_КОЛ (количество сотрудников) и ОТД\_СОТР (набор сотрудников отдела). Для каждого сотрудника нужно хранить СОТР\_НОМЕР (номер сотрудника), СОТР\_ИМЯ (имя сотрудника) и СОТР\_ЗАРП (заработная плата сотрудника). Как мы вскоре увидим, при правильном проектировании соответствующей БД в ней появятся два отношения:

ОТДЕЛЫ (ОТД\_НОМЕР, ОТД\_КОЛ) (первичный ключ – ОТД\_НОМЕР)  
 СОТРУДНИКИ (СОТР\_НОМЕР, СОТР\_ИМЯ, СОТР\_ЗАРП,  
 СОТР\_ОТД\_НОМ) (первичный ключ – СОТР\_НОМЕР).

Как видно, атрибут СОТР\_ОТД\_НОМ появляется в отношении СОТРУДНИКИ не потому, что номер отдела является собственным свойством сотрудника, а лишь для того, чтобы иметь возможность восстановить при необходимости полную сущность ОТДЕЛ. Значение атрибута СОТР\_ОТД\_НОМ в любом кортеже отношения СОТРУДНИКИ должно соответствовать значению атрибута ОТД\_НОМ в некотором кортеже отношения ОТДЕЛЫ. Атрибут такого рода называется внешним ключом, поскольку его значения однозначно характеризуют сущности, представленные кортежами некоторого другого отношения (т.е. задают значения их первичного ключа). Говорят, что отношение, в котором определен внешний ключ, ссылается на соответствующее отношение, в котором такой же атрибут является первичным ключом.

*Требование целостности по ссылкам, или требование внешнего ключа, состоит в том, что для каждого значения внешнего ключа, появляющегося в ссылающемся отношении, в таблице, на которую ведет ссылка, должен найтись*

*кортеж с таким же значением первичного ключа либо значение внешнего ключа должно быть неопределенным, то есть ни на что не указывать.* Для нашего примера это означает, что если для сотрудника указан номер отдела, то этот отдел должен существовать.

Ограничения целостности сущности и по ссылкам должны поддерживаться СУБД. Для соблюдения целостности сущности достаточно гарантировать отсутствие в любом отношении кортежей с одним и тем же значением первичного ключа. С целостностью по ссылкам дела обстоят несколько более сложно.

Понятно, что при обновлении ссылающегося отношения (вставке новых кортежей или модификации значения внешнего ключа в существующих кортежах) достаточно следить за тем, чтобы не появлялись некорректные значения внешнего ключа. Но как быть при удалении кортежа из отношения, на которое ведет ссылка? Здесь существуют три подхода, каждый из которых поддерживает целостность по ссылкам.

*Первый подход* заключается в том, что запрещается производить удаление кортежа, на который существуют ссылки (т.е. сначала нужно либо удалить ссылающиеся кортежи, либо соответствующим образом изменить значения их внешнего ключа). При *втором подходе* при удалении кортежа, на который имеются ссылки, во всех ссылающихся кортежах значение внешнего ключа автоматически становится неопределенным. Наконец, *третий подход* (каскадное удаление) состоит в том, что при удалении кортежа из отношения, на которое ведет ссылка, из ссылающегося отношения автоматически удаляются все ссылающиеся кортежи. В развитых реляционных СУБД обычно можно выбрать способ поддержания целостности по ссылкам для каждой отдельной ситуации определения внешнего ключа. Конечно, для принятия такого решения необходимо анализировать требования конкретной прикладной области.

### **3.6.3. Манипулирование данными с помощью языка запросов SQL**

Ранее уже было отмечено, что языковые средства СУБД, обеспечивающие обработку (представление и редактирование) данных, являются ее важнейшей компонентой. Развитие реляционной модели данных привело к появлению реляционных языков, наиболее распространенным из которых является SQL (Structured Query Language, язык структурированных запросов). На сегодняшний день язык SQL является де-факто стандартом реляционного языка баз данных и используется в большинстве коммерческих систем.

SQL был разработан в компании IBM Research в начале 1970-х годов, с того времени диалекты языка SQL стали Американским национальным стандартом (ANSI), Международным стандартом (ISO), стандартом системы UNIX, стандартом корпорации IBM, описывающим "общий интерфейс базы данных". Язык SQL используется для выполнения операций над таблицами (создание, удаление, изменение структуры) и над данными (выборка, добавление, изменение, удаление), а также некоторых сопутствующих операций (сортировка, группировка данных и др.). Как уже было отмечено, SQL является непроцедурным языком, что считается одним из основных его преимуществ, и не содержит имеющихся в обычных языках программирования операторов управления, организации под-

программ, ввода-вывода и т.п. Формулирование запроса к базе данных на проведение определенных операций над данными производится на естественном для пользователя языке (английском). Важно подчеркнуть, что эти возможности опираются только на *логическую* структуру базы данных, а не на ее физическую структуру, что согласуется с требованием реляционной модели. Основным требованием от пользователя является умение правильно сформулировать запрос, не заботясь о том, *как* происходит его выполнение. SQL автономно не используется, а обычно является встроенным языком программирования многих СУБД (Access, Paradox, Visual FoxPro и др.) или современных версий процедурных языков программирования (Pascal, C++ и др.).

В силу ограниченности объема данной книги отметим только некоторые функции языка по организации запросов, иллюстрирующие структуру операторов SQL. Для детального освоения языка SQL читателям можно рекомендовать специальную литературу [15]. Все запросы на получение практически любого количества данных из одной или нескольких таблиц выполняются с помощью единственного предложения SELECT. В общем случае результатом реализации предложения SELECT является другая таблица. К этой новой (рабочей) таблице может быть снова применена операция SELECT и т.д., т.е. такие операции могут быть вложены друг в друга. В упрощенном виде оператор SELECT имеет следующий формат:

```
SELECT [ALL|DISTINCT] <список данных>
      FROM <список таблиц>
      [WHERE <условие выборки>]
      [GROUP BY <имя столбца> [, <список таблиц>]...]
```

Смысл синтаксических конструкций оператора SELECT можно выразить следующим образом:

- SELECT (выбрать) – данные из указанных столбцов и (если необходимо) выполнить перед выводом их преобразование в соответствии с указанными выражениями и (или) функциями;
- FROM (из) – перечисленных таблиц, в которых расположены эти столбцы;
- WHERE (где) – строки из указанных таблиц должны удовлетворять указанному перечню условий отбора строк;
- GROUP BY (группируя по) – указанному перечню столбцов с тем, чтобы получить для каждой группы единственное агрегированное значение, используя во фразе SELECT SQL-функции SUM (сумма), COUNT (количество), MIN (минимальное значение), MAX (максимальное значение) или AVG (среднее значение).

Кроме традиционных операторов сравнения (= | <> | < | <= | > | >=), в WHERE фразе используются условия BETWEEN (между), LIKE (похоже на), IN (принадлежит), IS NULL (не определено) и EXISTS (существует), которые могут предваряться оператором NOT (не). Критерий отбора строк формируется из одного или нескольких условий, соединенных логическими операторами:

- AND (и) – когда должны удовлетворяться оба разделяемых с помощью AND условия;
- OR (или) – когда должно удовлетворяться одно из разделяемых с помощью OR условий;

- AND NOT (и не) – когда должно удовлетворяться первое условие и не должно второе;
- OR NOT (или не) – когда или должно удовлетворяться первое условие, или не должно удовлетворяться второе, причем существует приоритет AND над OR (сначала выполняются все операции AND и только после этого операции OR). Для получения желаемого результата WHERE условия должны быть введены в правильном порядке, который можно организовать введением скобок.

В качестве иллюстрации возможностей SQL рассмотрим примеры запросов, использующие указанные конструкции.

Запрос "Получить данные о видах загружаемых материалов в доменную печь" на языке SQL выглядит следующим образом:

```
SELECT Материалы FROM Доменная_печь
```

Пример более сложного запроса, предполагающего формирование специальных условий на результат отбора с помощью фразы WHERE: "Получить данные химического анализа о содержании кремния и углерода в чугуна, взятом на доменной печи №1 при выпуске №123212".

```
SELECT Si_чуг, C_чуг  
FROM Доменная_печь  
WHERE Д_печь='1' AND N_выпуска='123212'
```

Запрос, иллюстрирующий использование встроенных статистических функций (функций множества): "Получить общее количество выпусков на каждой из доменных печей за 01.10.2000".

```
SELECT COUNT(N_выпуска)  
FROM Доменная_печь  
WHERE Дата_выпуска=#01.10.2000#
```

### 3.7. Основы проектирования реляционных баз данных

#### 3.7.1. Основные требования при проектировании БД

Процесс разработки структуры (схемы) базы данных (БД) в соответствии с требованиями пользователей называется *проектированием базы данных*.

Большинство известных методов проектирования баз данных появилось в процессе разработки все более сложных информационных систем (ИС), которые должны были реализовывать потребности не одного пользователя, но больших групп и коллективов. Одна такая интегрированная БД создавалась для решения многих задач, каждая из которых использовала только "свою" часть данных, обычно пересекающуюся с частями, используемыми в других задачах. Поэтому главными методами проектирования стали методы исключения избыточности в данных и формирования *эффективной структуры БД*.

Было сформулировано принципиальное требование отделения прикладных программ от интегрированных данных. Этот принцип направлен на отчуждение данных в качестве ресурса предприятия, важен тем, что консервативные по характеру данные отделялись от прикладных программ, которые могли часто подвергаться изменениям.

Другой важной проблемой проектирования БД явилось обеспечение нужных *эксплуатационных параметров*, таких как объем внешней памяти или время выполнения различных операций. Известны и другие требования. Например, информация не должна потеряться не только из-за отказов оборудования, но и вследствие ошибки пользователя. Это отличается от того положения, при котором тот, кто решает некую задачу, сам и отвечает за сохранность данных для этой задачи.

Сформировалось понимание интегрированной БД как общего информационного ресурса предприятия. Хранимые данные стали аналогичны большому компьютеру, который одновременно используется многими пользователями с различными целями и должен быть все время работоспособен.

### 3.7.2. Основы классической методологии проектирования БД

Процесс проектирования БД начинается с установления *концептуальных требований* ряда пользователей (рис. 3.15). Концептуальные требования могут определяться и для некоторых приложений, которые в ближайшее время реализовываться не будут. Эти требования отдельных пользователей интегрируются в едином "обобщенном представлении". Последнее называют *концептуальной моделью* (схемой). Концептуальная модель представляет объекты и их взаимо-

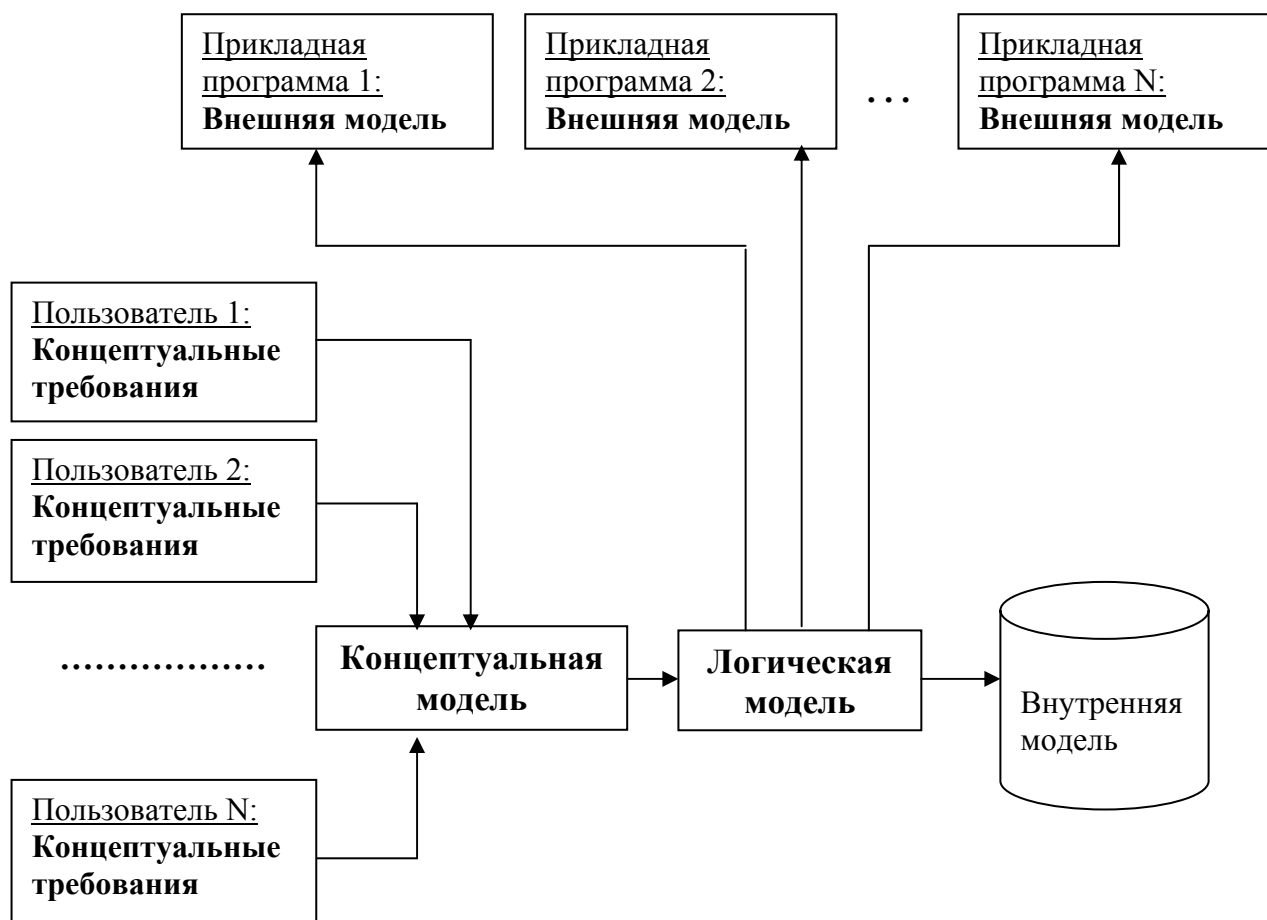


Рис. 3.15. Модели представления данных



связи без указания способов их физического хранения, типа используемого компьютера и версии конкретной СУБД. Таким образом, концептуальная модель является, по существу, моделью предметной области.

Концептуальная модель транслируется затем в модель данных, совместимую с выбранной СУБД. Возможно, что отраженные в концептуальной модели взаимосвязи между объектами окажутся впоследствии нереализуемыми средствами выбранной СУБД. Это потребует изменения концептуальной модели. Вариант концептуальной модели, которая может быть обеспечена СУБД, называется *логической моделью*. Важно помнить, что логическая модель данных обусловлена требованиями и характеристиками СУБД, поэтому при замене СУБД она также изменится. Пользователям выделяются подмножества этой логической модели, называемые внешними моделями (подсхемами), отражающие их требования и представления.

Логическая модель отображается в физическую память (например, диски). Физическая модель, специфицирующая размещение данных и методы доступа к ним, называется *внутренней моделью*.

Двумя основными результатами процесса проектирования баз данных являются *полная структура* базы данных (т.е. концептуальная, логическая и внутренняя модели) и *руководство для прикладных программистов*, основанные на внешних моделях и требованиях к обработке данных. В целом эти результаты могут рассматриваться как спецификации для реализации баз данных.

Одним из базовых понятий методологии проектирования является понятие жизненного цикла (ЖЦ) системы баз данных. Жизненный цикл системы баз данных делится на две отдельные фазы: фазу анализа и проектирования и фазу эксплуатации. В течение первой фазы происходит сбор требований пользователей и проектирование базы данных, в течение второй – компьютерная ее реализация, внедрение и использование. С точки зрения проектировщика и пользователя, можно детализировать содержание работ, выполняемых в течение этих фаз ЖЦ системы баз данных. В этом смысле указанные фазы включают следующие этапы:

#### **Фаза анализа и проектирования**

1. Формулирование и анализ требований.
2. Концептуальное проектирование.
3. Проектирование реализации.
4. Физическое проектирование.

#### **Фаза реализации и функционирования базы данных**

1. Реализация базы данных.
2. Анализ функционирования и поддержка.
3. Модификация.

#### *Фаза анализа и проектирования*

**Формулирование и анализ требований** являются наиболее трудным и длительным по времени этапом проектирования. Однако он является наиболее важным, т.к. на нем основано большинство последующих проектных решений. Основной задачей является сбор требований, предъявляемых к содержанию и

процессу обработки данных всеми известными и потенциальными пользователями базы данных. Анализ требований обеспечивает согласованность целей пользователей, а также согласованность их представлений об информационном потоке организации.

**Концептуальное проектирование** имеет целью построение независимой от СУБД информационной структуры путем объединения информационных требований пользователей. Результатом концептуального проектирования является концептуальная модель.

**Проектирование реализации** (проектирование логической схемы БД) состоит из двух компонентов: проектирование базы данных и проектирование программ. Структурой базы данных (логической моделью), полученной в результате проектирования реализации, является СУБД-ориентированное описание данных, обычно выраженное в терминах языка описания данных. Проектирование программного обеспечения имеет целью создание структурированных программ, использующих базовый язык программирования и язык манипулирования данными СУБД. Результатом являются функциональные спецификации программных модулей и набор возможных запросов к базе данных.

**Физическое проектирование**, так же как и проектирование реализации, состоит из двух компонентов: выбор физической структуры базы данных и уточнение спецификаций программных модулей, определенных на предыдущем этапе. Результатом физического проектирования являются полностью готовая к внедрению структура базы данных и набор реализуемых алгоритмов по ее использованию.

#### *Фаза реализации и функционирования базы данных*

**Реализация базы данных** подразумевает создание базы данных и прикладных программ на основе результатов трех главных этапов проектирования базы данных, а также загрузку базы данных. Имеющиеся данные необходимо преобразовывать из существующей формы представления логической и физической структуры в новую форму, соответствующую результатам проектирования базы данных. Разработка прикладных программ связана с выбором базового языка программирования и логической структурой БД. Целью этого этапа является создание надежных и эффективных программ доступа к базе данных, удовлетворяющих требованиям пользователей к обработке данных.

**Анализ функционирования и поддержка** используются для сбора и статистической обработки данных о функционировании системы. Эта информация позволяет выявить степень обоснованности требований пользователей, а также «узкие места» в процессе эксплуатации с целью пересмотра системы в будущем. Поддержка базы данных должна обеспечивать ее целостность и эффективное восстановление после сбоев.

**Модификация** предусматривает внесение в реализованный проект изменений, возникающих вследствие появления новых требований, анализа функционирования системы или анализа мнений пользователей о работе системы. Целью этого этапа является оптимизация функционирования существующей системы

путем реорганизации (корректировки логической или физической структуры БД) базы данных и/или внесения изменений в программное обеспечение.

Рассмотрим подробнее основные этапы проектирования базы данных.

### **3.7.3. Основные этапы проектирования базы данных**

Есть все основания называть рассматриваемую методологию проектирования классической: для указанных методов разработаны полные, целостные методические системы, для большинства методов предложены формализованные модели: эти модели – или, по крайней мере, их итоговые выразительные возможности – нашли реальное применение в практике проектирования. Использовалась дисциплина т.н. структурного анализа при проектном подходе "сверху вниз". Структурность связывается с использованием иерархических структур для детализации данных и функций и соответствующих достаточно "жестких" проектных процедур. Проектная схема получила название "каскадной" (рис. 3.16).

#### **Этап 1. Формулирование и анализ требований**

На этом этапе устанавливаются цели организации, определяются специфические требования к базе данных, вытекающие из этих целей или сформулированные непосредственно управляющим персоналом организации. Эти требования документируются в форме, доступной как конечному пользователю, так и проектировщику базы данных. Обычно при этом используется методика интервьюирования персонала различных уровней управления и ведущих специалистов организации, участвующих в процессах производства, обслуживания и обработки данных. В результате собеседований определяются информационные потоки, отображающие указанные процессы и их взаимодействие, а также вырабатывается однозначное представление о семантике информационной модели. Специфические цели и требования к базе данных необходимо определить на самом высоком уровне организации. Собранные и документированные требования должны включать ограничения, обуславливающие безопасность, надежность, уровень достигнутой технологии, а также политические и бюрократические ограничения. Должна быть также оценена политика организации в отношении персонала, управленческой деятельности и перспектив роста организации. Документирование может быть выполнено вручную с помощью языков представления требований или с помощью средств автоматизации. Может оказаться полезным на этапе документирования отделить требования, зависящие от процесса обработки данных (т. е. связанные со специфическими приложениями), от требований, не зависящих от процесса обработки.

#### **Этап 2. Концептуальное проектирование**

Этап концептуального проектирования связан с описанием и синтезом разнообразных концептуальных требований пользователей в первоначальный проект баз данных. Результатом этого этапа является высокоуровневое представление информационных требований, например, с помощью диаграмм "сущность-связь" (*Entity-Relationship Diagrams, ER-диаграммы*), которые мы рассмотрим в дальнейшем. Основу ER-диаграммы составляет набор сущностей ("сущность" в данном случае – синоним понятия "объект"), который представля-

ет или моделирует определенную совокупность сведений, специфицированную в требованиях. Сущности могут быть описаны атрибутами, позволяющими детализировать свойства сущности. Один или несколько атрибутов могут слу-

Общие информа-  
ционные требо-  
вания



Рис. 3.16. Основные этапы проектирования баз данных

жить идентификатором (*ключевой атрибут*) для обозначения отдельных экземпляров сущности. Связи между сущностями отображают функциональные аспекты информации, представленной сущностями.

Существует несколько подходов к построению диаграмм типа "сущность-связь". Общим для всех подходов является набор из четырех основных проектных решений или шагов:

- 1) определение сущностей;
- 2) определение атрибутов сущностей;
- 3) идентификация ключевых атрибутов сущностей;
- 4) определение связей между сущностями.

Выполнение этих шагов в том или ином порядке для каждого приложения называют моделированием частных представлений. При этом требования каждого пользователя анализируются и представляются в общей форме. Объекты и события, ассоциированные с представлением каждого пользователя, моделируются множеством сущностей, атрибутов и связей между сущностями.

Для построения концептуальной модели (иногда употребляют термин "*информационно-логическая (инфологическая) модель*") требуется произвести далее *интеграцию частных представлений*. Концептуальные требования отдельных пользователей, определенные в стандартной форме, такой как диаграммы "сущность-связь", сливаются в единое глобальное представление. При этом должны быть выявлены и ликвидированы противоречивые и избыточные данные. Важно помнить, что результат – первоначальная концептуальная модель, должна быть хотя бы частично независимой от процесса обработки, что позволит обеспечить гибкость процесса проектирования.

### Этап 3. Проектирование реализации

После построения первоначальной концептуальной модели следует ее уточнение и совершенствование. Главной целью этапа проектирования реализации является создание СУБД-ориентированной схемы (логической модели) с использованием в качестве исходных данных результатов концептуального проектирования и требований обработки конкретной СУБД.

Вначале анализируется содержание требований обработки данных. Формат локальных информационных структур, подлежащих обработке, соответствует формату первоначальной структуры, полученной на этапе концептуального проектирования. К локальным информационным структурам относят, в частности, **внешние** модели, рассмотренные ранее (см. **рис. 3.15**), а также **дatalogические** модели, представленные на языке описания данных конкретной СУБД. После того как представлены все виды обработки, первоначальная концептуальная модель может быть объединена со всеми внешними моделями в новую информационную структуру – логическую модель данных. Затем, используя знания, полученные в процессе пересмотра и объединения внешних моделей, учитывая связи обрабатываемых данных и характеристики типов записей, допускаемых СУБД, можно сформировать предварительные типы записей.

Построенная таким образом *логическая модель* данных может быть оценена количественно с помощью таких характеристик, как число обращений к логическим записям, объем обрабатываемых в каждом приложении данных, общий

объем хранимых данных. Эти оценки помогут приблизительно определить эффективность функционирования физической базы данных в терминах затрачиваемого на обработку времени и требуемой физической памяти. С учетом этих оценок логическая модель может быть усовершенствована с целью достижения большей эффективности. Основными показателями эффективности функционирования физической БД являются:

- **время ввода-вывода** – время, затрачиваемое на выборку данных с внешней памяти в оперативную, включая время передачи данных;
- **время доступа** – промежуток времени между выдачей команды, содержащей обращение к некоторым данным, и фактическим получением данных для обработки;
- **время отклика** – промежуток времени между вводом запроса к базе данных в компьютере и завершением обработки запроса с представлением результатов.

#### Этап 4. Физическое проектирование

Практика проектирования физической базы данных прошла несколько этапов своего развития, начиная от проектирования файлов и кончая проектированием физических структур интегрированных баз данных в рамках существующих ныне СУБД. Эта практика включает различные известные методы хранения структур, поисковые механизмы, сегментацию записей. Можно разделить основные проектные решения в физическом проектировании по меньшей мере на три категории.

- *Проектирование формата хранимых записей.* Сюда включаются все виды представления и сжатия данных в хранимых записях. Сюда же относится распределение элементов данных записи по различным участкам физической памяти в зависимости от размеров и характеристик их использования.
- *Анализ и проектирование кластеров.* Кластеризация включает размещение экземпляров записей в смежных участках физической памяти, распределение по различным устройствам внешней памяти, выбор размеров блоков с целью эффективной выборки.
- *Проектирование путей доступа.* Сюда включаются такие параметры, от которых в значительной степени зависит стоимость доступа при поиске и обновлении данных (например, логическое упорядочение записей, выбор указателей, методы доступа).

Отметим, что эти характеристики имеют важное значение и на этапе 3, то есть вопросы проектирования реализации и физического проектирования пересекаются.

#### 3.7.4. Обеспечение свойств БД в процессе проектирования

В процессе проектирования базы данных необходимо обеспечить ряд фундаментальных свойств:

- 1) целостность, согласованность, восстанавливаемость;
- 2) безопасность;
- 3) эффективность, рост, размер, эксплуатационные ограничения.

Это разделение свойств на группы отображает степени свободы проектировщика баз данных. Первая группа обычно отображает требования пользователей и состоит из "жестких" ограничений, в рамках которых должен работать проектировщик. Вторая и третья группы свойств обычно предоставляют проектировщику больше свободы в выборе средств. Обеспечение свойств третьей группы, как правило, осуществляется, когда уже найдено возможное решение, удовлетворяющее требованиям пользователей.

Напомним, что целостность – это правильность и точность данных в базе данных. База данных обладает свойством *целостности*, если она удовлетворяет некоторым определенным ограничениям значений данных и сохраняет это свойство при всех модификациях (замена, добавление или удаление) базы данных. В качестве примера рассмотрим ограничение целостности "зарплата служащего не может быть больше зарплаты его руководителя", определенное для базы данных служащих. Чтобы ввести это ограничение, необходимо сверить зарплату каждого служащего с зарплатой его руководителя. Если это ограничение не удовлетворяется для какого-либо служащего, то запись о данном служащем удаляется из базы данных. В дальнейшем это ограничение должно контролироваться всякий раз, когда в базу данных добавляется запись о служащем или модифицируется поле, в котором содержатся сведения о зарплате служащего. Из смысла этого ограничения следует, что оно не должно проверяться при операциях удаления записей из базы данных. Существуют такие ограничения, которые необходимо проверять либо во время операций обновления, либо удаления, либо в обоих случаях.

База данных обладает свойством *согласованности* по отношению к некоторой совокупности пользователей, если в любой момент времени база данных реагирует на их запросы одинаковым образом (т. е. все пользователи на заданный ими конкретный запрос получают одинаковый ответ). В качестве примера рассмотрим файл МАТЕРИАЛОВ, используемый совместно несколькими пользователями. Предположим, что работник склада корректирует сведения о ценах на материалы, а в то же самое время бухгалтер, используя базу данных, готовит бухгалтерский отчет. Если прикладная программа кладовщика не имеет возможности блокировать доступ к базе данных (т. е. исключить всех других пользователей) на все время выполнения корректировки сведений о ценах на материалы, бухгалтер может получить неправильный отчет. С другой стороны, если возможность блокирования доступна как кладовщику, так и бухгалтеру и используется надлежащим образом, то при этом оба достигнут согласованных результатов. Существует много способов поддержания согласованности базы данных.

*Восстанавливаемость* – это запроецированная возможность восстановления с помощью СУБД целостности базы данных после любого сбоя системы. Процесс проектирования такой возможности включает разработку систем проверки адекватности баз данных, предназначенных для того, чтобы избежать потерь данных. В случае сбоя системы основной задачей является восстановление в течение приемлемого времени существовавшего до сбоя состояния данных. Время восстановления может быть определено в требованиях пользователя как количественная характеристика. Однако такая характеристика должна включать четкое определение того, кем или чем будет инициироваться процесс восстанов-

ления: СУБД, операционной системой или оператором пульта управления. Это, конечно, будет зависеть от источника сбоя, которым могут являться СУБД, операционная система или технические средства.

Под **безопасностью** понимается защита данных от преднамеренного или непреднамеренного доступа, модификации или разрушения. Основной задачей поддержания безопасности базы данных является запрещение несанкционированного доступа к данным при минимальных затратах. Поэтому контроль доступа является наиболее важной проблемой реализации. Он может быть осуществлен посредством внешних моделей, которые позволяют отдельным пользователям иметь дело только с той частью логической модели данных, которая им необходима. Внутри такой модели можно усовершенствовать контроль доступа (только читать, писать, включить и т.д.) с помощью механизма блокирования. Большинство СУБД позволяют блокировать как всю базу данных целиком, так и отдельные записи и элементы данных. Выбор объектов блокирования может оказать значительное влияние на эффективность. Блокирование легче всего осуществить на уровне базы данных, однако это может лишить пользователей возможности доступа ко многим нечувствительным и срочно необходимым данным. Блокирование на уровне элементов данных представляет наибольшую гибкость, но значительно увеличивает накладные расходы при обработке элементов данных.

Наиболее приемлемым, по-видимому, будет решение, позволяющее без излишних накладных расходов на обработку добиться обеспечения безопасности на более низком уровне детализации данных. Поэтому четкое определение ограничений безопасности наряду с требованиями производительности (время отклика и т. д.) является решающим для достижения приемлемого компромисса при проектировании схемы базы данных.

При проектировании и реализации мер безопасности необходимо также принимать во внимание способы реализации и алгоритмы, используемые для обеспечения целостности и согласованности. Необходимо проектировать систему таким образом, чтобы поддержка целостности возлагалась на тех пользователей, которые имеют право на модификацию базы данных, в то время как для остальных пользователей это не является необходимым.

Реализация методов поддержания целостности и согласованности оказывает существенное влияние на **эффективность** СУБД. Разумный подход при их реализации может сократить связанные с этим накладные расходы в любой СУБД и, следовательно, повысить эффективность. В общем же случае поддержание целостности и согласованности увеличивает время отклика. Разумные методы и алгоритмы могут только минимизировать это увеличение. Если можно спланировать работу системы таким образом, что поддержание целостности и согласованности осуществляется в то время, когда система не занята обслуживанием пользователей, то рост времени отклика может быть снижен до нуля. Поэтому влияние этих компонентов на эффективность системы зависит исключительно от методов реализации. Чтобы оценить влияние поддержания целостности на эффективность, можно определить частоту и удельную стоимость процесса поддержания целостности в терминах вычислительных ресурсов (например, процессорного времени и времени ввода-вывода) и рассматривать программное



обеспечение целостности как одно из приложений наряду с приложениями конечных пользователей. Поддержание согласованности, восстанавливаемости и безопасности может быть оценено таким же способом.

### 3.8. Проектирование реляционных баз данных с использованием принципов нормализации

Как было показано в материале предыдущего параграфа, при проектировании базы данных решаются две основные проблемы.

- Каким образом отобразить объекты предметной области в абстрактные объекты модели данных, чтобы это отображение не противоречило семантике предметной области и было по возможности лучшим (эффективным, удобным и т.д.)? Часто эту проблему называют проблемой *логического* проектирования баз данных.
- Как обеспечить эффективность выполнения запросов к базе данных, т.е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создания каких дополнительных структур (например, индексов) потребовать и т.д.? Эту проблему называют проблемой *физического* проектирования баз данных.

В случае реляционных баз данных трудно представить какие-либо общие рецепты по части физического проектирования. Здесь слишком много зависит от типа используемой СУБД. Поэтому мы ограничимся вопросами логического проектирования реляционных баз данных, которые существенны при использовании любой реляционной СУБД.

Так что будем считать, что проблема проектирования реляционной базы данных состоит в обоснованном принятии решений о том,

- из каких отношений должна состоять БД,
- какие атрибуты должны быть у этих отношений.

Основная цель проектирования БД – это сокращение избыточности хранимых данных, а следовательно, экономия объема используемой памяти, уменьшение затрат на многократные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте. Так называемый "чистый" проект БД ("Каждый факт в одном месте") можно создать, используя методологию нормализации отношений, основанную на концепции нормальных форм.

Говорят, что отношение находится в некоторой нормальной форме, если оно удовлетворяет заданному набору условий. Метод нормализации предполагает последовательное приведение данного набора отношений к некоторой более желательной форме. Группировка атрибутов в отношениях должна быть рациональной, т.е. минимизирующей дублирование (избыточность) данных и упрощающей процедуры их обработки и обновления. Окончательная цель нормализации сводится к получению такого проекта базы данных, в котором каждый факт появляется лишь в одном месте, т.е. исключена избыточность информации. Это делается не столько с целью экономии памяти, сколько для исключения возможной противоречивости хранимых данных.

Исходной точкой является представление предметной области в виде одного или нескольких отношений. На каждом шаге проектирования производится формирование новых наборов схем отношений, обладающих лучшими свойствами при включении, модификации, удалении данных, чем все остальные возможные наборы отношений. Таким образом, процесс проектирования представляет собой процесс нормализации схем отношений.

**Нормализация схем отношений** – это формальный аппарат ограничений на формирование отношений (таблиц), который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на сопровождение (ввод, корректировку) базы данных.

Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений.

В теории реляционных баз данных выделяются следующие основные нормальные формы:

- первая нормальная форма (1НФ);
- вторая нормальная форма (2НФ);
- третья нормальная форма (3НФ).

Главные свойства нормальных форм заключаются в следующем:

- каждая следующая нормальная форма в некотором смысле лучше предыдущей;
- при переходе к следующей нормальной форме свойства предыдущих нормальных свойств сохраняются.

Заметим, что уровень нормализации данного отношения определяется *семантикой*, а не конкретными значениями данных в некоторый момент времени. Нельзя с первого взгляда на таблицу с данными для заданного отношения определить, находится ли оно, например, в 3НФ. Для этого также необходимо представить себе их смысл, то есть существующие между ними зависимости. Следует также отметить: даже зная о зависимостях данного отношения, нельзя *доказать* того, что оно находится в 3НФ. В таком случае можно лишь показать, что эти данные не нарушают никаких зависимостей, и высказать предположение о том, что эти данные не противоречат гипотезе о принадлежности отношения к 3НФ. Однако этот факт не гарантирует, что предложенная гипотеза верна.

На практике, как правило, исходное отношение приводят к 3НФ, при этом достигаются основные свойства реляционной модели данных. Поэтому ограничимся рассмотрением процедуры нормализации до 3НФ.

### 3.8.1. Первая нормальная форма

**Первая нормальная форма (1НФ, First Normal Form, 1NF)** – это основа реляционной системы. В 1НФ нельзя хранить несколько значений в одном столбце. В терминах теории баз данных каждое значение в столбце таблицы является *элементарным (атомарным)*, то есть состоящим из одного экземпляра.

Рассмотрим, например, систему обеспечения продажи товаров, в которой все коды заказанных продуктов могут храниться в записи одного заказа (**рис.3.17**). Такой подход к организации данных не удовлетворяет условию нор-

мализованной базы данных, поскольку в 1НФ не допускается ввод множества значений в одной записи (строке). Требуется создать схему, в которой в каждую запись заказа вводится только одно значение.

Номер заказа	Код товара	Стоимость заказа
1	1320, 1405, 1602, 1201, 1000	1174
2	2001, 1001, 2345	61
3	3021, 4000	186

*Рис. 3.17. Фрагмент системы обеспечения продажи товаров*

Чтобы реализовать систему обеспечения продажи, удовлетворяющую условию 1НФ, необходимо иметь заказ, представленный комбинацией от 1 до N записей, содержащих информацию о коде элементов заказа. Это решение усложняет выборку информации о содержании товаров в конкретном заказе, т.к. в этом случае требуется извлекать каждую ассоциированную с заказом запись и проверять, что извлекается именно нужная запись. Результат применения первого прохода при нормализации таблицы базы данных приведен на **рис. 3.18**.

Номер заказа	Код товара	Наименование товара	Оптовая цена
1	1320	Болты	12
1	1405	Молоток	20
1	1602	Гвозди	10
1	1201	Гайки	16
1	1000	Электропила	880
2	2001	Напильник	19
2	1001	Ключ гаечный	14
2	2345	Отвертка	15
3	3021	Стамеска	24
3	4000	Рубанок	124

*Рис. 3.18. Фрагмент организации данных, удовлетворяющих условию 1НФ*

### 3.8.2. Вторая нормальная форма

Первым требованием **второй нормальной формы** (2НФ, *Second Normal Form, 2NF*) является выполнение требований первой нормальной формы. В соответствии со вторым требованием каждая строка в таблице базы данных должна быть уникально идентифицирована.

В таблице заказов на **рис. 3.18** на первый взгляд все выглядит так, как будто структура соответствует этому правилу. Мы установили идентификатор заказа (номер заказа). Если его скомбинировать с идентификатором товара (код товара), то получится уникальный указатель на строку – составной ключ. Однако можно отметить неполную функциональную зависимость атрибута "Наименование товара" от этого ключа, поскольку семантика отношения для данного атрибута на самом деле включает следующие функциональные зависимости:

*Номер заказа, Код товара* → *Наименование товара*;

*Код товара* → *Наименование товара*.

Это приводит к следующим аномалиям.

*Аномалия включения.* Нельзя вставить данные в случае, когда в одном заказе может оказаться несколько одинаковых идентификаторов товара. Например, при покупке болтов можно заказать несколько коробок болтов в одном и том же заказе.

*Аномалия удаления.* Если какой-либо товар удалить из базы данных (снят с продажи), то при этом будет потеряна не только информация о тех заказах, в которых присутствовал этот товар, но и об оптовой цене на этот товар.

*Аномалия обновления.* При изменении цены или наименования товара необходим полный просмотр отношения с целью найти все заказы, чтобы изменение наименования товара было отражено во всех заказах. Пусть, к примеру, необходимо изменить описание "Болты" на "Болты М6", поскольку на склад поступили болты с другой маркой. Чтобы реализовать такую директиву, следует написать специальную программу, изменяющую все таблицы базы данных продаж и любые другие таблицы, в которых упоминается "Болт". Проблемы с целостностью информации, которые имеют место в реальных базах данных, могут привести к настоящему кошмару. Забыв или просто не зная о существовании всего одной таблицы, можно полностью привести в негодность информацию об этом товаре.

Для того чтобы выполнить требование второй нормальной формы для нашего примера, достаточно добавить к таблице новый столбец – идентификатор товара в заказе (код товара в заказе на **рис. 3.19**). Хотя новый товар был добавлен под номером 11, он по-прежнему относится к заказу, определенному порядковым номером заказа. В результате обеспечивается уникальная идентификация строк в таблице путем ввода нового первичного ключа и преодолевается проблема, связанная с операциями включения данных. Проблемы, связанные с операциями удаления и обновления, в данном случае не решаются, и необходима дальнейшая нормализация.

Код товара в заказе	Номер заказа	Код товара	Наименование товара	Оптовая цена
1	1	1320	Болты	12
2	1	1405	Молоток	20
3	1	1602	Гвозди	10
4	1	1201	Гайки	16
5	1	1000	Электропила	880
6	1	1320	Болты	12
1	2	2001	Напильник	19
2	2	1001	Ключ гаечный	14
3	2	2345	Отвертка	15
1	3	3021	Стамеска	24
2	3	4000	Рубанок	124

**Рис. 3.19.** Фрагмент организации данных, удовлетворяющих условию 2НФ

### 3.8.3. Третья нормальная форма

В таблице, соответствующей *третьей нормальной форме* (3НФ, *Third Normal Form, 3NF*), помимо требований для 2НФ, ни одно из неключевых полей таблицы не должно зависеть функционально от любого другого неключевого поля. Что это означает?

На приведенном выше **рис. 3.19** показан пример нарушения условий 3НФ, поскольку существует транзитивная функциональная связь непервичного атрибута "Код товара" от ключа "Номер товара в заказе" вида

*Наименование товара* → *Код товара* → *Номер товара в заказе*

Возникает аномалия включения данных: если товар не участвует в заказе, то в базе данных нельзя хранить информацию о его оптовой цене. Кроме этого по-прежнему остаются аномалии удаления и обновления.

На **рис. 3.20** представлена таблица 3НФ, в которой проблема скорректирована путем удаления столбцов "Наименование товара" и "Оптовая цена".

Код товара в заказе	Номер заказа	Код товара
1	1	1320
2	1	1405
3	1	1602
4	1	1201
5	1	1000
6	2	2001
7	2	1001
8	2	2345
9	3	3021
10	3	4000
11	1	1320

**Рис. 3.20.** Фрагмент таблицы из базы данных, находящихся в 3НФ

Чтобы не потерять информацию о кодировке товаров и ценах, создадим дополнительно таблицу "Инвентаризация" (**рис. 3.21**), в которой можно хранить информацию о наименовании товаров и ценах, и свяжем ее с исходной таблицей по ключевому полю "Код товара". Виды взаимосвязей в моделях данных мы рассмотрим далее.

После такой нормализации возникает вопрос о получении полной картины проданных товаров. Каким образом можно узнать всю информацию об определенном товаре, его описание, цену и так далее? В реляционной базе данных для этого существует механизм связывания таблиц по ключевым полям, для вывода необходимой информации используются так называемые виды (или представления), которые, в отличие от таблиц базы данных, не хранятся в ней постоянно, а служат лишь для представления результатов на экран. В данном примере две

таблицы связаны по ключевым полям "Код товара", и нужно просто создать запрос к БД, который возвратит всю необходимую информацию. На **рис. 3.22** показан вид, который объединяет рассматриваемые таблицы.

Код товара	Наименование товара	Оптовая цена
1000	Электропила	880
1001	Ключ гаечный	14
1201	Гайки	16
1320	Болты М4	12
1320	Болты М6	12
1405	Молоток	20
1602	Гвозди	10
2001	Напильник	19
2345	Отвертка	15
3021	Стамеска	24
4000	Рубанок	124

*Рис. 3.21. Фрагмент таблицы "Инвентаризация"*

Номер заказа	Код товара	Наименование товара	Розничная цена
1	1000	Электропила	1100
1	1001	Ключ гаечный	18
1	1201	Гайки	20
1	1320	Болты М4	16
1	1320	Болты М6	16
2	1405	Молоток	25
2	1602	Гвозди	13
2	2001	Напильник	24
3	2345	Отвертка	19
3	3021	Стамеска	30
3	4000	Рубанок	156

*Рис. 3.22. Пример выполнения запроса на выборку данных из двух связанных таблиц*

Распределяя информацию по нескольким таблицам, можно, как и раньше, получить необходимую для работы полноту данных. При этом изменяются только методы извлечения данных.

### **3.9. Семантическое моделирование данных. Диаграммы «сущность–связь»**

#### **3.9.1. Основные понятия**

Большинство проектировщиков современных информационных систем используют *реляционную модель* организации данных. В реляционной базе данных все данные представлены исключительно в формате таблиц или, в терминах реляционной алгебры, – отношений. Таблица в реляционной алгебре – это неупорядоченное множество записей (строк), состоящих из одинакового набора полей (столбцов). Каждая строка таблицы характеризует некий объект предметной области, а каждый столбец – одну из его характеристик. Совокупность таких связанных таблиц и составляет реляционную базу данных, при этом таблицы полностью равноправны, т.е. между ними не существует никакой иерархии.

Реляционная модель является простейшей и наиболее привычной для пользователя формой *представления* данных, однако разработка реляционной БД в терминах плоских таблиц часто представляет собой очень сложный и неудобный для проектирования процесс. Ограниченность реляционного подхода при разработке информационной модели предметной области проявляется, главным образом, в отсутствии достаточных средств для представления *смысла (семантики)* данных. Семантика реальной предметной области должна в этом случае представляться в голове проектировщика независимым от модели образом.

Потребности проектировщиков баз данных в более удобных и мощных средствах моделирования предметной области вызвали развитие семантических моделей данных. Главным назначением семантических моделей является обеспечение возможности выражения смысла данных реальной предметной области.

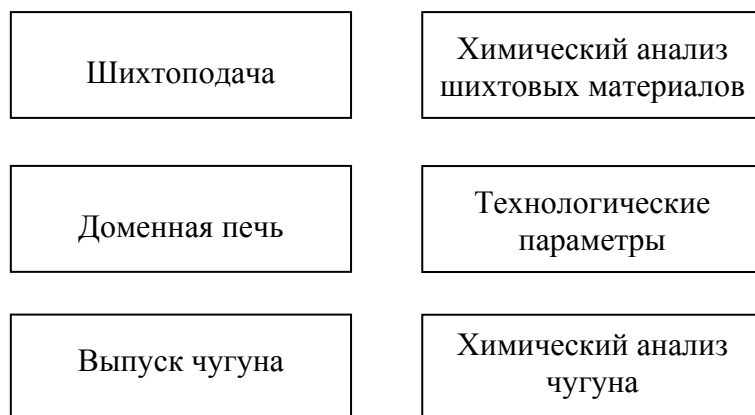
Семантическое моделирование широко используется на начальной стадии проектирования базы данных, а именно при разработке ее концептуальной модели. Ранее было отмечено, что концептуальная модель БД отражает наиболее важные с точки зрения пользователей информационные объекты предметной области и связи между ними безотносительно к конкретным программным средствам ее реализации, т.е. к СУБД. В дальнейшем концептуальная модель (схема) преобразуется к реляционной (или какой-нибудь другой) схеме под управлением методик, в которых четко оговорены все этапы такого преобразования.

Одной из наиболее распространенных семантических моделей данных является модель «сущность–связь» (Entity–Relationship), или ER-модель, которая впервые была предложена П.Ченом в 1976 году. Моделирование предметной области базируется на использовании графических диаграмм, или ER-диаграмм (Entity–Relationship Diagrams). Преимуществом такого подхода является наглядность представления концептуальных схем баз данных, благодаря чему ER-модели получили широкое распространение в компьютерных системах автоматизированного проектирования баз данных, или CASE-средствах (Computer Aided System Engineering).

Базовыми понятиями ER-модели являются «сущность», «связь» и «атрибут».

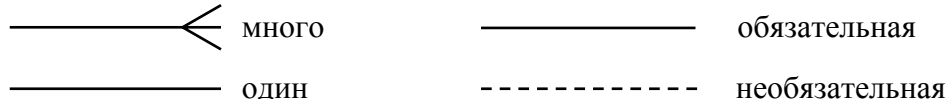
*Сущность (Entity)* – это реальный или представляемый объект (предмет или явление), имеющий существенное значение для рассматриваемой предметной области, о котором необходимо хранить информацию. Сущности именуются обычно существительными, например «Агрегат», «Параметры», «Шихтоподача», «Выпуск». Для уточнения предметной области к существительным могут добавляться определения, например «Металлургический агрегат», «Технологи-

ческие параметры» и т.д. В литературе можно встретить более строгое определение сущности, под которой понимают множество индивидуальных объектов – экземпляров, причем каждый экземпляр сущности должен быть отличим от любого другого экземпляра этой же сущности. С этой целью каждая сущность должна обладать одним или несколькими характеристиками (атрибутами), которые однозначно идентифицируют каждый экземпляр сущности. Это требование аналогично требованиям отсутствия одинаковых строк в реляционной таблице. На диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности (**рис. 3.23**).



**Рис. 3.23.** Графическое изображение сущности на ER-диаграмме

**Связь (Relationship)** – это именованная ассоциация, функциональная зависимость между двумя сущностями, *значимая* для рассматриваемой предметной области. В этом случае каждый экземпляр одной сущности ассоциирован с произвольным (в том числе и нулевым) количеством экземпляров второй сущности и наоборот. На ER-диаграммах связь представляется в виде линии, связывающей две сущности. Связи может даваться имя, выражаемое оборотом глагола и помещаемое возле линии связи. При этом в месте «стыковки» связи с сущностью используется трехточечный вход в прямоугольник сущности, если для этой сущности в связи могут использоваться много экземпляров сущности, и одноточечный вход, если в связи может участвовать только один экземпляр сущности. Обязательный конец связи изображается сплошной линией, необязательный – прерывистой (**рис. 3.24**).



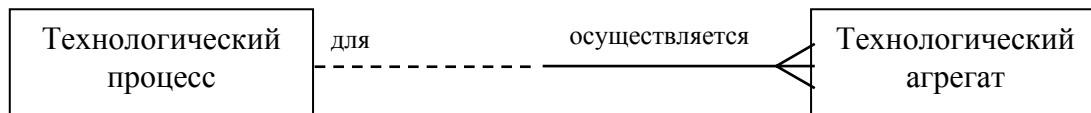
**Рис. 3.24.** Изображение степени и обязательности связи

На **рисунке 3.25** приведен пример связи между двумя сущностями «Технологический процесс» и «Технологический агрегат». Смысл данной связи можно выразить устно следующей трактовкой:



- каждый технологический процесс *осуществляется* в одном или более технологических агрегатах;
- каждый технологический агрегат служит *для* осуществления одного и только одного технологического процесса.

**Атрибутом (Attribute)** называется любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная с общим случае для классификации, идентификации, количественной характеристики и



**Рис. 3.25.** Связь между сущностями «Технологический процесс» и «Технологический агрегат»

выражения состояния сущности. Другими словами, атрибут выражает некоторое отдельное, интересующее пользователя свойство сущности, которое характеризует ее экземпляр. Отдельный атрибут определяется типом (числовой, текстовый, логический, временной и др.), а также значением, которое он принимает. Например, при описании доменного процесса в терминах ER-модели сущность «Выпуск чугуна» можно характеризовать четырьмя атрибутами: «Номер печи», «Номер выпуска», «Время начала выпуска» и «Время окончания выпуска». Два первых атрибута имеют целый тип и служат для идентификации, т.е. однозначного определения какого-либо выпуска на конкретной печи, а остальные два имеют тип дата и предназначены для его количественной оценки, в частности для расчета продолжительности процесса выпуска.

### 3.9.2. Методология IDEF1

Предложенный П.Ченом подход к моделированию данных послужил основой для разработки ряда методик по проектированию баз данных. В частности, получила широкое распространение методология IDEF1, позволяющая проектировать модель данных, эквивалентную реляционной модели в третьей нормальной форме. Методология IDEF1 (Icam DEFinition) составляла основную часть стратегической программы развития производства ICAM (интегрированная компьютеризация производства), проводимой по инициативе ВВС США. В настоящее время на основе совершенствования методологии IDEF1 создана ее новая версия – методология IDEF1X, отличающаяся простотой изучения и возможностью автоматизации процесса проектирования баз данных. В частности, IDEF1X-диаграммы используются в некоторых распространенных CASE-средствах для автоматизированной разработки программного обеспечения поддержки баз данных (ERwin, Design/IDEF и др.).

Метод IDEF1X определяет стандарты терминологии, используемой при информационном моделировании предметной области, и графического изображения типовых элементов (сущностей, связей и атрибутов) на диаграммах. Рассмотрим некоторые терминологические особенности и правила графического изображения ER-диаграмм, принятые в методологии IDEF1X.

В методологии IDEF1X сущность классифицируется по ее отношению с другими сущностями. Сущность является *независимой от идентификаторов* или просто независимой, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями. Сущность называется *зависимой от идентификаторов* или просто зависимой, если однозначная идентификация экземпляра сущности зависит от его отношения с другой сущностью. Зависимая сущность отображается прямоугольником с закругленными углами. Каждой сущности присваивается уникальное имя, помещаемое над блоком.



**Рис. 3.26.** Пример независимых (а) и зависимых (б) от идентификатора сущностей

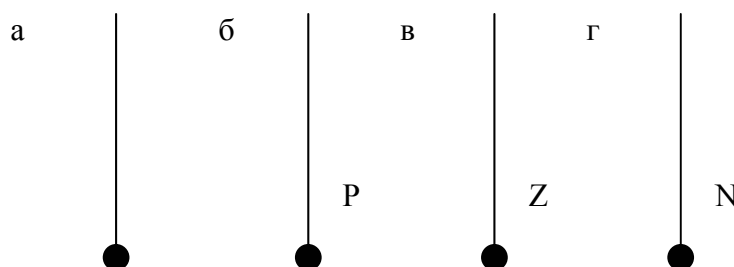
На **рисунке 3.26** приведены примеры независимых и зависимых от идентификаторов сущностей из области доменного производства. Все характеристики (экземпляры) сущностей «Доменная печь» и «Коксовая батарея» могут быть определены независимо от других сущностей, т. к. они отражают информацию об отдельных технологических агрегатах. Поэтому эти сущности являются независимыми от идентификаторов (см. **рис. 3.26, а**). Чтобы идентифицировать любое свойство такой сущности (экземпляр сущности), например диаметр колошника конкретной доменной печи, используют атрибут *этой же сущности*, называемый *ключевым атрибутом*. Ключевым атрибутом сущности «Доменная печь», с помощью которого определяются все остальные характеристики этой сущности, может быть, к примеру, номер доменной печи.

Сущности «Выпуск чугуна» и «Химический анализ кокса» являются зависимыми от идентификаторов, поскольку однозначная идентификация каждого отдельного значения атрибутов этих сущностей зависит от их отношения с другими сущностями (см. **рис. 3.26, б**). К примеру, сущность «Выпуск чугуна» находится в отношении с сущностью «Доменная печь» через атрибут «Номер печи». Действительно, чтобы получить информацию о времени начала и окончания выпуска чугуна на конкретной доменной печи, надо знать значения атрибутов «Номер выпуска» и «Номер печи». Так как количество доменных печей в цехе ограничено, то атрибут «Номер печи» может принимать только те значения, которые определены сущностью «Доменная печь». Аналогично можно показать, что сущность «Химический анализ кокса» находится в отношении с сущностью «Коксовая батарея» через атрибут «Номер коксовой батареи».

Определение связи в методологии IDEF1X несколько отличается от определения, данного П.Ченом. **Связь** – это ассоциация между сущностями, при ко-

торой каждый экземпляр одной сущности, называемой *родительской сущностью*, ассоциирован с произвольным (в том числе и нулевым) количеством экземпляров второй сущности, называемой *сущностью-потомком*, или *дочерней сущностью*, а каждый экземпляр сущности-потомка ассоциирован в точности с одним экземпляром сущности-родителя. Таким образом, экземпляр сущности-потомка может существовать только при существовании сущности-родителя. Связь на ER-диаграммах может дополнительно определяться с помощью указания *степени* или *мощности* (количества экземпляров сущности-потомка, которое может существовать для каждого экземпляра сущности-родителя). В IDEF1X могут быть выражены следующие 4 варианта мощности связей:

- каждый экземпляр сущности-родителя может иметь ноль, один или более связанных с ним экземпляров сущности-потомка;
- каждый экземпляр сущности-родителя должен иметь не менее одного связанного с ним экземпляра сущности-потомка;



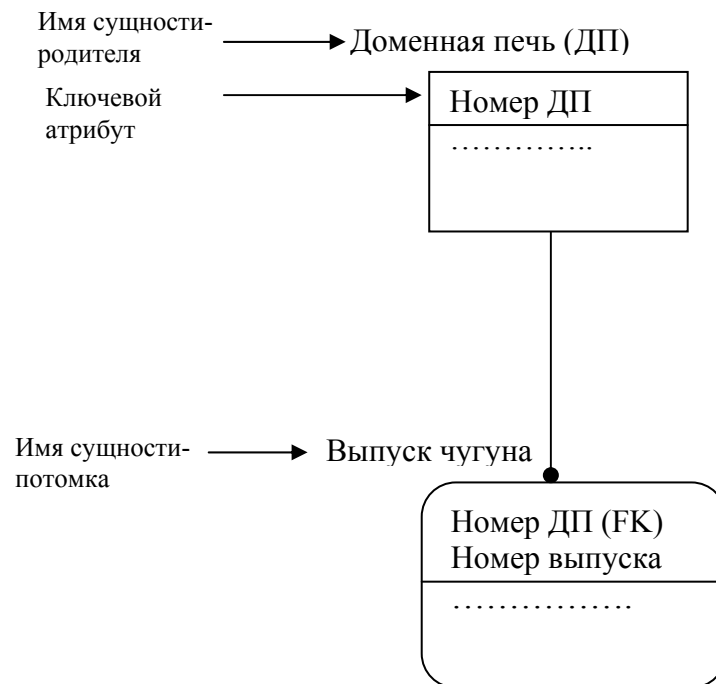
**Рис. 3.27.** Графическое изображение мощности связи в методологии IDEF: а – ноль, один или более; б – один или много (P); в – ноль или один (Z); г – мощность в точности равна некоторому положительному числу (N)

- каждый экземпляр сущности-родителя должен иметь не более одного связанного с ним экземпляра сущности-потомка;
- каждый экземпляр сущности-родителя связан с некоторым фиксированным числом экземпляров сущности-потомка.

Если экземпляр сущности-потомка однозначно определяется своей связью с сущностью-родителем, то связь называется *идентифицирующей*, в противном случае – *неидентифицирующей*. Связь изображается линией, проводимой между сущностью-родителем и сущностью-потомком, с точкой на конце линии у сущности-потомка. Мощность связи может принимать последующие значения: N – ноль, один или более; Z – ноль или один; P – один или более. В большинстве случаев мощность связи принимается равной N (**рис. 3.27**).

Идентифицирующая связь между сущностью-родителем и сущностью-потомком изображается сплошной линией. Сущность-потомок в идентифицирующей связи является зависимой от идентификатора сущностью. Сущность-родитель в идентифицирующей связи может быть как независимой, так и зависимой от идентификатора сущностью (это определяется ее связями с другими сущностями). На **рис. 3.28** показан пример изображения идентифицирующей связи между двумя сущностями «Доменная печь» и «Выпуск чугуна». Ранее было показано, что сущность «Выпуск чугуна» зависит от сущности «Доменная печь» через атрибут «Номер ДП», поэтому в данном случае сущность «Выпуск

чугуна» является потомком сущности «Доменная печь» и изображается прямоугольником с закругленными углами.



**Рис. 3.28.** Пример идентифицирующей связи

Пунктирная линия изображает неидентифицирующую связь. Сущность-потомок в неидентифицирующей связи будет независимой от идентификатора, если она не является также сущностью-потомком в какой-либо идентифицирующей связи. На **рис. 3.29** изображена неидентифицирующая связь между двумя сущностями «Доменная печь» и «Коксовая батарея», которые являются независимыми друг от друга в том смысле, что однозначная характеристика любого экземпляра этих сущностей не зависит от их взаимного отношения. Анализ особенностей технологии доменного производства показывает, что сущность «Коксовая батарея» является сущностью-родителем для сущности «Доменная печь», поскольку в коксовых батареях производят кокс, который затем поступает в доменные печи.

Атрибуты изображаются в виде списка имен внутри блока сущности. Атрибуты, определяющие первичный ключ, размещаются наверху списка и отделяются от других атрибутов горизонтальной чертой. Сущности могут иметь также внешние ключи (Foreign Key), которые могут использоваться в качестве части или целого первичного ключа или неключевого атрибута. Внешний ключ изображается с помощью помещения внутрь блока сущности имен атрибутов, после которых следуют буквы FK в скобках (см. **рис. 3.28**).

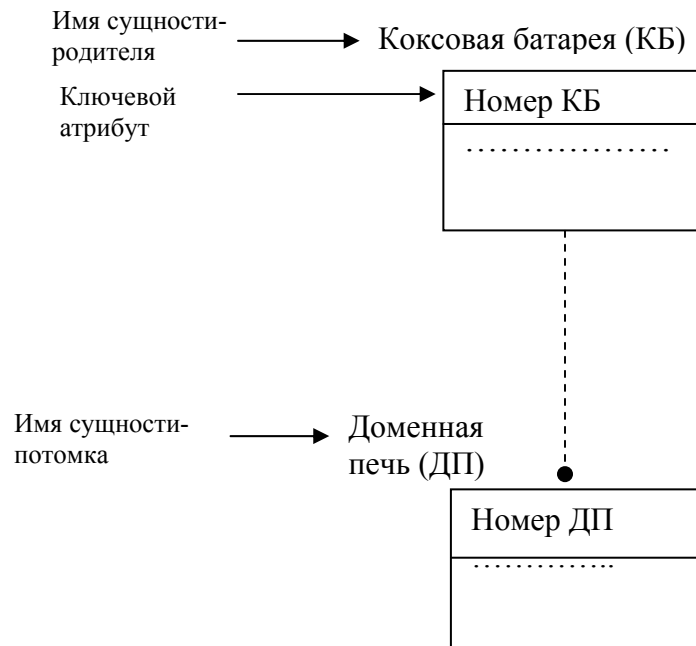


Рис. 3.29. Пример неидентифицирующей связи

### 3.10. Информационное моделирование с помощью CASE-средства ERwin

#### 3.10.1. Общая характеристика программы ERwin

Методология IDEF1X, как уже было отмечено, является основой для автоматизации процесса разработки систем баз данных и широко используется для информационного моделирования предметной области с помощью CASE-средств, реализующих CASE-технология создания и сопровождения информационных систем. Первоначальное значение термина CASE (Computer Aided System Engineering), ограниченное вопросами автоматизации разработки только лишь программного обеспечения, в настоящее время приобрело новый смысл, охватывающий процесс разработки информационных систем в целом. Под ним понимаются программные средства, поддерживающие процессы создания и сопровождения информационных систем, которые в общем случае включают следующие этапы:

- анализ и формулировку требований предметной области;
- проектирование баз данных и прикладного программного обеспечения;
- генерацию кода для выбранной СУБД и языка приложений;
- тестирование;
- документирование;
- обеспечение требуемого качества работы информационной системы и др.

**CASE-технология** представляет собой методологию проектирования информационных систем, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах разработки и сопровождения информационной системы и разрабатывать приложения в соответствии с информационными потребностями пользователей.

Рассмотрим некоторые аспекты информационного моделирования и его автоматизации с использованием программного CASE-средства ERwin v3.5 американской фирмы LogicWorks.

ERwin – это прежде всего средство концептуального моделирования базы данных, которое сочетает графический интерфейс Windows, инструменты для построения ER-диаграмм, редакторы для создания логической и физической моделей данных, а также поддержку различных сетевых реляционных СУБД и настольных баз данных. Существенным преимуществом является то, что с помощью ERwin можно создавать или проводить обратное проектирование (реинжиниринг) баз данных, т.е. преобразовывать физическую модель базы данных в концептуальную модель, не привязанную к конкретной СУБД.

ERwin создает визуальное представление (модель данных) для решаемой задачи в виде ER-диаграмм. Это представление может использоваться для детального анализа, уточнения и распространения в качестве части документации, необходимой в цикле разработки. В ERwin существуют два уровня представления и моделирования – логический и физический. *Логический уровень* означает прямое отображение фактов сущностей из реальной жизни. Например, печи, персонал, оборудование являются реальными объектами. Они именуются на естественном языке, с любыми разделителями слов (пробелы, запятые и т.д.). На логическом уровне не рассматривается использование конкретной СУБД, не определяются типы данных (например, целое или вещественное число) и индексы для таблиц. Целевая СУБД, имена объектов и типы данных, индексы составляют второй, *физический уровень* модели ERwin. ERwin предоставляет возможности создавать и управлять этими двумя различными уровнями представления диаграмм. Выбор между логическим и физическим уровнями отображения осуществляется через линейку инструментов или меню. Кроме этого, уровень детализации диаграммы информационной модели может изменяться проектировщиком. Например, могут отображаться только имена сущностей (таблиц), может быть включено/выключено отображение мощности связи и т.д.

Программа ERwin позволяет работать не со всей диаграммой, а с логически законченными группами сущностей (Subject Area), переключение между которыми производится выбором из раскрывающегося списка. Такая возможность позволяет проектировщику информационной модели удалить с экрана уже спроектированные блоки, чтобы они не загромождали диаграмму.

Все графические элементы модели ERwin могут редактироваться средствами, принятыми в Windows, – группировка, копирование, удаление, перемещение, использование системного буфера обмена. С помощью удобных диалоговых окон имеется возможность использовать цветовое и шрифтовое выделение для различных компонентов диаграммы. Выделения могут быть выполнены как для всей модели (например, все внешние ключи отображать красным цветом), так и

для отдельного компонента (таблицы, атрибутов одной таблицы, одной связи и т.д.). Компоненты модели, представленные текстом (имена сущностей, атрибутов, текстовые комментарии) могут редактироваться непосредственно на экране. Использование цветового и шрифтового выделений на диаграмме информационной модели делает ее более наглядной и позволяет проектировщику обратить внимание пользователей диаграммы на ее отдельные элементы.

### **3.10.2. Этапы построения информационной модели в ERwin**

Процесс построения информационной модели в ERwin состоит из следующих этапов:

- 1) определение сущностей;
- 2) определение связей (зависимостей) между сущностями;
- 3) задание первичных и составных (альтернативных) ключей;
- 4) определение атрибутов сущностей;
- 5) приведение модели к требуемому уровню нормальной формы;
- 6) переход к физическому описанию модели: назначение соответствий имя сущности – имя таблицы, атрибут сущности – атрибут таблицы; задание ограничений предметной области;
- 7) генерация базы данных, т.е. формирование физической схемы для конкретной выбранной (целевой) СУБД.

Сущность на диаграмме изображается прямоугольником. В зависимости от режима представления диаграммы прямоугольник может содержать имя сущности, ее описание, список ее атрибутов и другие сведения. Горизонтальная линия прямоугольника разделяет атрибуты сущности на два набора – атрибуты, составляющие первичный ключ в верхней части, и прочие (не входящие в первичных ключ) в нижней части. Сущность – это логическое понятие. Сущности соответствует таблица в реальной СУБД. В ERwin сущность визуально представляет три основных вида информации:

- атрибуты, составляющие первичный ключ;
- неключевые атрибуты;
- тип сущности (независимая/зависимая).

Связи отображают функциональную зависимость между двумя сущностями. Связь – это понятие логического уровня, которому соответствует внешний ключ на физическом уровне. В ERwin связи представлены пятью основными элементами информации:

- тип связи (идентифицирующая или неидентифицирующая связь);
- родительская сущность;
- дочерняя (зависимая) сущность;
- мощность связи;
- допустимость пустых (null) значений.

Напомним, что связь называется идентифицирующей, если экземпляр дочерней сущности идентифицируется через ее связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в первичный ключ дочерней сущности. Дочерняя сущность при иденти-

фицирующей связи всегда является зависимой. Связь называется неидентифицирующей, если экземпляр дочерней сущности идентифицируется иначе, чем через связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в состав неключевых атрибутов дочерней сущности.

Для определения связей ERwin выбирается тип связи, затем мышью указывается родительская и дочерняя сущности. Идентифицирующая связь изображается сплошной линией; неидентифицирующая – пунктирной линией. Линии заканчиваются точкой со стороны дочерней сущности. При определении связи происходит автоматическое перемещение (миграция) атрибутов первичного ключа родительской сущности в соответствующую область атрибутов дочерней сущности. Поэтому такие атрибуты не вводятся вручную. Атрибуты первичного ключа родительской сущности по умолчанию мигрируют со своими именами. ERwin позволяет ввести для них роли, т.е. новые имена, под которыми мигрирующие атрибуты будут представлены в дочерней сущности. На физическом уровне имя роли – это имя колонки внешнего ключа в дочерней таблице.

Мощность связи в соответствии с методологией IDEF1X представляет собой отношение количества экземпляров родительской сущности к соответствующему количеству экземпляров дочерней сущности. Мощность связи записывается как 1:N. ERwin предоставляет 4 варианта для n, которые изображаются дополнительным символом у дочерней сущности: ноль, один или больше (по умолчанию); ноль или один; один или более; ровно N, где N – конкретное число. Допустимость пустых (null) значений в неидентифицирующих связях ERwin изображает пустым ромбиком на дуге связи со стороны родительской сущности.

Для каждой связи на логическом уровне могут быть заданы требования по обработке операций вставки, обновления и удаления (insert, update, delete) для родительской и дочерней сущностей. Программа ERwin предоставляет следующие варианты обработки этих событий:

- отсутствие проверки;
- проверка допустимости;
- запрет операции;
- каскадное выполнение операции удаления/обновления (delete/update);
- установка пустого (null-значения) или заданного значения по умолчанию.

В соответствии с выбранным вариантом программа ERwin автоматически создает необходимые процедуры обработки этих событий (триггеры) на языке SQL целевой СУБД, которые могут быть переопределены после генерации схемы базы данных.

Разработанные модели ERwin сохраняются на диске в виде файла с расширением .er1. Имеется возможность хранить модель в целевой СУБД. Для этого с помощью самой программы ERwin в целевой СУБД создается метабаза ERwin, в которой сохраняется информация о модели.



## 3.11. Проектирование базы данных доменного производства

### 3.11.1. Концептуальное и логическое проектирование

Рассмотрим пример информационного моделирования и разработки базы данных доменного цеха с использованием средства автоматизированного проектирования ERwin. Предполагаем, что читатель знаком в общих чертах с технологией доменного производства, поэтому в материале будут отражены, главным образом, специфические особенности доменной печи как объекта информационных систем.

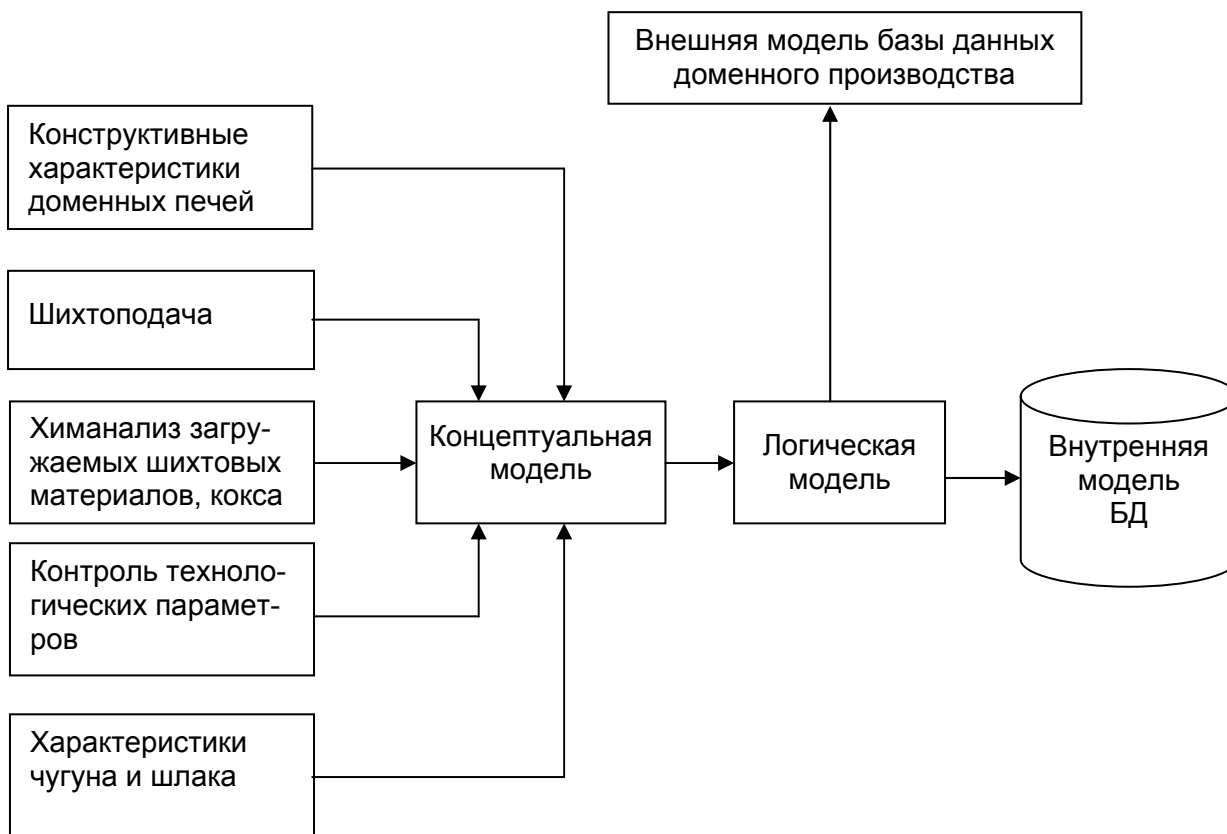
Учитывая специфику технологии доменной плавки, особенности доменной печи как объекта информационных систем, требования к содержанию и качеству представляемой информации, можно выделить следующие основные информационные объекты (сущности) для проектирования базы данных доменного цеха:

1. **Доменная печь**, включающая информацию о конструктивных характеристиках печей доменного цеха:
  - номер печи;
  - наименование печи;
  - геометрические размеры и конструктивные характеристики основных элементов печи.
2. **Шихтоподача**, которая отражает информацию о системах загрузки, типах и расходах загружаемых материалов. В частности, в него включены следующие параметры:
  - тип загружаемых материалов;
  - соотношение загружаемых материалов в подачу;
  - вес подачи;
  - порядок загрузки;
  - номер подачи;
  - дата поступления информации в базу данных.
3. **Химический анализ загружаемых шихтовых материалов**, отражающий информацию о физико-химических свойствах загружаемых материалов и времени проведения химического анализа, определения физических свойств:
  - дата проведения анализа;
  - химический состав шихтовых материалов:
    - агломерат: Fe, FeO, CaO, SiO<sub>2</sub>, Al<sub>2</sub>O<sub>3</sub>, MgO, Mn, S, CaO/SiO<sub>2</sub>;
    - окатыши: Fe, FeO, CaO, SiO<sub>2</sub>, Al<sub>2</sub>O<sub>3</sub>, MgO, Mn, S, P, CaO/SiO<sub>2</sub>, прочность;
    - железная руда: Fe, FeO, CaO, SiO<sub>2</sub>, Al<sub>2</sub>O<sub>3</sub>, MgO, Mn, S, P;
    - добавки: Fe, FeO, CaO, SiO<sub>2</sub>, Al<sub>2</sub>O<sub>3</sub>, MgO, Mn, S, P;
    - кокс: зола кокса, летучие кокса, влага в коксе, механическая прочность, C, S.
4. **Технологические параметры** доменной плавки, отражающие основные контролируемые характеристики дутьевых параметров, отходящего колошникового газа, характеризующих газодинамический режим плавки:

- дата поступления информации;
  - параметры:
    - расход дутья,  $\text{м}^3$ ;
    - давление горячего дутья, ати;
    - давление холодного дутья, ати;
    - содержание  $\text{O}_2$  в дутье, %;
    - влажность дутья,  $\text{г}/\text{м}^3$ ;
    - расход природного газа,  $\text{м}^3/\text{мин}$ ;
    - расход  $\text{O}_2$ ;
    - температура дутья,  $^\circ\text{C}$ ;
    - давление газов под колошником, ати;
    - $\text{CO}$  в газе, %;
    - $\text{CO}_2$  в газе, %;
    - $\text{H}_2$  в газе, %;
    - $\text{SO}_2$  в газе, %;
    - температура колошникового газа,  $^\circ\text{C}$ ;
    - перепад давлений верхний, ати;
    - перепад давлений общий, ати;
    - температура газов в общем газоотводе,  $^\circ\text{C}$ ;
    - температура газов в свечевых газоотводах,  $^\circ\text{C}$ ;
    - температура газов по показаниям периферийных термопар,  $^\circ\text{C}$ .
5. *Характеристики чугуна и шлака*, отражающие временные интервалы обработки жидких продуктов плавки, их химический состав и физический нагрев:
- номер выпуска;
  - время начала выпуска;
  - время окончания выпуска;
  - дата выпуска;
  - температура чугуна на выпуске;
  - химический анализ чугуна и шлака:
    - в чугуне: Si, Mn, S, P, Ti, V, C;
    - в шлаке:  $\text{SiO}_2$ , MgO, MnO,  $\text{Al}_2\text{O}_3$ ,  $\text{TiO}_2$ ,  $\text{V}_2\text{O}_5$ , S;

Структура информационной модели базы данных доменного производства приведена на **рис. 3.30**.

После того как выделены основные информационные объекты (сущности) информационной системы, необходимо определить схему отношения между отдельными объектами с указанием типа связи между ними, определить ключевые и неключевые атрибуты сущностей. С учетом основных принципов нормализации реляционных отношений, а также для обеспечения свойств целостности, согласованности и непротиворечивости проектируемой базы данных введем дополнительные, вспомогательные сущности, которые представлены в **табл. 3.2**. Полный список сущностей с указанием их ключевых и неключевых атрибутов, типов полей данных приведен в **табл. 3.3**.



*Рис. 3.30. Структура информационной модели базы данных доменного производства*

Таблица 3.2

**Характеристика вспомогательных сущностей  
для информационной модели доменного производства**

Наименование	Описание
Шихтовые материалы	Наименования используемых шихтовых материалов
Доли шихтовых материалов	Соотношение весовых долей шихтовых материалов, загружаемых в печь
Коксовая батарея	Характеристики коксовых батарей
Химанализ кокса	Химический анализ кокса, поступающего в доменный цех
Кокс на печь	Соответствие между коксовой батареей и доменной печью
T_параметры	Перечень технологических параметров, контролируемых в ходе работы печи, с указанием их идентификаторов для однозначного определения их в базе данных
Выпуск	Информация о выпуске жидких продуктов плавки из доменной печи (дата выпуска, время начала и окончания выпуска, температура чугуна на выпуске)
НСИ элементов	Нормативно-справочная информация по физико-химическим соединениям, образующимся в жидких продуктах плавки

Таблица 3.3

**Данные по доменному переделу, приведенные  
к третьей нормальной форме**

Имя сущности (таблицы)	Имя атрибута сущности (поля таблицы)	Тип данных	Описание
1	2	3	4
Доменная печь	<b>№ печи</b>	Числовой	Ключевое поле
	Наименование печи	Текстовый	
	Полная высота_м	Числовой	
	Полезная высота_м	Числовой	
	Высота зумпфа_м	Числовой	
	Высота горна_м	Числовой	
	Высота заплечиков_м	Числовой	
	Высота распара_м	Числовой	
	Высота шахты_м	Числовой	
	Высота колошника_м	Числовой	
	Диаметр горна_м	Числовой	
	Диаметр распара_м	Числовой	
	Диаметр колошника_м	Числовой	
	Угол наклона шахты_град	Числовой	
	Угол наклона заплечиков_град	Числовой	
	Полезный объем печи_м3	Числовой	
	Число воздушных фурм_шт	Числовой	
Диаметр фурм_м	Числовой		
Число чугунных леток_шт	Числовой		
Число работающих фурм_шт	Числовой		
Шихтоподача	<b>№ печи</b>	Числовой	Составной ключ
	<b>Дата шихтовки</b>	Дата/время	
	<b>№ подачи</b>	Числовой	
	Вес подачи_кг	Числовой	
	Порядок загрузки	Текстовый	
Доли шихтовых материалов	<b>№ печи</b>	Числовой	Составной ключ
	<b>№ подачи</b>	Числовой	
	<b>Код материала</b>	Числовой	
	<b>Дата шихтовки</b>	Дата/время	
	Доля материала_%	Числовой	
Т_Параметры	<b>Код параметра</b>	Числовой	Ключевое поле
	Наименование параметра	Текстовый	
Шихтовые материалы	<b>Код материала</b>	Числовой	Ключевое поле
	Наименование материала	Текстовый	

Окончание таблицы 3.3

1	2	3	4
Химанализы шихтовых материалов	<b>Код материала</b>	Числовой	Составной ключ
Технологические параметры	<b>Код элемента</b>	Числовой	
	Значение	Числовой	
	Дата	Дата/время	
	<b>№ печи</b>	Числовой	Составной ключ
НСИ элементов	<b>Код параметра</b>	Числовой	
	<b>Дата</b>	Дата/время	
	Значение	Числовой	
	<b>Код элемента</b>	Числовой	Ключевое поле
Кокс на печь	Наименование элемента	Текстовый	
	<b>№ печи</b>	Числовой	Составной ключ
Коксовая батарея	<b>№ коксовой батареи</b>	Числовой	
	<b>№ коксовой батареи</b>	Числовой	Ключевое поле
Химанализ кокса	Наименование коксовой батареи	Текстовый	
	<b>№ коксовой батареи</b>	Числовой	Составной ключ
Выпуск	<b>Дата</b>	Дата/время	
	<b>Код элемента</b>	Числовой	
	Значение	Числовой	
	<b>№ печи</b>	Числовой	Составной ключ
Химанализ чугуна и шлака	<b>№ выпуска</b>	Числовой	
	<b>Характер расплава</b>	Текстовый	
	Дата выпуска	Дата/время	
	Время начала выпуска	Дата/время	
	Время окончания выпуска	Дата/время	
	Температура чугуна	Числовой	
	<b>№ печи</b>	Числовой	Составной ключ
	<b>№ выпуска</b>	Числовой	
	<b>Характер расплава</b>	Текстовый	
	<b>Код элемента</b>	Числовой	
	Значение	Числовой	

Перейдем к реализации модели данных доменного производства в программе ERwin.

Сначала создадим логический уровень модели. Для этого зададим режим отображения сущностей Edit/Logical Model. Создадим при помощи линейки инструментов (ERwin Toolbox) сущности "Доменная печь", "Шихтоподача", "Химанализы шихтовых материалов" и т.д. Сущности будем именовать на русском языке. Выбрав каждую сущность, зададим для нее подробное описание (Definition) на русском языке в редакторе "Entity Editor" (рис. 3.31). Это описание появится в отчетах ERwin и может быть отображено на диаграмме.

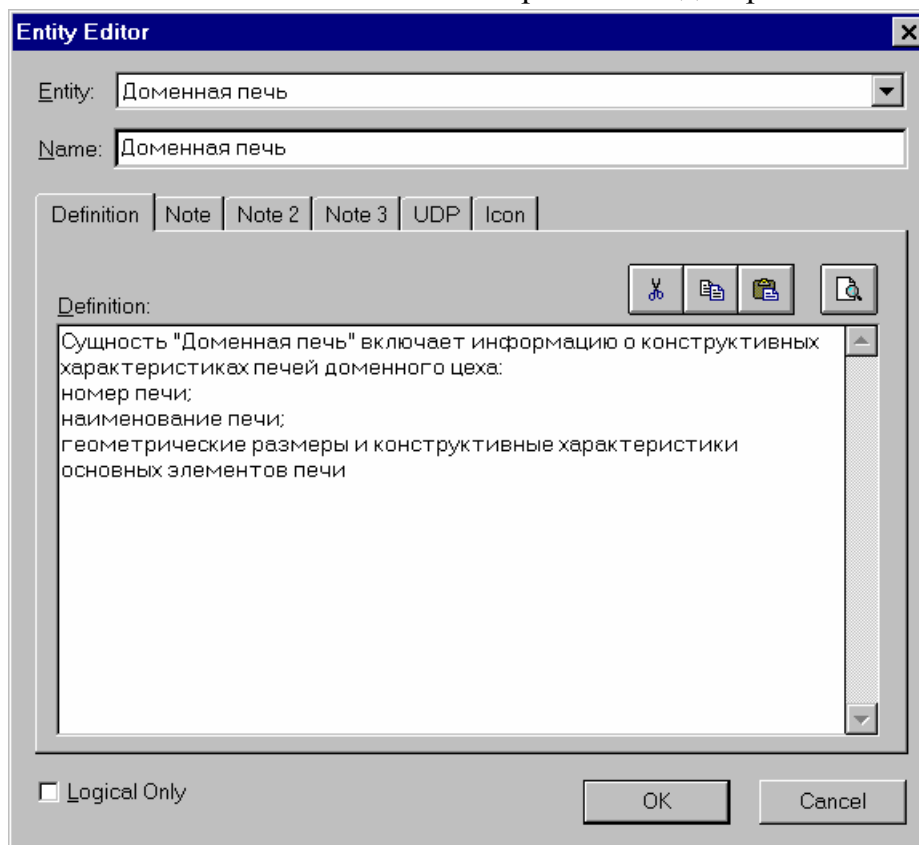


Рис. 3.31. Редактор описания сущностей

Далее укажем связи между сущностями. Например, сущность "Доменная печь" связана идентифицирующей связью, т.е. является родительской сущностью для сущностей "Шихтоподача", "Выпуск", "Кокс на печь" и "Технологические параметры". Описание связи вводится в редакторе "Relationship Editor" (рис. 3.32). На вкладке General отображаются названия сущности-родителя и сущности-потомка, имеется возможность задать мощность (Cardinality) связи, а также ее тип (идентифицирующая, неидентифицирующая). На других вкладках можно ввести подробное описание связи (вкладка Definition), определить правила для контроля целостности отношений между сущностями (вкладка Rolename/RI Actions), а также определить так называемые правила (проверки допустимых значений) и начальные (по умолчанию) значения для любых логических или физических объектов ERwin (вкладка UDP).

Теперь перейдем в режим задания атрибутов Edit/Attribute. В диалоговом окне "Attribute Editor" (рис. 3.33) зададим на русском языке имена ключевых и

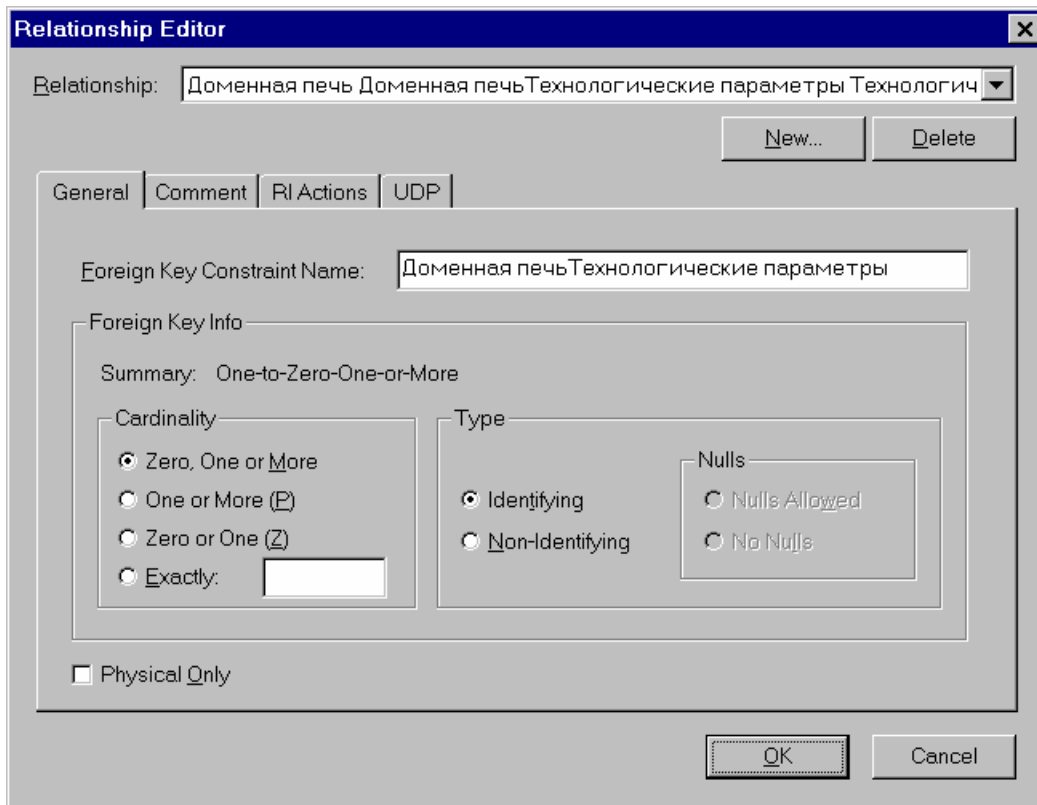


Рис. 3.32. Редактор описания связей

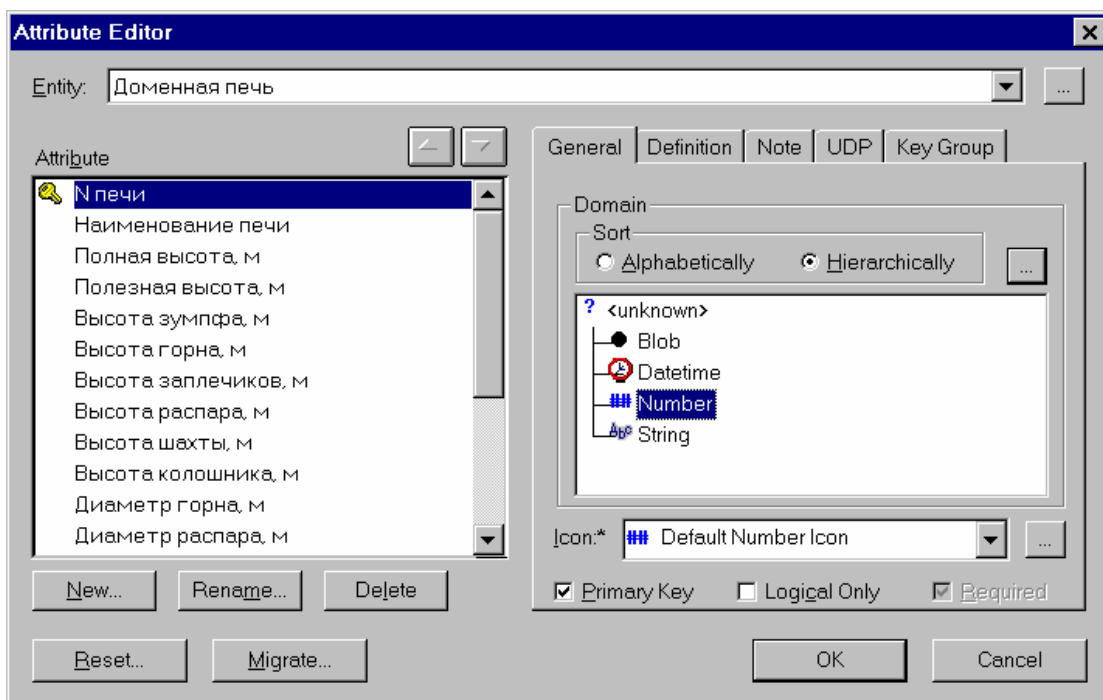


Рис. 3.33. Редактор описания атрибутов сущности

неключевых атрибутов. Заметим, что некоторые ключевые атрибуты для дочерних сущностей не указываются вручную, они автоматически переходят

(мигрируют) из родительской сущности. Например, при вводе атрибутов для сущности "Технологические параметры" в качестве ключевого здесь уже присутствует атрибут "№ печи", который перешел из родительской сущности "Доменная печь". Дополнительно здесь же зададим два ключевых атрибута "Код параметра" и "Дата", а также неключевой атрибут "Значение". Таким образом мы создали составной ключ из трех атрибутов "№ печи", "Код параметра" и "Дата", чтобы однозначно идентифицировать каждый технологический параметр любой доменной печи цеха в определенный момент времени. Аналогично введем другие атрибуты всех сущностей для разработанной схемы базы данных.

Отображение на логическом уровне информационных объектов доменного производства, их атрибутов, а также отношений между ними в виде ER-диаграммы приведено на **рис. 3.34**.

### **3.11.2. Физическая реализация информационной модели**

Перейдем к физической реализации базы данных доменного производства. Для этого выберем целевую систему управления базой данных в диалоговом окне "Server/Target Server", например Microsoft (MS) Access 97 (**рис. 3.35**). Как видно из рисунка, ERwin поддерживает как самые современные, так и предыдущие версии основных программ СУБД – INFORMIX, ORACLE, SQL Server, SQL Base, SYBASE, FoxPro, Clipper, dBASE, Paradox и др.

После выбора целевой СУБД проектирование на физическом уровне выполняется в терминах той базы данных, которую предполагается использовать в информационной системе. Важно подчеркнуть, что программе ERwin "известны" соответствия между возможностями СУБД различных производителей, вследствие чего возможно преобразование физической схемы, спроектированной для одной СУБД, в другую. Такой процесс преобразования называется *обратным проектированием (reverse engineering)* и используется при выборе оптимальной аппаратной платформы для существующей базы данных, а также при расширении или модификации существующей логической структуры информационной системы.

Выберем команду меню "Tasks/Forwards Engineer/Schema Generation...". Появляется окно "Access Schema Generation" (**рис. 3.36**), в котором показан диалог выбора параметров генерации базы данных. В ходе процесса генерации программа ERwin строит пакет SQL-команд для создания структуры базы данных. На рисунке видно, что пользователь может определить фильтр (Filter), т.е. генерировать не все таблицы, пакет SQL-команд можно просмотреть (Preview), распечатать (Print), сохранить в файл отчета (Report), выполнить генерацию (Generate).



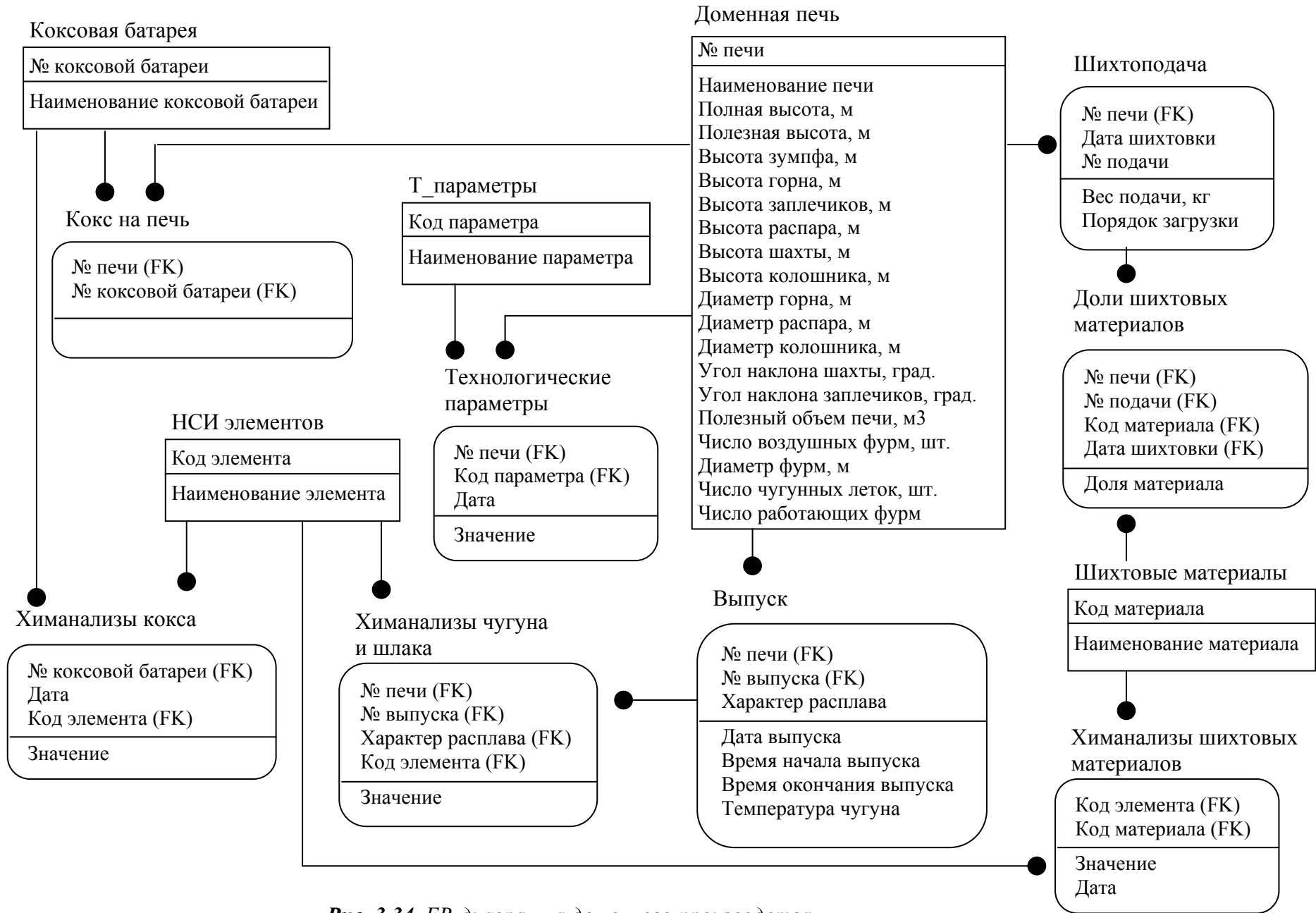
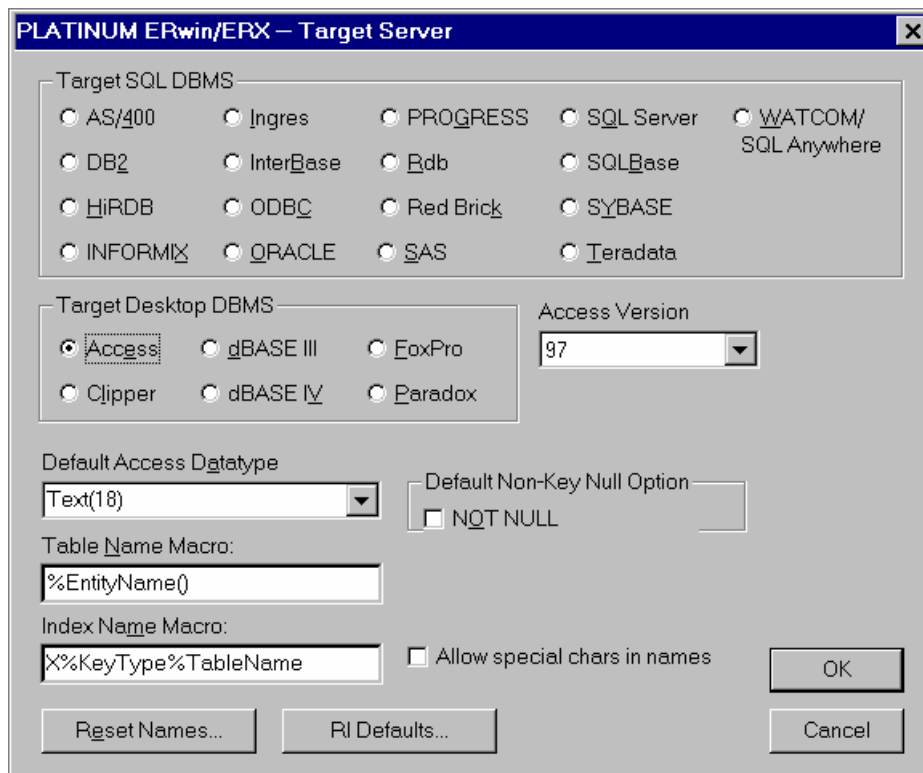
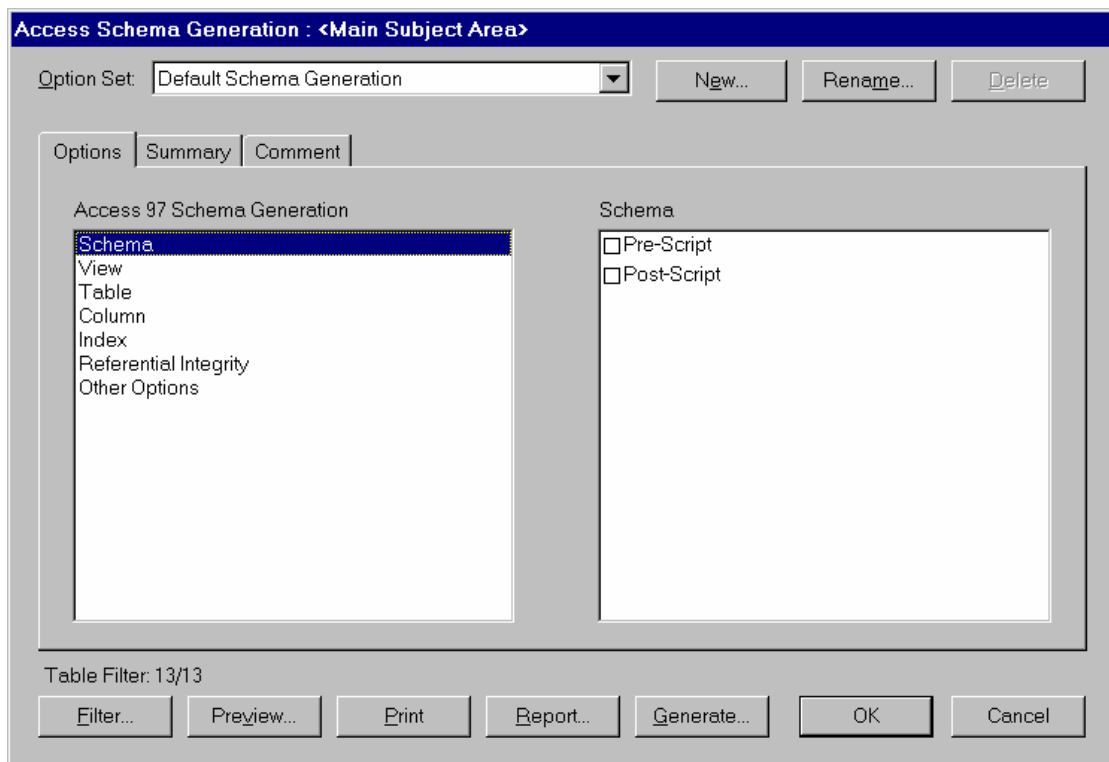


Рис. 3.34. ER-диаграмма доменного производства

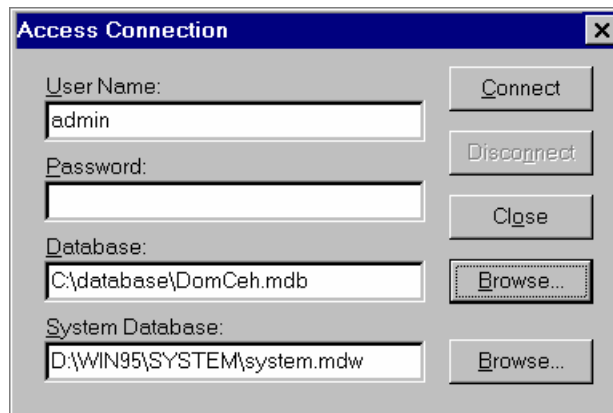


*Рис. 3.35. Диалог выбора целевой СУБД для физической реализации модели доменного производства*



*Рис. 3.36. Окно генерации схемы базы данных в MS Access*

После выбора процедуры генерации следует заполнить пустые поля появившегося окна подключения к СУБД MS Access (**рис. 3.37**), в котором содержится следующая информация:



*Рис. 3.37. Окно подключения к СУБД MS Access*

- имя пользователя (User Name), от имени которого производится подключение к системе управления базой данных MS Access. В нашем случае полагаем, что подключение осуществляется от имени администратора базы данных, имя которого admin;
- пароль (Password) пользователя;
- местоположение и имя файла базы данных (Database), в которую будет сгенерирована разработанная логическая схема. Заметим, что файл базы данных с указанным именем должен быть предварительно создан в программе MS Access 97. В нашем случае был создан пустой файл с именем Dom.mdb в корневом каталоге диска "C";
- местоположение и имя системной базы данных (System Database) пакета MS Access.

После заполнения указанных полей окна подключения следует нажать кнопку Connect и ждать выполнения всех SQL-команд по созданию структуры базы данных, которое отражается в окне "Generate Database Schema". Об успешном окончании процесса генерации структуры базы данных свидетельствует запись Schema Generation Complete, 494 query succeeded.

Результат выполнения можно просмотреть, загрузив в программу MS Access файл "Dom.mdb". На **рис. 3.38** показана схема базы данных доменного производства, которая состоит из нескольких таблиц, связанных между собой по ключевым полям.

В процессе разработки и совершенствования информационной системы может возникнуть ситуация, когда физическая структура базы данных, например в MS Access, и информационная модель в ERwin не соответствуют друг другу. Для этого в программе ERwin предусмотрена функция синхронизации с базой данных, которая вызывается командой меню "Tasks/Complete Compare...". Появившееся диалоговое окно (**рис. 3.39**) предлагает список несоответствий между существующей структурой базы данных MS Access и моделью ERwin.

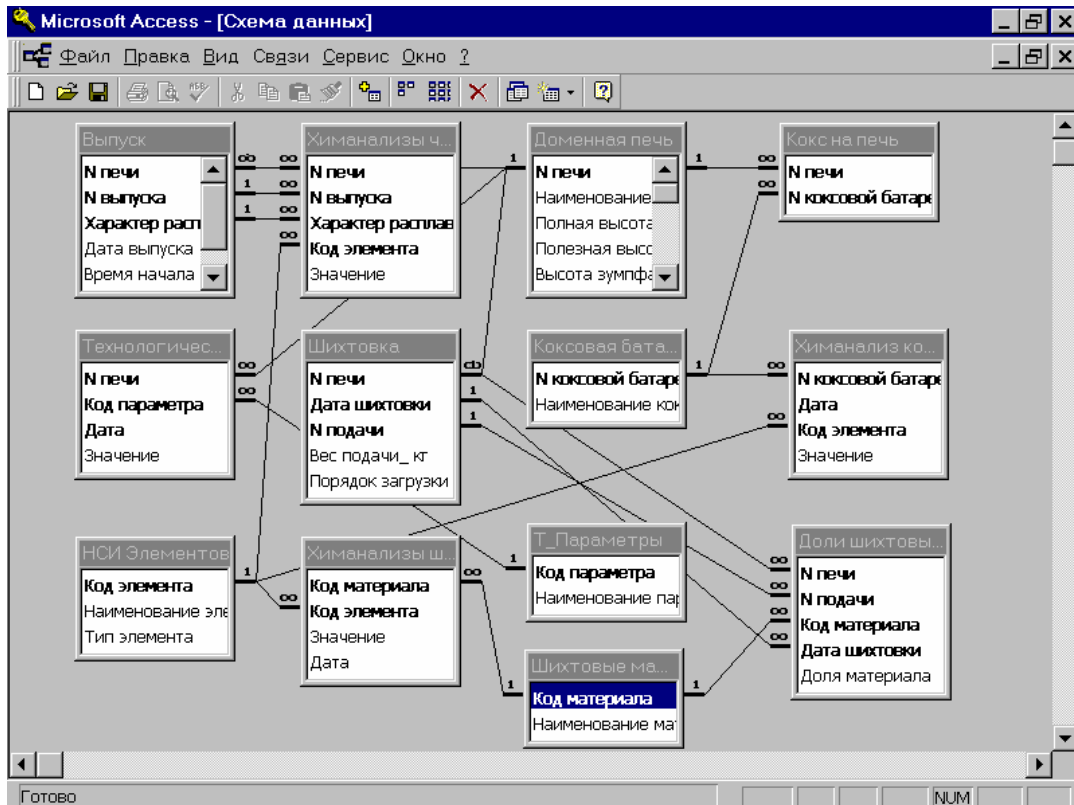


Рис. 3.38. Схема базы данных доменного производства, сгенерированная в СУБД MS Access

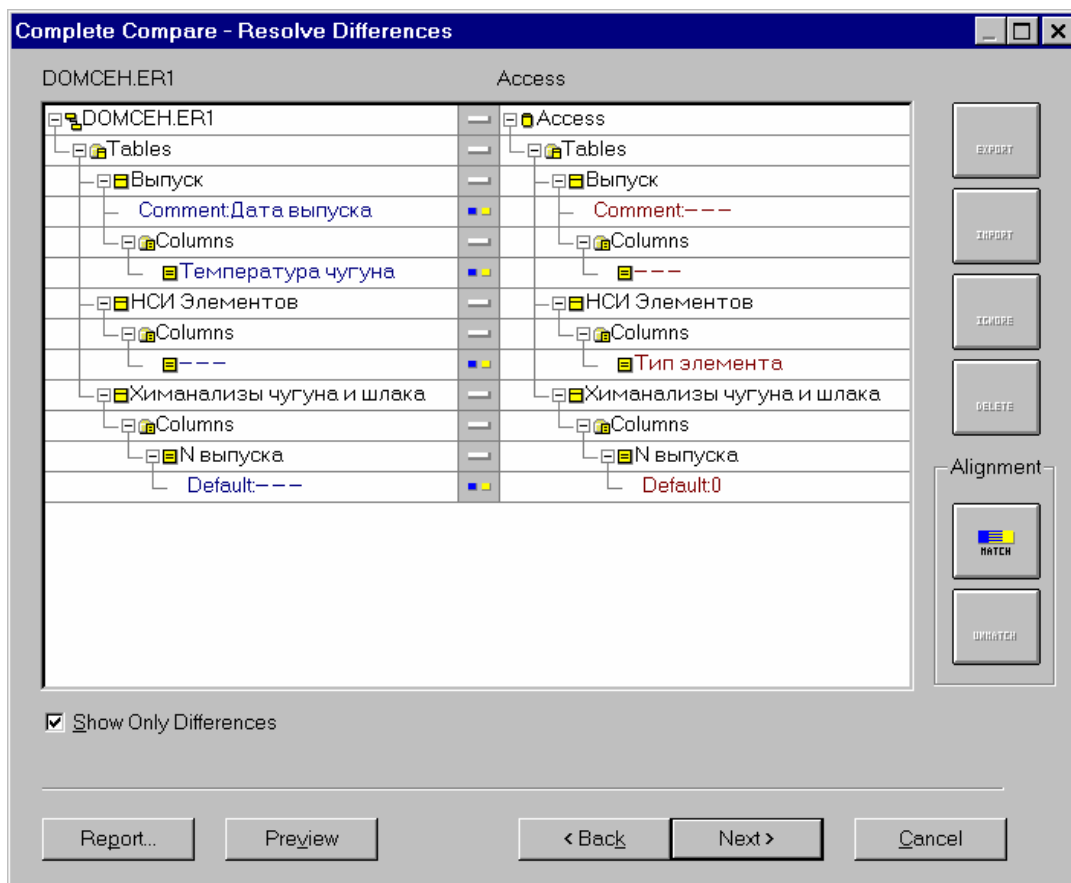


Рис. 3.39. Окно выбора синхронизируемых объектов информационной модели ERwin и базы данных MS Access

Например, если в базе данных создана новая таблица, то ERwin предложит провести включение ее в модель. Если в модель добавлена новая сущность, ERwin предложит создать на ее основе таблицу в реальной базе данных. Аналогично: при добавлении или модификации полей в базе данных или атрибутов сущностей в информационной модели ERwin предлагает провести соответствующие операции по синхронизации.

При завершении работы над информационной моделью, как правило, необходимо распечатать логический и физический уровни диаграммы, а также отчеты по соответствиям "сущность–таблица", "атрибут–имя колонки". Диаграмма физической модели является необходимым и очень удобным материалом для разработчиков прикладных программ, использующих обращения к базе данных. Для этого с помощью команды меню "Tasks/Generate Reports..." можно вызвать

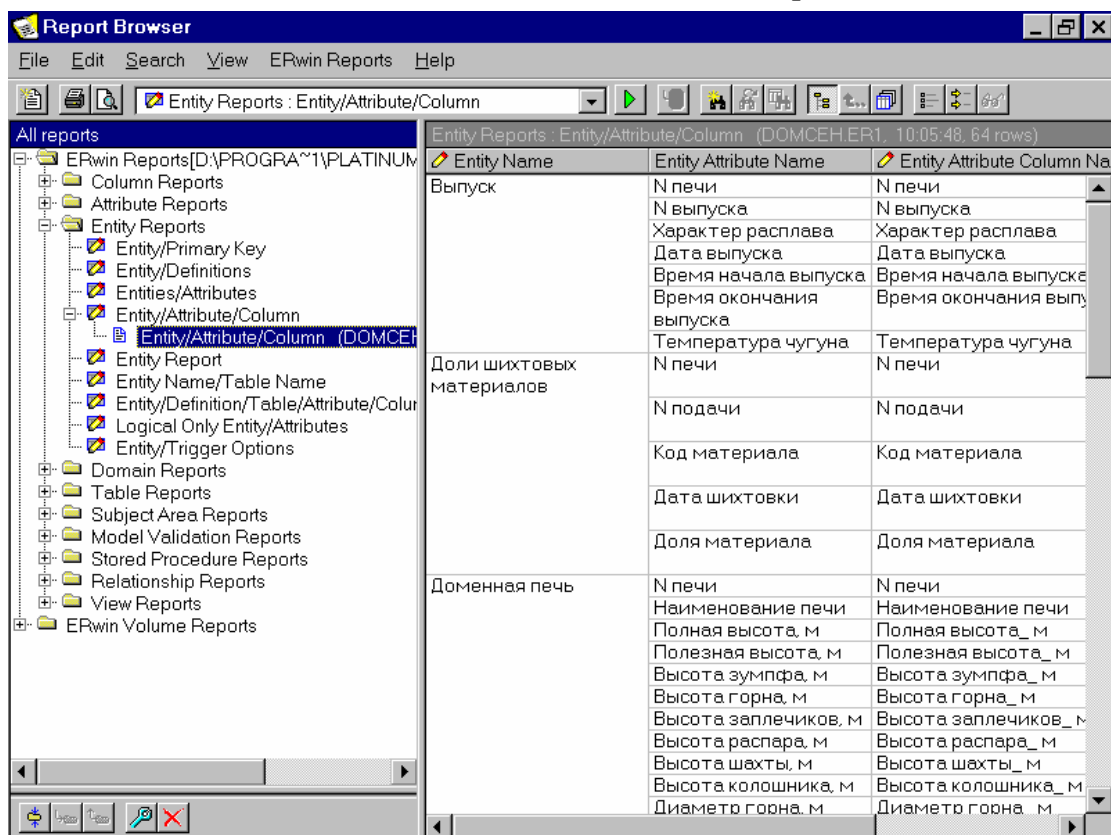


Рис. 3.40. Окно просмотра различных видов отчета ERwin

диалоговое окно "Report Browser", в котором представлены всевозможные виды отчетов по разработанной информационной модели. Рисунок 3.40 иллюстрирует пример построения отчета о соответствии названий сущностей и их атрибутов именам колонок реальной базы данных.

Сгенерированный отчет может быть выведен на печать или сохранен на диске в удобном формате, предусматривающем его последующую корректировку в текстовом редакторе или электронных таблицах.

### 3.12. Контрольные вопросы

1. Чем характеризуется традиционный подход к организации данных? В чем проявляется его ограниченность?
2. Какие компоненты включает в себя система баз данных?
3. В чем заключаются преимущества и недостатки использования системы баз данных для построения информационных систем?
4. Какие варианты архитектур используются для построения многопользовательских централизованных систем баз данных с удаленным сетевым доступом?
5. Поясните принципы работы централизованной и распределенной систем баз данных. Какая из этих систем является более перспективной и почему?
6. Дайте характеристику клиент/серверной технологии построения программного обеспечения. С какой целью производится деление компьютерного приложения на отдельные уровни?
7. На каких принципах основан реляционный подход к организации данных? Перечислите основные понятия реляционных баз данных.
8. Из каких этапов состоит процесс разработки баз данных при классической методологии проектирования? Какие свойства при этом необходимо обеспечить?
9. В чем состоит основная идея метода нормализации схемы базы данных? Поясните условия, которые необходимо обеспечить для приведения схемы отношения базы данных к первой, второй и третьей нормальным формам.
10. Чем вызвана необходимость семантического моделирования данных? Дайте определения понятиям «сущность», «связь», «атрибут».
11. Что понимается под CASE-технологией разработки информационных систем?

---

ГЛАВА 4. ИНФОРМАЦИОННЫЕ СЕТЕВЫЕ ТЕХНОЛОГИИ	4-2
4.1. Общие принципы построения компьютерной сети	4-2
4.1.1. Понятие компьютерной сети	4-2
4.1.2. Основные программные и аппаратные компоненты сети	4-5
4.1.3. Характеристики коммуникационной сети	4-5
4.1.4. Адресация компьютеров	4-6
4.2. Особенности локальных компьютерных сетей	4-7
4.2.1. Два типа сетей	4-7
4.2.2. Топология сети	4-8
4.2.3. Физическая передающая среда локальных компьютерных сетей	4-12
4.2.4. Платы сетевого адаптера	4-15
4.3. Функционирование сети	4-17
4.3.1. Сетевые модели OSI и IEEE Project 802	4-17
4.3.2. Назначение и функции протоколов	4-24
4.3.3. Основные стандартные стеки коммуникационных протоколов	4-25
4.3.4. Передача данных по сети	4-29
4.3.5. Методы доступа в сетях	4-33
4.4. Стандартные сетевые технологии	4-38
4.5. Сетевые операционные системы	4-41
4.6. Способы и средства объединения локальных компьютерных сетей	4-45
4.7. Методы передачи данных в компьютерных сетях	4-54
4.8. Общая характеристика Internet	4-59
4.9. Система адресации в Internet (Intranet)	4-63
4.10. Назначение и функции брандмауэров и прокси-серверов	4-73
4.11. Типы сервиса Internet (Intranet)	4-76
4.12. Контрольные вопросы	4-80

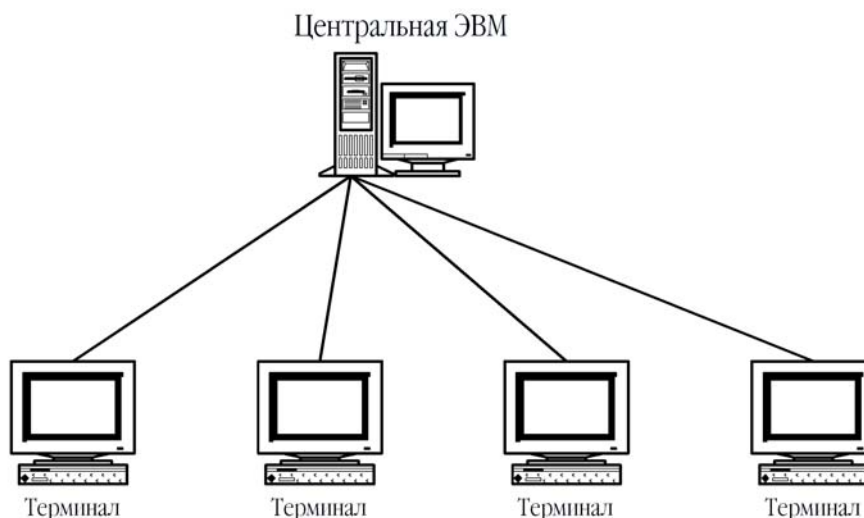
## Глава 4. ИНФОРМАЦИОННЫЕ СЕТЕВЫЕ ТЕХНОЛОГИИ

### 4.1. Общие принципы построения компьютерной сети

#### 4.1.1. Понятие компьютерной сети

История развития сетей является подтверждением теории спирального развития.

Связь на небольших расстояниях в вычислительной технике применялась задолго до появления первых персональных компьютеров, еще в то время, когда существовали только большие ЭВМ (мэйнфреймы), к которым подключались неинтеллектуальные терминалы. «Интеллекта» в таких дисплеях было мало, и практически никакой обработки информации они не делали. Такой терминал представляет собой монитор с клавиатурой без жесткого диска, процессора и других компьютерных компонентов. Таким образом, раньше использовался в основном принцип *централизованной обработки данных*. Основная цель организации связи состояла в том, чтобы разделить «интеллект» большой мощности и дорогой ЭВМ между пользователями, работающими за этими терминалами. При этом достигалось совместное использование вычислительных ресурсов, самых дорогих в то время. Этот режим работы получил название *режима разделения времени*, так как большая ЭВМ последовательно во времени решала задачи множества пользователей (**рис. 4.1**).



**Рис. 4.1.** Система централизованной обработки данных

Однако принцип централизованной обработки данных не отвечал требованиям надежности процесса обработки, затруднял развитие систем и не мог обеспечить необходимые временные параметры при диалоговой обработке данных в многопользовательском режиме. Возникло логически обоснованное требование перехода от использования отдельных ЭВМ в системах централизованной обработки данных к распределенной обработке данных.

*Распределенная обработка данных – обработка данных, выполняемая на независимых, но связанных между собой компьютерах, представляющих рас-*



*предделенную систему*. Самая простая сеть состоит как минимум из двух компьютеров, соединенных друг с другом, например, кабелем. Это позволяет им использовать данные совместно. Все сети независимо от их сложности основываются именно на этом простом принципе.

Когда появились микропроцессоры и первые микроЭВМ, возникла возможность разместить компьютер на столе у каждого пользователя, так как вычислительные ресурсы резко подешевели. Но зато все остальные ресурсы оставались еще довольно дорогими. А что значит голый интеллект без средств хранения информации и ее документирования? Не будешь же каждый раз после включения питания заново набирать выполняемую программу или хранить ее в маловместительном постоянном запоминающем устройстве (ПЗУ). И тут на помощь снова пришли средства связи. Связав несколько микроЭВМ, можно было организовать совместное использование ими компьютерной периферии. При этом вся обработка информации проводилась на месте, но ее результаты передавались на централизованные ресурсы. Этот режим получил название *режим об-ратного разделения времени* (рис. 4.2). Как и в первом случае, средства связи снижали стоимость всей компьютерной системы в целом.

Затем появились персональные компьютеры, которые отличались от первых микрокомпьютеров тем, что имели комплект достаточно развитой для пол-



*Рис. 4.2. Объединение в сеть первых микрокомпьютеров*

ностью автономной работы периферии: магнитные диски, принтеры, мыши и т.п. Периферия подешевела и стала по цене вполне сравнимой с компьютером. Зачем опять соединять персональные компьютеры? Интеллекта на месте хватает, периферии тоже. Что же может дать сеть в этом случае?

Самое главное, как уже отмечалось, – это совместное использование ресурсов. Это то же самое обратное разделение времени, но уже на другом уровне. Здесь имеется возможность более эффективно использовать ресурсы всех имеющихся в распоряжении компьютеров. Так, нередко случается, что на одном компьютере емкости своего диска не хватает, а на другом – избыток свободного места. Еще одна ситуация: все пользователи работают с одной программой, соответственно, эта программа хранится на диске каждого из них. При использовании же сети она может находиться только в одном месте, но все будут иметь к ней столь же удобный доступ, как если бы она была на их компьютере. Таким

образом, *основное назначение компьютерной сети – это совместное использование ресурсов и осуществление интерактивной связи как внутри одной фирмы, так и за ее пределами.* Отметим, что разделять с помощью локальных сетей можно не только диски и принтеры, но и другие ресурсы, например возможность выхода в глобальную сеть. Если бы локальной сети не было, пришлось бы оснащать соответствующей аппаратурой каждый компьютер, а с локальной сетью достаточно подключить к глобальной сети только один из них, а все остальные уже получают возможность доступа к ней автоматически.

В последние годы разгораются страсти по поводу сетевого компьютера NC (Network Computer). Смысл этого предложения заключается в том, чтобы разгрузить персональный компьютер от дисковой подсистемы и заменить его коммуникационной. Такой компьютер должен быть сетью связан с компьютером NC, с которого загружаются ОС, приложения и даже пользовательские документы. Такое решение, по мнению разработчиков, удешевит сам компьютер и его эксплуатацию, т.к. все заботы о модернизации программного обеспечения и его сопровождения должны брать на себя хозяева компьютеров NC, обеспечивая беззаботное существование пользователя.

Заметим, что сетевой компьютер NC предполагает наличие высокопроизводительных каналов связи пользовательских компьютеров с некоторыми вычислительными устройствами, которых у нас пока нет. На сегодняшний день для большинства пользователей лучше использовать полноценный компьютер, и выбор используемых коммуникационных средств должен производиться исходя из решаемых задач и существующей коммуникационной инфраструктуры.

*Таким образом, компьютерной сетью будем называть группу соединенных компьютеров и других устройств в единую систему, удовлетворяющую требованиям распределенной обработки данных, взаимодействующих между собой с помощью средств связи.*

Условно по размерам и числу пользователей компьютерные сети можно разделить на следующие классы.

*Локальные компьютерные сети (LAN – Local Area Network)* представляют собой системы распределенной обработки данных, которые охватывают небольшие территории внутри отдельных цехов, предприятий, офисов и обеспечивают связь на небольших расстояниях (обычно до 10 км, чаще до 1 км). Локальные сети предназначены для организации раздельного использования общих (база данных и пр.) или дорогостоящих (принтеров, устройств резервного копирования и пр.) ресурсов.

Локальная сеть может объединяться в более крупномасштабные образования:

- *CAN (Campus Area Network)* – компьютерная сеть, объединяющая группу близко расположенных зданий;
- *MAN (Metropolitan Area Network)* – сеть городского масштаба;
- *WAN (Wide Area Network)* – широкомасштабная сеть;
- *GAN (Global Area Network)* – глобальная сеть;
- и наконец, *Internet*, которую называют «сетью сетей».

### **4.1.2. Основные программные и аппаратные компоненты сети**

Компьютерная сеть – это сложный комплекс взаимосвязанных и согласованно функционирующих программных и аппаратных компонентов. Компонентами сети являются:

- компьютеры;
- коммуникационное оборудование;
- операционные системы;
- сетевые приложения.

Весь комплекс программно-аппаратных средств сети может быть описан многослойной моделью.

В основе любой сети лежит аппаратный слой стандартизированных компьютерных платформ. В настоящее время в промышленных сетях широко применяются компьютеры различных классов – от персональных до мэйнфреймов и суперЭВМ.

Второй слой – это коммуникационное оборудование, которое играет не менее важную роль, чем компьютеры. Изучение принципов работы коммуникационного оборудования требует знакомства с протоколами, используемыми как в локальных, так и в глобальных сетях.

Третий слой – это операционные системы. От того, какие концепции управления локальными и распределенными ресурсами положены в основу сетевой ОС, зависит эффективность работы всей сети.

Самым верхним слоем сетевых средств являются различные сетевые приложения, такие как сетевые базы данных, системы автоматизации коллективной работы и др.

### **4.1.3. Характеристики коммуникационной сети**

Главным требованием, предъявляемым к сетям, является выполнение сетью ее основной функции – обеспечение пользователям потенциальной возможности доступа к разделяемым ресурсам всех компьютеров, объединенных в сеть. Все остальные характеристики связаны с качеством выполнения этой основной задачи. Для оценки качества сети можно использовать следующие характеристики:

- скорость передачи данных по каналу связи;
- пропускная способность канала связи;
- достоверность передачи информации;
- надежность канала связи.

*Скорость передачи* данных по каналу связи измеряется количеством битов информации, передаваемых за единицу времени – секунду. Скорость передачи данных зависит от типа и качества канала связи, типа используемых модемов и принятого способа синхронизации. Для пользователей вычислительных сетей значение имеют не абстрактные биты в секунду, а информация, единицей измерения которой служат байты или знаки.

Поэтому более удобной характеристикой канала является его *пропускная способность*, которая оценивается количеством знаков, передаваемых по каналу за единицу времени. При этом в состав сообщения включаются и все служебные символы. Теоретическая пропускная способность определяется скоростью передачи данных. Реальная пропускная способность зависит от ряда факторов, среди которых и способ передачи, и качество канала связи, и условия его эксплуатации, и структура сообщений.

*Достоверность передачи информации* оценивают как отношение количества ошибочно переданных знаков к общему числу переданных знаков. Таким образом, единицей измерения достоверности является количество ошибок на знак, т.е. ошибок/знак. Обычно допускается не более одной ошибки на миллион или десять миллионов переданных знаков.

*Надежность* сетей определяется либо долей времени исправного состояния в общем времени работы, либо средним временем безотказной работы. Для сетей среднее время безотказной работы должно составлять, как минимум, несколько тысяч часов.

#### 4.1.4. Адресация компьютеров

Одной из основных проблем, которую нужно учитывать при объединении трех и более компьютеров, является проблема адресации. На адресации компьютеров мы остановимся подробнее позднее, а пока отметим, что наибольшее распространение получили три схемы адресации.

**Аппаратные адреса (MAC-адреса).** Типичным представителем адреса такого типа, как мы увидим в дальнейшем, является адрес сетевого адаптера локальной сети. Такой адрес обычно используется только аппаратурой и записывается в виде двоичного или шестнадцатеричного значения, например 00–A0–17–5E–24–01, занимающего обычно 6 байт.

**Символьные адреса или имена (доменные адреса).** Эти адреса предназначены для запоминания людьми, поэтому они обычно несут смысловую нагрузку. Символьные адреса легко использовать как в небольших, так и крупных сетях. Для работы в больших сетях символьное имя может иметь сложную, иерархическую структуру, например <http://www.primaluda.ural.ru>.

**Числовые составные адреса.** Символьные имена удобны для людей, но из-за переменного формата и потенциально большой длины их передача по сети не очень экономична. В связи с этим во многих случаях для работы в больших сетях в качестве адресов используют числовые составные адреса фиксированного и компактного форматов. Типичным представителем адресов этого типа являются так называемые IP-адреса. В них поддерживается двухуровневая иерархия, адрес делится на старшую часть – номер сети и младшую – номер компьютера (узла). Например, 142.131.34.122, где 142.131 – номер сети, 34.122 – номер компьютера (узла). Такое деление позволяет передавать сообщения между сетями только на основании номера сети, а номер узла используется только после доставки сообщения в нужную сеть; точно так же, как название улицы почтальон использует только после того, как письмо доставлено в нужный город.

В современных сетях для адресации узлов применяются, как правило, одновременно все три приведенные схемы. Пользователи адресуют компьютеры символьными именами, которые автоматически заменяются в сообщениях, передаваемых по сетям, на числовые номера. С помощью этих числовых номеров сообщения передаются из одной сети в другую, а после доставки сообщений в сеть назначения вместо числового номера используется аппаратный адрес компьютера. Сегодня такая схема, как мы увидим в дальнейшем, характерна даже для небольших автономных сетей, где, казалось бы, она явно избыточна – это делается для того, чтобы при включении этой сети в большую сеть не нужно было менять состав операционной системы и схему адресации.

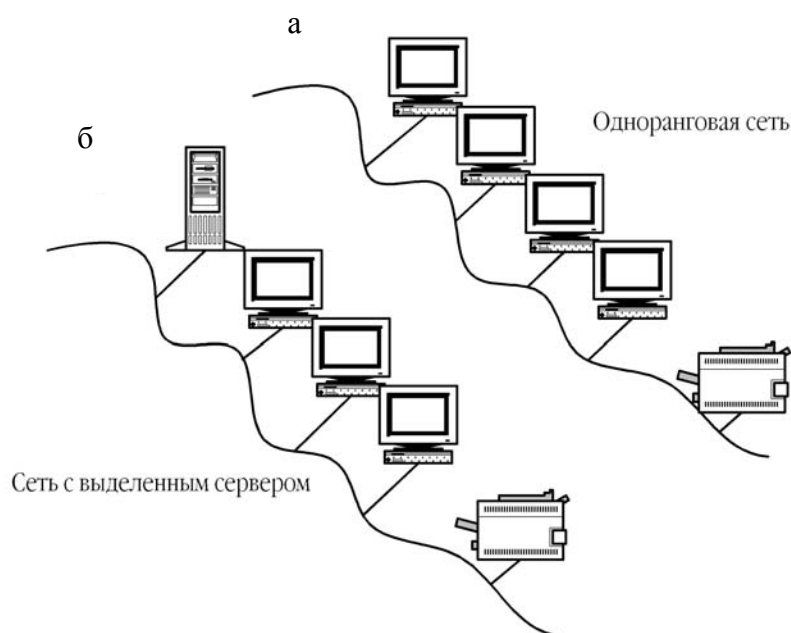
## 4.2. Особенности локальных компьютерных сетей

### 4.2.1. Два типа сетей

Локальные сети разделяются на два типа (рис. 4.3):

- одноранговые;
- с выделенным сервером.

**Одноранговые сети.** В такой сети нет единого центра управления взаимодействием компьютеров и нет единого устройства для хранения данных. В



**Рис. 4.3.** Простейшие примеры типов сетей: а – одноранговая; б – с выделенным сервером

одноранговой сети все компьютеры равноправны: среди них нет иерархии. Все пользователи самостоятельно решают, какие данные на своем компьютере сделать общедоступными по сети. Одноранговые сети называют также рабочими группами. В одноранговых сетях требования к производительности и к уровню защиты для сетевого программного обеспечения, как правило, ниже. В боль-

шинство операционных систем, например Microsoft Windows NT, Microsoft Windows 95/98/2000, встроена поддержка одноранговых сетей, поэтому дополнительного программного обеспечения не требуется. В таких сетях пользователи должны обладать достаточным уровнем знаний, чтобы работать и как пользователи, и как администраторы своего компьютера, в них возникают проблемы с защитой данных, обеспечением доступа к ресурсам и т.п.

**Сети на основе сервера (на выделенном сервере).** Выделенным называют такой компьютер, который функционирует как сервер (исключая функции клиента или рабочей станции). В сети с выделенным сервером один из компьютеров (сервер) выполняет функции хранения данных, предназначенных для использования всеми компьютерами, управления взаимодействием между ними. Они специально настроены для быстрой обработки запросов от сетевых клиентов и для управления защитой файлов и каталогов. *Сети на основе сервера стали промышленным стандартом.* Функции сервера бывают различными: контроллер домена, сервер приложений, доступ в Internet через Internet Information Server, Web-сервер и др. Остановимся на их функциях позднее. Сетевой сервер и операционная система работают как единое целое. Некоторые ОС (Microsoft Windows NT Server, Novell Netware) были созданы специально для того, чтобы использовать преимущества наиболее передовых серверных технологий. Сервер спроектирован так, чтобы предоставлять доступ к множеству файлов и принтеров, обеспечивая при этом высокую производительность и защиту. Администрирование и управление доступом к данным осуществляется централизованно, ресурсы, как правило, расположены также централизованно. Основным аргументом при выборе сети на основе сервера является, как правило, защита данных. В таких сетях, например с операционной системой Microsoft Windows NT Server, проблемами безопасности занимается один администратор, т.е. осуществляется централизованное администрирование. Поскольку жизненно важная информация расположена централизованно, т.е. сосредоточена на одном или нескольких серверах, нетрудно обеспечить ее регулярное копирование. Сети такого типа способны поддерживать тысячи пользователей. Так как компьютер пользователя не выполняет функции сервера, требования к его характеристикам зависят от потребностей самого пользователя.

#### **4.2.2. Топология сети**

Термин «топология» характеризует физическое расположение компьютеров, кабелей и других компонентов сети, т.е. другими словами «форму» или «очертание» сети. Топология сети обуславливает ее характеристики и влияет на состав необходимого сетевого оборудования, характеристики сетевого оборудования и способ управления сетью.

Все сети строятся на основе трех *базовых топологий*, к числу которых относятся шина, звезда, кольцо. Хотя сами по себе базовые топологии несложны, в реальности часто встречаются довольно сложные комбинации, объединяющие свойства нескольких топологий.

Заметим, что при компоновке сетей выделяют два типа соединений. Соединением типа «точка-к-точке» называется соединение между двумя и только

двумя сетевыми устройствами (компьютерами, принтерами и т.п.) с использованием одного кабеля. При этом никакие промежуточные ответвления соединительного кабеля недопустимы. Многоточечным соединением называется соединение, при котором один кабель используется для соединения более двух сетевых устройств. Такой кабель имеет несколько промежуточных ответвлений.

### Шина

Наиболее дешевой схемой организации сети является топология «шина» (рис. 4.4), предполагающая непосредственное подключение всех компьютеров к одному сетевому кабелю. Топологию «шина» часто называют «линейной ши-

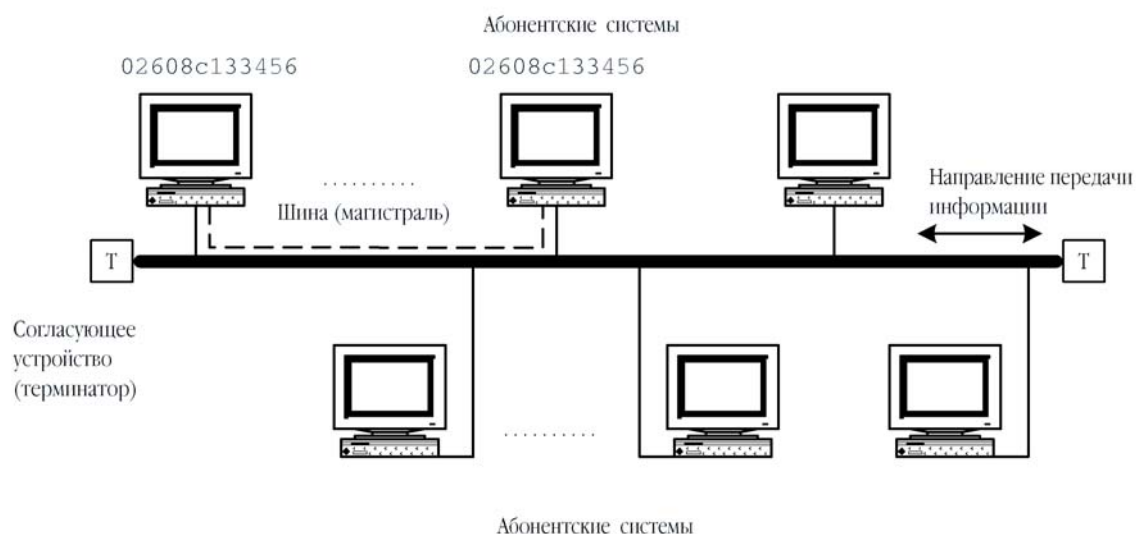


Рис. 4.4. Шинная топология сети

ной». Данная топология относится к наиболее простым и широко распространенным топологиям. В ней используется один кабель (магистраль), вдоль которого подключены все компьютеры сети. Тип соединения – многоточечный. В сети компьютеры адресуют данные конкретному компьютеру, передавая их по кабелю в виде электрических сигналов, однако информацию принимает только тот, адрес которого соответствует адресу получателя, зашифрованному в этих сигналах. Причем в каждый момент времени только один компьютер может вести передачу.

Так как данные в сеть передаются лишь одним компьютером, ее производительность зависит от количества компьютеров, подключенных к шине. Чем больше компьютеров, ожидающих передачи данных, тем медленнее работает сеть. Кроме числа компьютеров, на быстродействие сети влияют множество факторов:

- расстояние между компьютерами;
- тип сетевого кабеля;
- характеристики аппаратного обеспечения компьютеров в сети и др.

Шина, как правило, – *пассивная технология*. Это означает, что компьютеры только «слушают» передаваемые по сети данные, но не перемещают их от отправителя к получателю, поэтому, если один из компьютеров выйдет из строя,

это не скажется на работе остальных. В активных топологиях, как мы увидим далее, компьютеры регенерируют сигналы и передают их по сети.

Данные, или электрические сигналы, распространяются по всей сети – от одного конца кабеля к другому. Если не принимать никаких мер, то сигнал, достигая конца кабеля, будет отражаться и не позволит другим компьютерам осуществить передачу. Чтобы предотвратить отражение электрических сигналов, на каждом конце кабеля устанавливают *терминаторы (развязки)*, поглощающие эти сигналы. Разрыв сетевого кабеля приводит к отражению электрических сигналов в кабеле и прекращению функционирования сети.

Применение общей шины снижает стоимость проводов, унифицирует подключение различных модулей, обеспечивает возможность практически мгновенного широковещательного обращения ко всем станциям сети. Основным недостатком топологии шина является ее низкая надежность и высокая вероятность выхода из строя всей сети при возникновении неисправности на любом участке сетевого кабеля. Другим недостатком общей шины является ее невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные в сеть.

## Звезда

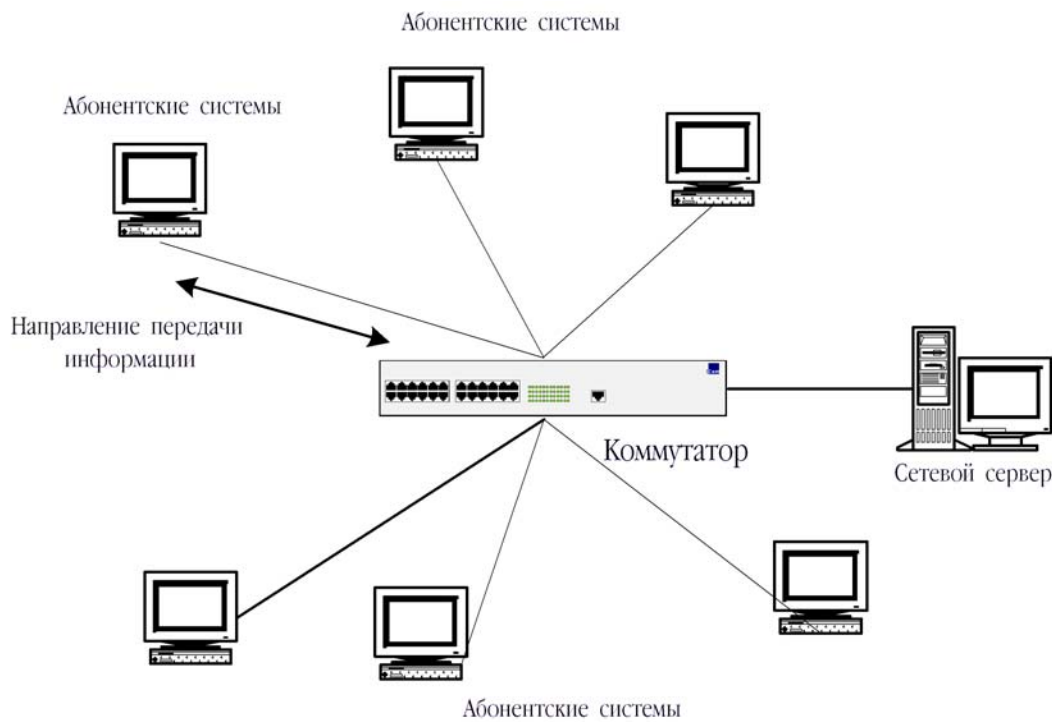
Для построения большинства сетей в настоящее время применяется топология «звезда». В данном случае недостаток, связанный с вероятностью возникновения обрыва в общем кабеле, преодолевается путем применения отдельных кабелей для подключения каждого из компьютеров к главному сетевому кабелю. Все компьютеры с помощью кабеля подключаются к центральному узлу, имеющему несколько портов и называемому *концентратором (коммутатором, хабом)* (рис. 4.5). Тип подключения «точка-к-точке». В этих сетях подключение кабеля и управление конфигурацией сети централизованное.

Название "звезда" применяется для двух различных типов топологий: "активная звезда" и "пассивная звезда". "Активная звезда" предполагает наличие в центре некоторого активного оборудования, которое создает каналы между двумя абонентами. В качестве такого активного оборудования может выступать коммутатор. "Пассивная звезда" фактически представляет собой шину, где абоненты соединены в виде звезды, в центре этой звезды нет абонента, и все кабели просто соединены между собой. Там же могут находиться повторители сигналов (концентраторы), т.е. некие пассивные преобразователи. В любом случае в центре "пассивной звезды" нет никакой обработки информации и никакого управления обменом.

*В большинстве современных звездообразных сетей функции коммутации абонентов и управления сетью или разделены между коммутатором и сетевым сервером (последний подключается к коммутатору в этом случае с максимальным приоритетом), или сервер одновременно выполняет функции и коммутатора и сервера.* В случае обрыва кабеля, соединяющего концентратор с рабочей станцией, связь с сетью утратит только эта станция. Остальные же смогут беспрепятственно продолжать работу в сети. Однако в случае выхода из строя концентратора связь с сетью теряют все подключенные к нему рабочие станции.



Конечно, такого рода неисправность выявляется довольно легко, поскольку все пользователи, подключенные к этому концентратору, немедленно обратят внимание администратора сети на возникшую проблему. В отличие от топологии шины, где на отыскание неполадки уходит большое количество сил и времени, вышедший из строя концентратор буквально заявляет о себе сам.



**Рис. 4.5.** Структура активной звездообразной сети с распределенным управлением

Этот вид топологии сети имеет некоторые недостатки. Значительно увеличивается расход кабеля. Если центральный компонент выйдет из строя, нарушается работа всей сети. Однако если выйдет из строя только один компьютер или кабель, соединяющий его с концентратором, то лишь этот компьютер не сможет передавать или принимать данные по сети.

### Кольцо

Компьютеры подключаются к кабелю, замкнутому в кольцо (рис. 4.6), поэтому у кабеля просто не может быть свободного конца, к которому надо подключить терминатор. Сигналы передаются по кольцу в *одном направлении* и проходят через каждый компьютер. В отличие от пассивной топологии «шина», здесь каждый компьютер усиливает сигналы и передает их следующему компьютеру. Поэтому, если выйдет из строя один компьютер, прекращает функционировать вся сеть. Эта топология обычно допускает значительное количество абонентов (1000 и более). В сетях с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прервался канал связи между остальными станциями.

Рассмотренные выше сетевые топологии являются базовыми, на основании которых формируется конкретная структура реальной сети. Следует обратить внимание, что звездообразные, шинные и кольцевые сетевые топологии были характерны для первых локальных компьютерных систем. Современные высокоскоростные локальные компьютерные сети широко используют и другие

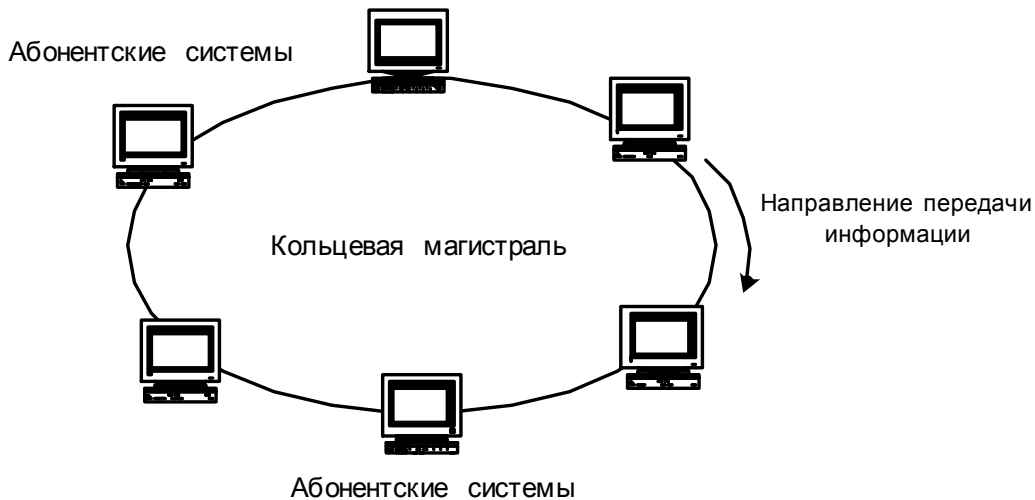


Рис. 4.6. Кольцевая топология сети

топологии, в частности древовидную топологию. Заметим, что современный высокоскоростной вариант магистральной сети Fast Ethernet имеет древовидную структуру.

### 4.2.3. Физическая передающая среда локальных компьютерных сетей

Физическая передающая среда обеспечивает перенос информации между абонентами компьютерной сети.

#### Использование соединительных кабелей

Кабель состоит из проводников, заключенных в один или несколько слоев изоляции. Кроме того, кабель может быть оснащен разъемами, позволяющими быстро выполнять присоединение к нему различного оборудования. В компьютерных сетях применяются три основных типа кабеля: кабели на основе витой пары, коаксиальные кабели с медной жилой, а также волоконно-оптические кабели.

- **Витая пара (twisted pair)** – это скрученная пара медных проводов. Скручивание проводов уменьшает влияние внешних электромагнитных полей на передаваемые сигналы. Витые пары имеют различные характеристики, определяемые размерами, изоляцией и шагом скручивания.

В настоящее время используется два типа витых пар.

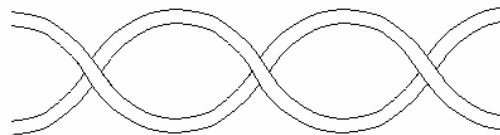


Рис. 4.7. Витая пара проводов

- **Неэкранированная витая пара (Unshielded Twisted Pair, UTP).** Этот кабель состоит из проводов в диэлектрической изоляции, скрученных между собой (рис.4.7).

- **Экранированная витая пара (Shielded Twisted Pair, STP).** Ее отличие от неэкранированной витой пары состоит в том, что каждая из пар проводов заключена в экранирующую оплетку и, кроме того, все витые пары проводов помещены в общую экранирующую оплетку. Ее следует применять при наличии сильных электромагнитных помех.

- **Коаксиальный кабель (coaxial).** Этот кабель практически аналогичен телевизионному антенному кабелю. Он состоит из центрального медного провода и изоляции, окруженных двумя слоями: экранирующей оплетки и наружным изоляционным слоем (рис. 4.8).

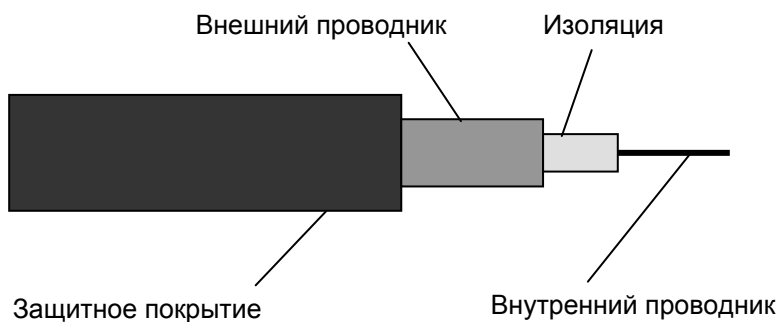


Рис. 4.8. Коаксиальный кабель

Коаксиальный кабель по сравнению с витой парой обладает более высокой механической прочностью, помехозащищенностью и обеспечивает скорость передачи до 50 Мбит/с. Для промышленного использования выпускаются два типа коаксиальных кабелей: толстый (thick) диаметром 10 мм и тонкий (thin) диаметром 5 мм.

- **Волоконно-оптический кабель (optical fiber).** В оптоволоконном кабеле цифровые данные распространяются по оптическим тонким (5–62,5 микрон) волокнам в виде модулированных световых импульсов. По этому кабелю свет передается по тонкому стекловолкну, заключенному в оболочку, которая имеет значительно меньший коэффициент преломления (рис. 4.9), что приводит к эффекту полного внутреннего отражения, так как свет проходит по кабелю, не выходя наружу. Это относительно надежный (защищенный) способ передачи,

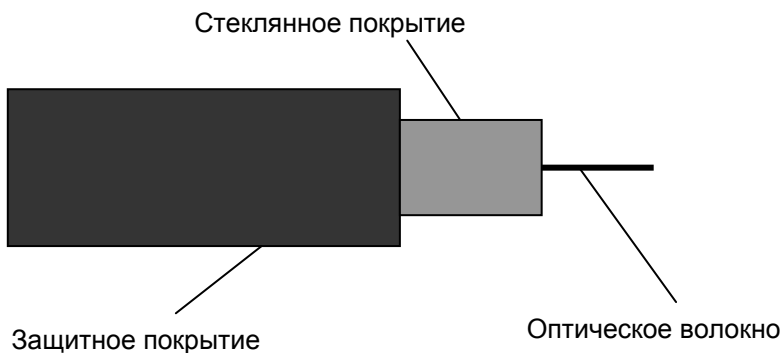


Рис. 4.9. Оптоволоконный кабель

поскольку электрические сигналы не передаются. Оптоволоконные линии предназначены для перемещения больших объемов данных на очень высоких скоростях, так как сигнал в них практически не затухает и не искажается. По нему можно передавать световые импульсы на многие километры. Каждое стеклянное оптоволокно передает сигнал только в одном направлении, поэтому кабель состоит из двух волокон: одно служит для приема, другое – для передачи. Оптоволоконный кабель – идеальная передающая среда, которая не подвержена действию электромагнитных полей и сама практически не имеет излучения. Последнее позволяет использовать этот кабель в сетях, требующих повышенной секретности информации. Скорость передачи данных более 10 Гбит/с. Так как сигнал по волоконно-оптическому кабелю передается не электрический, а световой – на передающем конце и обратное преобразование – на приемном конце. Это, конечно, увеличивает стоимость аппаратуры. Главный недостаток волоконно-оптического кабеля – это значительная сложность его монтажа, так как при соединении кабеля и разъема необходима микронная точность.

### *Беспроводные сети*

Зачастую единственным и наиболее приемлемым способом решения задачи построения территориально-распределенных сетей передачи данных является применение цифровых радиосредств. Именно с использованием радиосредств создаются компьютерные сети крупных металлургических предприятий. При этом беспроводные компоненты взаимодействуют с сетью, в которой в качестве среды передачи используется кабель. Такая сеть со смешанными компонентами называется гибридной. Типичная беспроводная сеть выглядит и функционирует практически так же, как и обычная, за исключением среды передачи.

Кратко касаясь существующих сегодня технологий беспроводной связи, можно выделить:

- передачу данных с помощью инфракрасных лучей. Она обеспечивает передачу данных со скоростью сетей, однако дальность связи при этом ограничена прямой видимостью;
- передачу данных с помощью лазера;
- радиопередачу в узком спектре (одночастотная передача);
- радиопередачу в рассеянном спектре.

Среди преимуществ в построении сетей передачи данных с применением радиосредств следует выделить:

- быстроту развертывания, что критично при высоких требованиях к скорости построения сети;
- отказ от дорогостоящих работ по прокладке кабеля;
- отказ от дорогостоящей аренды уже существующих каналов связи;
- возможность подключения мобильных объектов;
- гибкость архитектуры сети, т.е. мобильность оборудования, а вместе с тем и самого канала.

Радиоканал может поддерживать связь на десятки километров и удобен, прежде всего, тем, что абоненты ничем не связаны друг с другом, т.е. они могут легко изменять свое местоположение.

К недостаткам беспроводных сетей следует отнести более низкую скорость передачи по сравнению с проводными сетями, немалые вложения при организации каналов связи по сравнению с проводными, влияние таких факторов, как необходимые метео- и геоусловия, и, конечно, необходимость проведения работ по легализации сети в целом или даже отдельного канала связи.

#### **4.2.4. Платы сетевого адаптера**

Сетевые адаптеры (контроллеры, карты) – это одна из основных частей аппаратуры локальной сети. Минимальный набор аппаратуры, которой надо оснастить компьютеры для объединения их в сеть, должен включать в себя адаптеры (по одному на каждый компьютер) и соединительные кабели с соответствующими разъемами. Все остальное оборудование сети служит для улучшения ее характеристик, а также для повышения удобства ее использования.

Плата сетевого адаптера состоит из аппаратной части и встроенных программ, записанных в ПЗУ. Каждый тип кабеля имеет различные физические характеристики, которым должна соответствовать сетевая плата.

Основные сетевые функции адаптера

1. Гальваническая развязка компьютера и локальной сети. Эта функция не является обязательной. При некоторых типах среды передачи (оптоволоконный кабель, радиоканал) развязка не нужна.

2. Преобразование уровней сигналов из логических (0,1) в сетевые при передаче и из сетевых в логические при приеме.

3. Кодирование сигналов при передаче и декодирование при приеме.

4. Распознавание своего пакета при приеме.

5. Преобразование параллельного кода в последовательный при передаче и последовательного кода в параллельный при приеме.

6. Буферирование передаваемых и принимаемых данных в буферном оперативном запоминающем устройстве.

7. Контроль состояния сети, разрешение конфликтов (проведение арбитража).

Первые четыре функции всегда реализуются аппаратно, хотя третья и четвертая в принципе могут выполняться программно.

Перед тем как послать данные в сеть, плата сетевого адаптера должна перевести их из формы, понятной компьютеру, в форму, в которой они могут передаваться по сетевому кабелю. Внутри компьютера данные передаются по шинам. Это, как правило, несколько проводников, расположенных близко друг к другу. Так как линий несколько, то и биты данных могут передаваться по ним группами, а не последовательно. Современные компьютеры оснащены 32-разрядной шиной, т.е. 32 бита движутся параллельно друг друга (32-полосная автострада, по которой рядом движутся 32 машины, каждая из которых перевозит бит). В сетевом же кабеле данные должны перемещаться в виде потоков битов. Иными словами, кабель – это дорога с одной полосой, по таким дорогам дан-

ные в каждый момент времени движутся в одном направлении. Следовательно, информация в локальных сетях передается в последовательном коде, бит за битом. Почему бы не использовать здесь параллельную передачу, ведь она гораздо быстрее, к тому же не требует преобразования параллельного кода компьютера в последовательный, и наоборот? Прежде всего, это связано с тем, что при параллельной передаче увеличивается количество соединительных проводов в число раз, равное количеству разрядов параллельного кода (проложить один кабель гораздо проще, чем 32 или 64). Значительно дороже при параллельной передаче обойдется также поиск повреждений и ремонт кабеля. Передача при любом типе кабеля требует сложной передающей и приемной аппаратуры: надо формировать мощный сигнал на передающем конце и детектировать слабый сигнал на приемном конце. При последовательной передаче для этого требуется всего один передатчик и один приемник. При параллельной же передаче количество передатчиков и приемников возрастает пропорционально разрядности используемого параллельного кода.

Таким образом, плата сетевого адаптера принимает параллельные данные и организует их для последовательной побитовой передачи. Этот процесс завершается переводом цифровых данных компьютера в электрические и оптические сигналы, которые и передаются по сетевым кабелям.

Помимо преобразования данных, плата сетевого адаптера должна указать свое местонахождение, или адрес, чтобы ее могли отличить от остальных плат. *Каждая плата имеет свой уникальный адрес*, размер которого 6 байт, например 00–A0–17–5E–24–01. *Сетевые адреса* определены комитетом IEEE (Institute of Electrical and Electronics Engineers, Inc. – Организация, объединяющая специалистов в области инженерных разработок и электроники). Этот комитет закрепляет за каждым производителем плат сетевого адаптера некоторый интервал адресов. Первые три байта адреса называются префиксом и закреплены за производителем. Производители зашивают эти адреса в микросхемы. Благодаря этому каждая плата и, следовательно, каждый компьютер имеют уникальный адрес в сети.

Перед тем как послать данные по сети, плата сетевого адаптера проводит электронный диалог с принимающей платой, во время которого они «обговаривают»:

- максимальный размер блока передаваемых данных;
- объем данных, передаваемых без подтверждения о получении;
- интервалы между передачами блоков данных;
- интервал, в течение которого необходимо послать подтверждение;
- объем данных, который может принять каждая плата, не переполняясь;
- скорость передачи данных;
- каждая плата оповещает другую о своих параметрах, принимая чужие параметры и подстраиваясь к ним. После того как все детали определены, платы начинают обмен данными.

### 4.3. Функционирование сети

#### 4.3.1. Сетевые модели OSI и IEEE Project 802

В этом параграфе будем изучать кухню работы сети, т.е. организацию сложного взаимодействия между устройствами в сети. Как известно, для решения сложных задач используется универсальный прием – декомпозиция, то есть разбиение одной сложной задачи на несколько более простых. При декомпозиции часто применяют многоуровневый подход, который как раз и используется при взаимодействии компьютеров в сети.

Предварительно отметим, что сетевая операционная система при выполнении всех задач следует строгому набору процедур. Эти процедуры называются протоколами или правилами поведения. Стандартные протоколы позволяют программному и аппаратному обеспечению различных производителей нормально взаимодействовать.

#### *Модель OSI*

В 1984 году International Organization for Standardisation (ISO, организация, объединяющая группы стандартизации различных стран) выпустила эталонную модель взаимодействия открытых систем, которая и стала международным стандартом. Являясь многоуровневой системой, она отражает взаимодействие программного и аппаратного обеспечения при осуществлении сеанса связи.

*Открытой системой считается система, отвечающая требованиям эталонной модели взаимодействия открытых систем, реализующая стандартный набор услуг и поддерживаемая стандартными протоколами.*

Термин "открытая система" в данном случае означает, что система не замкнута в себе и может наращиваться до бесконечности, в частности, связываться с другими системами. В этой универсальной модели нет никакой привязки к конкретной аппаратуре используемых компьютеров, к аппаратуре соединяющих их сетей, к типу программного обеспечения, то есть модель OSI представляет собой некую абстрактную модель.

Соблюдение этих требований обеспечивает возможность взаимодействия открытых систем между собой, несмотря на их технические и логические различия в реализации, что является достаточно существенным фактором построения компьютерных сетей.

Если две сети построены с соблюдением принципов открытости, то это дает следующие преимущества:

- возможность построения сети из аппаратных и программных средств различных производителей, придерживающихся одного и того же стандарта;
- возможность безболезненной замены отдельных компонентов сети другими, более совершенными, что позволяет развивать сети с минимальными затратами;
- возможность легкого сопряжения одной сети с другой.

Модель OSI работает подобно современной почтовой службе. Первым шагом при отправке информации будет написание письма; затем оно должно

быть помещено в конверт с именем, адресом и почтовым индексом; потом письмо должно быть опущено в почтовый ящик. После этого уже не ваша забота, как именно письмо дойдет до получателя, – вы исходите из того, что оно как-нибудь да дойдет. Все, что вы должны были сделать, – это правильно поместить письмо в почтовый ящик. Как правило, этого достаточно, чтобы письмо дошло до адресата. Если часть процесса должна быть изменена, чтобы применить новую технологию или улучшить реализацию процесса (например, заменить доставку почты автотранспортом на доставку самолетом), изменения должны быть произведены только на непосредственно касающемся изменении участке, не всей системы в целом.

Перейдем к рассмотрению многоуровневой архитектуры OSI.

### Многоуровневая архитектура

В модели OSI сетевые функции распределены между *семью уровнями*.

Каждому уровню соответствуют различные сетевые операции, оборудование и протоколы.

На **рис. 4.10** представлена многоуровневая архитектура модели OSI. На каждом уровне выполняются определенные сетевые функции, которые взаимодействуют с функциями соседних уровней, вышележащего и нижележащего. Например, сеансовый уровень должен взаимодействовать только с представительным и транспортным уровнем. Каждый уровень предоставляет несколько услуг (т.е. выполняет несколько операций), подготавливающих данные для доставки по сети на другой компьютер. Все запросы от одного уровня к другому передаются через интерфейс. Каждый уровень использует услуги нижележащего уровня.

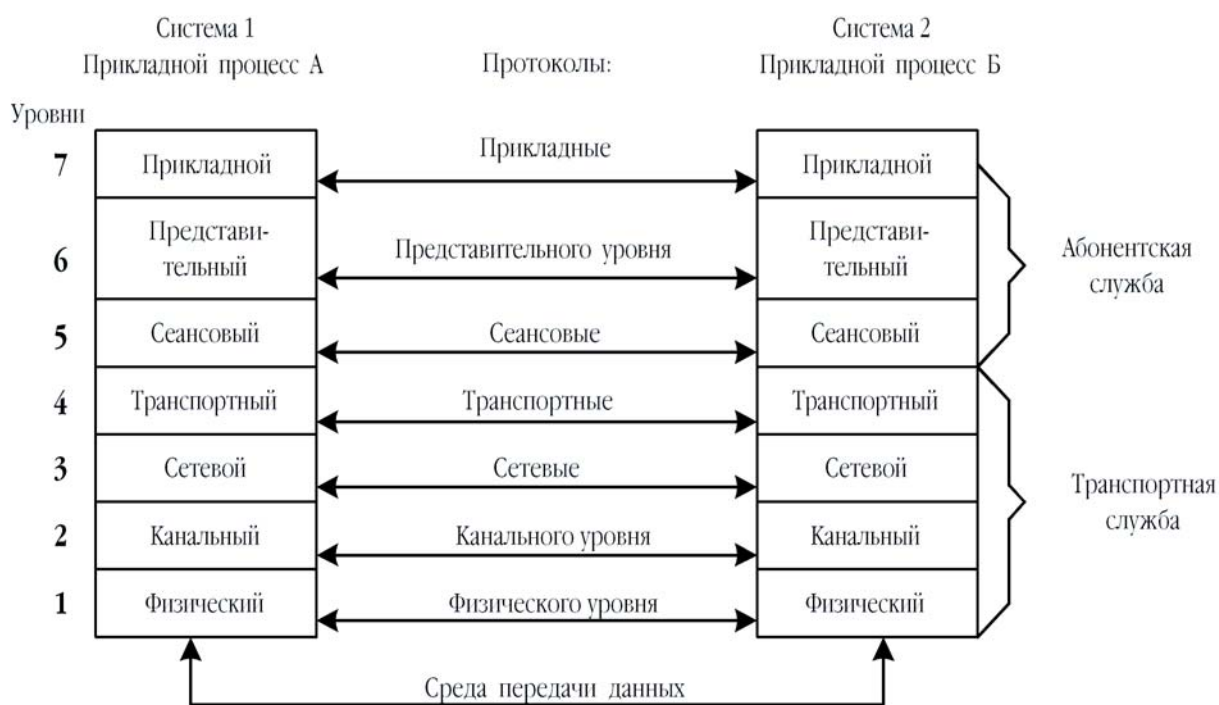


Рис. 4.10. Эталонная модель взаимодействия открытых систем



### *Взаимодействие уровней моделей OSI*

Каждый уровень на одном компьютере работает так, как будто он напрямую связан с таким же уровнем на другом компьютере. Эта логическая, или *виртуальная связь* между одинаковыми уровнями показана на **рис. 4.10**. Однако в действительности связь осуществляется между смежными уровнями одного компьютера – программное обеспечение, работающее на каждом уровне, реализует определенные сетевые функции в соответствии с набором протоколов.

Информация на компьютере-отправителе должна пройти через все уровни. Затем она передается по сетевому кабелю на компьютер-получатель и опять проходит сквозь все уровни, пока не достигнет того же уровня, с которого она была послана на компьютере-отправителе.

Перед подачей в сеть данные разбиваются на пакеты. Пакет – это единица информации, передаваемая между устройствами сети как единое целое. Пакет проходит последовательно через все уровни. В дальнейшем остановимся на процедуре формирования пакета подробнее, а пока отметим, что на каждом уровне к пакету добавляется некоторая информация, которая необходима для успешной передачи данных по сети. На принимающей стороне пакет проходит через все уровни в обратном порядке. Программное обеспечение на каждом уровне читает информацию пакета, затем удаляет информацию, добавленную к пакету на этом же уровне отправляющей стороной, и передает пакет следующему уровню. Когда пакет дойдет до прикладного уровня, вся адресная информация будет удалена и данные примут свой первоначальный вид.

Таким образом, за исключением самого нижнего уровня сетевой модели, никакой иной уровень не может непосредственно послать информацию соответствующему уровню другого компьютера. Данная модель напоминает собой вложенные друг в друга матрешки. Самая маленькая из них – это и есть пользовательские данные, а все остальные служат для доставки данных в точку назначения. Иными словами, в результате этого механизма каждый пакет более высокого уровня вкладывается в конверт протокола более низкого уровня, и сами протоколы отвечают за целостность и доставку данных.

Взаимодействие смежных уровней осуществляется через интерфейс. Интерфейс определяет услуги, которые нижний уровень предоставляет верхнему уровню, а также способ доступа к ним. В связи с этим каждому уровню одного компьютера «кажется», что он непосредственно взаимодействует с таким же уровнем другого компьютер.

Рассмотрим подробнее назначение каждого из уровней в отдельности. Предварительно заметим, что блоки, находящиеся наверху иерархии OSI, отвечают за функции, наиболее «близкие» к пользователю или приложению, в то время как нижние блоки выполняют функции, «близкие» к физической сети или сетевым интерфейсам.

**Прикладной уровень (уровень приложений). Уровень 7.** Это самый верхний уровень модели OSI, он представляет собой окно для доступа прикладных процессов к сетевым услугам. Прикладной уровень управляет общим доступом к сети, потоком данных и обработкой ошибок. Именно с него приложения

начинают осуществлять связь по сети, используя этот интерфейс. Если вы, работая с программой Microsoft Word для Windows, сохраняете файл на сетевом диске, и в этом случае для осуществления связи используется уровень приложений модели. Прикладной уровень обеспечивает широкий набор услуг: управление терминалом, управление файлами, управление диалогом, управление сетью в целом. Дополнительные услуги – электронная почта, передача массивов сообщений, файлов и т.п.

**Представительный уровень (уровень представления данных). Уровень 6.** Определяет формат, используемый для обмена данными между сетевыми компьютерами. Этот уровень носит название переводчика, т.е. данные, поступившие с прикладного уровня, переводятся в общедоступный формат. Этот уровень отвечает также за сжатие данных для уменьшения передаваемых битов. Уровень представления принимает данные, поступающие от уровня приложений, и преобразует их в промежуточный формат. Затем он осуществляет сжатие, а также, при необходимости, кодирование данных, прежде чем передать их на следующий уровень. Получая данные от сеансового уровня, уровень представления выполняет обратные операции, преобразуя данные из промежуточного формата в формат, доступный уровню приложений, и производя их распаковку и, при необходимости, декодирование.

**Сеансовый уровень. Уровень 5.** Сеансовый уровень отвечает за установку и завершение связи между двумя компьютерами. Кроме того, этот уровень определяет, какой из компьютеров может осуществлять передачу данных и сколько она может длиться в одном направлении, прежде чем возможности осуществить передачу получит второй компьютер. Следовательно, он позволяет двум приложениям на разных компьютерах устанавливать, использовать и завершать соединение, называемое сеансом. На этом уровне выполняются такие функции, как распознавание имен и защиты, необходимые для связи двух приложений в сети. Сеансовый уровень обеспечивает синхронизацию между пользовательскими задачами посредством расстановки в потоке данных так называемых контрольных точек. Таким образом, в случае сетевой ошибки потребуются заново передавать только данные, следующие за последней контрольной точкой. Этот же уровень выполняет управление диалогом между взаимодействующими процессами, т.е. регулирует, какая из сторон осуществляет передачу, когда, как долго и т.п.

**Транспортный уровень. Уровень 4.** Транспортный уровень обеспечивает надежную передачу данных по сети. При передаче данных транспортный уровень принимающей стороны посылает сообщение транспортному уровню передающей стороны, подтверждающее факт приема данных. Если такое сообщение не будет получено, передача данных будет осуществлена повторно. Кроме того, транспортный уровень отвечает за переупаковку сообщений. Он гарантирует доставку пакетов без ошибок, в той же последовательности, без потерь и дублирования. На этом уровне сообщения переупаковываются, длинные разбиваются на несколько пакетов, а короткие объединяются в один, что увеличивает эффективность передачи пакетов по сети. Этот же уровень управляет потоком, проверяет ошибки и участвует в решении проблем, связанных с отправкой и получением пакетов. Транспортный уровень обеспечивает установление и разъе-

динение транспортных соединений, формирование блоков данных, взаимодействия сеансовых соединений с транспортным соединением, целостность блоков данных во время передачи, предоставление приоритетов в передаче блоков, передача подтверждений о принятых блоках и др.

**Сетевой уровень. Уровень 3.** Основными задачами сетевого уровня является передача данных по сети и определение оптимального маршрута для выполнения этой операции. Он отвечает за адресацию сообщений и перевод логических адресов и имен в физические адреса. Кроме того, он выполняет функции регулировщика движения и управляет маршрутизацией пакетов данных в сети. Одним словом, исходя из конкретных сетевых условий, приоритета услуги и других факторов, здесь определяется маршрут от компьютера-отправителя к компьютеру-получателю. Сетевой уровень также преобразует данные в сетевые пакеты и заключает их в оболочку. В ходе преобразования данные дополняются кодом коррекции ошибок. Если пакет слишком велик, чтобы с ним мог работать уровень передачи данных, сетевой уровень преобразует один большой пакет в несколько мелких, объем которых соответствует требованиям канала передачи. При передаче данных верхнему уровню осуществляются обратные преобразования, в ходе которых несколько малых пакетов соединяются в один большой и после этого передаются транспортному уровню.

**Канальный уровень. Уровень 2.** Внутри локальной сети доставка данных обеспечивается соответствующим канальным уровнем, а доставкой данных между сетями занимается сетевой уровень, который поддерживает возможность выбора маршрута передачи данных. Канальный уровень осуществляет передачу кадров данных от сетевого уровня к физическому. *Кадры – это логически организованная структура, в которую можно помещать данные.* По существу, кадр является еще одной оболочкой вокруг данных, которая может быть разделена на меньшие части и содержит дополнительные сетевые адреса. На **рис. 4.11** представлен простой кадр данных, где идентификатор отправителя – адрес компьютера-отправителя, а идентификатор получателя – адрес компьютера-получателя. Управляющая информация используется для маршрутизации, а также указывает на тип пакета. Данные – это собственно передаваемая информация. CRC (остаток избыточной циклической суммы или коды коррекции ошибок) – это сведения, которые помогут выявить ошибки, что, в свою очередь, гарантирует правильный прием информации. Кадры данных являются зависимыми от сети, поэтому кадр данных, передаваемый серверу NetWare, может отличаться от кадра, передаваемого серверу Windows NT, даже если оба они содержат одни и те же данные. Как и сетевой уровень, канальный уровень добавляет к данным дополнительные коды коррекции ошибок (остаток избыточной циклической суммы), гарантирующие достоверность передаваемой информации. Если обнаруживается несоответствие кода коррекции ошибок, прием кадра не производится. Это исключает возможность передачи искаженных данных верхним уровням. Если кадр не был принят, т.е. подтверждение приема отсутствует, канальный уровень передаст кадр повторно. Канальный уровень получателя проверяет наличие возможных ошибок передачи. Кадры, поврежденные при передаче, или кадры, получение которых не подтверждено, посылаются вторично.

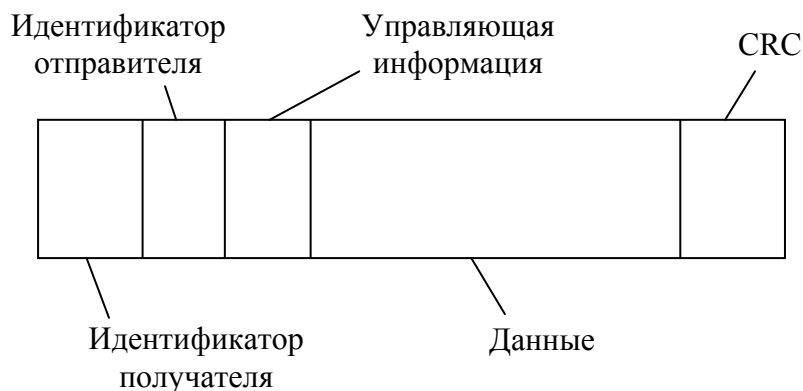


Рис. 4.11. Кадр данных

**Физический уровень. Уровень 1.** Физический уровень отвечает за передачу потока битов, поступающих от канального уровня, физической среде (сетевому кабелю). Здесь же производится кодирование и синхронизация передаваемых данных. Физический уровень также определяет характеристики среды передачи данных. Здесь реализуются электрический, оптический, механический и функциональный интерфейсы с кабелем. На этом уровне определяется способ соединения сетевого кабеля с платой сетевого адаптера, в частности количество контактов в разъемах и их функции. Кроме того, здесь определяется способ передачи данных по сетевому кабелю. Таким образом, физический уровень предназначен для передачи битов (нулей и единиц) от одного компьютера к другому.

*В чем же основное достоинство семиуровневой модели?* В процессе развития и совершенствования любой системы возникает потребность изменить ее отдельные компоненты. Здесь и проявляются преимущества семиуровневой модели. Если между уровнями определены однозначно интерфейсы, то изменение одного из уровней не влечет за собой необходимости внесения изменений в другие уровни. Таким образом, существует относительная независимость уровней друг от друга.

Как любая универсальная модель, модель OSI сильно избыточна. Она содержит в себе все возможные функции, которые в данной конкретной системе и данной локальной сети могут просто не использоваться. Более того, в конкретной сети может даже не существовать такого разделения функций на уровни, которое предусмотрено OSI, а выделить отдельные функции нередко сложно.

**Сетезависимые и сетезависимые уровни.** Функции всех уровней модели OSI могут быть отнесены к одной из двух групп: либо к функциям, зависящим от конкретной технической реализации сети, либо к функциям, ориентированным на работу с приложениями.

Три нижних уровня – *физический, канальный, сетевой* – являются *сетезависимыми*, т.е. протоколы этих уровней тесно связаны с технической реализацией сети и используемым коммуникационным оборудованием.

Три верхних уровня – *прикладной, представительный и сеансовый сетезависимые уровни* – ориентированы на приложения и мало зависят от технических особенностей построения сети. На протоколы этих уровней не влияют ка-

кие бы то ни было изменения в топологии сети, замена оборудования или переход на другую сетевую технологию.

*Транспортный уровень является промежуточным.* Он скрывает все детали функционирования нижних уровней от верхних. Это позволяет разрабатывать приложения, не зависящие от технических средств непосредственной транспортировки сообщений.

### Модель IEEE Project 802

Для того чтобы учесть требования передающей среды, используемой в локальных вычислительных сетях, была проведена некоторая модернизация семиуровневой модели взаимодействия открытых систем для локальных компьютерных сетей, и появилась модель IEEE Project 802.

IEEE (Institute of Electrical and Electronics Engineers, Институт инженеров по электронике и радиоэлектронике) национальная организация США, определяющая сетевые стандарты. Project 802 назван в соответствии с годом и месяцем своего издания (1980, февраль). Она установила стандарты для физических компонентов сети – интерфейсных плат и кабельных систем, с которыми имеют дело физический и канальный уровни модели OSI.

Стандарты семейства IEEE 802 охватывают только два нижних уровня семиуровневой модели OSI - физический и канальный. Это связано с тем, что именно эти уровни в наибольшей степени отражают специфику локальных сетей. Старше же уровни, начиная с сетевого, в значительной степени имеют общие черты как для локальных, так и для глобальных сетей.

IEEE Project 802, подробно описывая канальный уровень, разделил его на два подуровня (**рис. 4.12**):



**Рис. 4.12.** Подуровни: управление логической связью и управление доступом к среде

- LLC (Logical Link Control) – логический уровень передачи данных.
- MAC (Media Access Control) – уровень управления доступом к среде.

*Уровень MAC* появился из-за существования в локальных сетях разделяемой среды передачи данных. Именно этот уровень обеспечивает корректное совместное использование общей среды, предоставляя ее в соответствии с определенным алгоритмом в распоряжение той или иной станции сети.

*Уровень LLC* отвечает за передачу кадров данных между узлами с различной степенью надежности, а также реализует функции интерфейса с прилегающим к нему сетевым уровнем. Именно через уровень LLC сетевой протокол запрашивает у канального уровня нужную ему транспортную операцию.

### 4.3.2. Назначение и функции протоколов

Мы уже неоднократно упоминали термин «протоколы» для компьютерных сетей, конкретизируем это понятие. *Протоколы – это правила и технические процедуры, позволяющие нескольким компьютерам при объединении в сеть общаться друг с другом.* Другими словами, протокол – это набор правил, которых должны придерживаться все компании, чтобы обеспечить совместимость производимого аппаратного и программного обеспечения. Идеология открытых систем предполагает использование стандартных аппаратных средств и программного обеспечения.

Отметим три основных момента, касающихся протоколов.

1. Хотя для обмена данными между компьютерами используются различные протоколы, все сетевые коммуникационные протоколы *должны обеспечивать выполнение определенного набора основных функций.* Эти функции могут быть по-разному реализованы различными разработчиками, но они все должны иметь несколько общих основных характеристик.

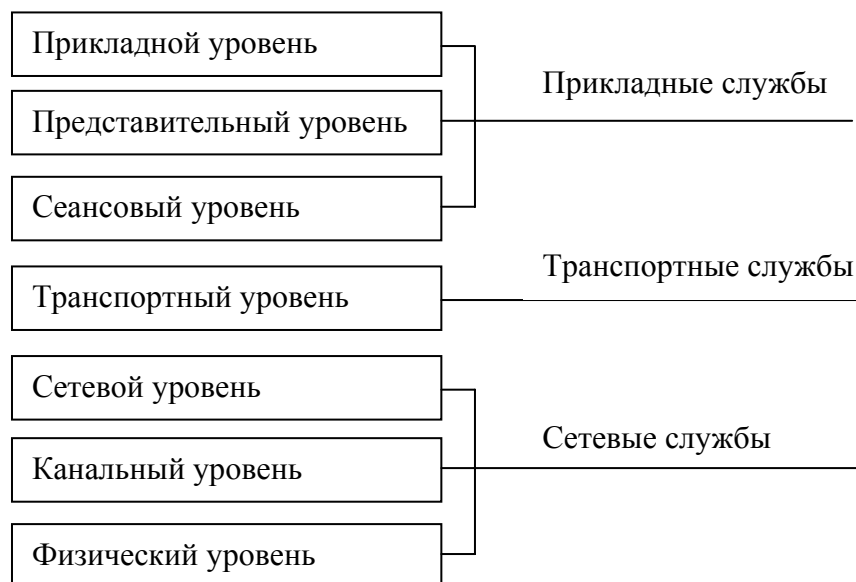
2. Существует множество протоколов. И хотя все они участвуют в реализации связи, *каждый протокол имеет различные цели, выполняет различные задачи,* обладает своими преимуществами и ограничениями.

3. *Протоколы работают на разных уровнях OSI.* Если, например, какой-то протокол работает на физическом уровне, то это означает, что он обеспечивает прохождение пакетов через плату сетевого адаптера и их поступление в сетевой кабель.

4. *Несколько протоколов могут работать совместно.* Это так называемый стек, комбинация или набор правил.

Протоколы делятся на три типа служб (**рис. 4.13**):

- *прикладные службы* (включают протоколы прикладного, представительного и сеансового уровней);
- *транспортные службы* (транспортный уровень);
- *сетевые службы* (сетевой, канальный и физический уровни).



*Рис. 4.13. Модель OSI и типы протоколов*

В компьютерной промышленности в качестве стандартных моделей протоколов разработано несколько стандартных стеков. Наиболее важные из них: TCP/IP, ISO/OSI, IPX/SPX и др.

Рассмотрим кратко наиболее распространенные протоколы.

### **4.3.3. Основные стандартные стеки коммуникационных протоколов**

#### *Набор протоколов OSI*

Набор протоколов OSI – полный стек протоколов, где каждый протокол соответствует конкретному уровню модели OSI. Следует различать модель OSI и стек OSI. В то время как модель OSI является концептуальной схемой взаимодействия открытых систем, стек OSI представляет собой набор вполне конкретных спецификаций протоколов.

Набор протоколов OSI использовался на практике, но на сегодня он обычно используется в качестве теоретического протокола для определения блоков, из которых должна состоять хорошая система, реализующая сетевой протокол. Большинство используемых в настоящее время сетевых протоколов содержат все основные функции, заданные в модели OSI, или часть их.

До того как модель OSI была признана в качестве подходящей для разработки протоколов, большинство разработчиков создавали свои собственные протоколы, мало заботясь о возможности взаимодействия с программными продук-

тами других производителей. Однако поскольку все большее и большее количество покупателей принимали идею открытых стандартов, то стандарты, являющиеся чьей-либо собственностью, стали проигрывать на рынке. Возможность взаимодействия становилась все более важной, и необходимость стандартной модели, такой как OSI, стала очевидна. Функции протокола четко разделены на отдельные группы, поэтому программисту, вместо того чтобы писать один монолитный кусок кода, предоставляется возможность создать несколько отдельных, небольших частей, каждая из которых обеспечивает выполнение определенной группы функций. Представим, что программисту удалось создать более эффективный способ отправки данных через сеть. Поскольку он придерживался модульной концепции при реализации набора протоколов, он может переписать и перекомпилировать только тот кусок кода, который обеспечивает выполнение функций канального и физического уровня. Однако если бы он не использовал модель OSI, обновить реализацию протокола было бы намного сложнее и, возможно, ему понадобилось бы переписать весь код.

Из-за своей сложности протоколы OSI требуют больших затрат вычислительной мощности центрального процессора, что делает их наиболее подходящими для мощных машин, а не для сетей персональных компьютеров.

### *Стек протоколов управления передачей данных TCP/IP (Transmission Control Protocol / Internet Protocol)*

TCP/IP (Протокол управления передачей данных/Internet протокол) – промышленный стандартный набор протоколов, которые обеспечивают связь в неоднородной среде, т.е. обеспечивают совместимость между компьютерами разных типов. Этот протокол используется для передачи данных в глобальных сетях и во многих локальных сетях. Если точнее, то можно говорить не о семействе, а о целом комплексе, стеке «протоколов Internet».

TCP/IP относится к целому семейству (стеку) связанных между собой протоколов, разработанных для передачи информации по сети и одновременного обеспечения информацией о состоянии сети, и является программным компонентом сети. Каждая часть семейства TCP/IP решает определенную задачу, например отправление электронной почты, пересылку файлов, маршрутизацию сообщений или обработку сбоев в сети и т.д.

Совместимость – главное преимущество TCP/IP, поэтому большинство локальных вычислительных сетей поддерживает этот протокол. Кроме того, он обеспечивает доступ к ресурсам Internet. Поскольку TCP/IP поддерживает маршрутизацию, он обычно используется в качестве межсетевых протоколов.

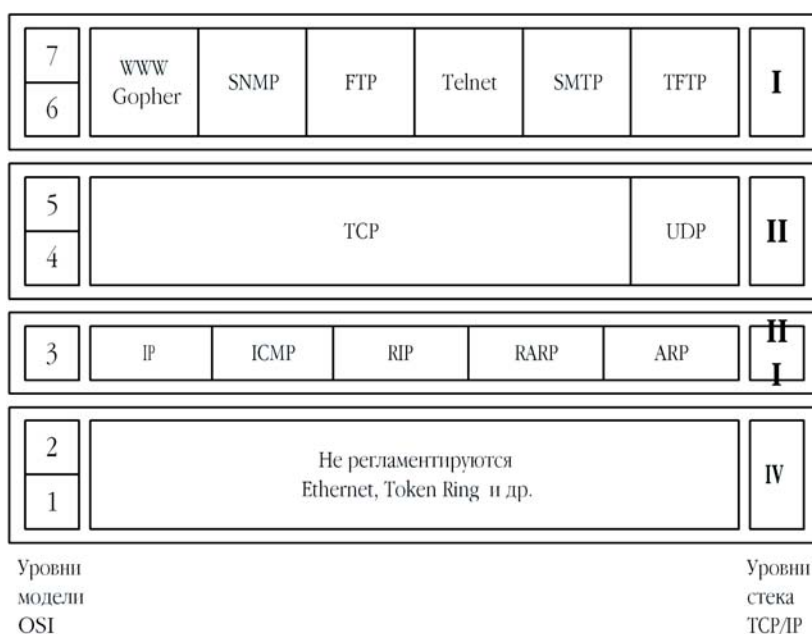
Обычно общее название TCP/IP протокола используется при обсуждении всего семейства протоколов. В действительности лучше назвать каждый из сервисов своим именем, чтобы внести большую ясность. На **рис. 4.14** приведены основные протоколы этого семейства и их соотношение с эталонной моделью взаимодействия открытых систем.

Приведем название и краткую характеристику нескольких, наиболее распространенных протоколов:



- *IP-Internet Protocol* (межсетевой протокол, протокол Internet), относящийся к сетевому (3-му) уровню по иерархии протоколов 7-уровневой модели ISO/OSI. IP-протокол считается наиболее важным во всем семействе TCP/IP, поскольку этот протокол используют все остальные. Протокол IP содержит спецификации, описывающие формат, адресацию и передачу IP-пакетов между сетями. Каждый IP-пакет, называемый дейтограммой, содержит информацию об адресах отправителя и получателя, так называемые IP-адреса, которые используются для маршрутизации пакетов данных в сети. Принципы формирования IP-адресов мы рассмотрим далее.

- *TCP – Transmission Control Protocol* (коммуникационный протокол транспортного уровня), давший название всему семейству протоколов TCP/IP. Протокол TCP гарантирует, что информационные пакеты, передаваемые с одно-



**Рис. 4.14.** Соответствие уровней стека TCP/IP семиуровневой модели OSI

го узла на другой, придут в пункт назначения в правильном порядке и содержащиеся в них данные останутся неизменными. Восстановление порядка следования пакетов – очень важный вопрос, так как в процессе маршрутизации два пакета, отправленные одним за другим, могут быть переданы, как мы покажем в дальнейшем, по разным маршрутам. Один из этих маршрутов может оказаться длиннее другого, и это приведет к тому, что первый пакет достигнет узла назначения позже второго. Если это произойдет, то TCP восстановит исходный порядок следования пакетов. TCP дополняет каждый передаваемый пакет контрольной суммой, которая проверяется на узле назначения. Все поврежденные пакеты аннулируются, и, так как с точки зрения передачи данных такая ситуация аналогична утере пакета, TCP обеспечивает повторную его передачу.

- *UDP – User Datagram Protocol* (протокол пользовательских дейтограмм) – второй по распространенности транспортный протокол. Он предоставляет прикладным процессам ограниченный набор транспортных услуг, обеспечивая надежную доставку дейтограмм. В отличие от UDP, протокол TCP обеспечивает гарантированную доставку соединений в виде потоков байт.

- *Telnet* является протоколом эмуляции терминала и позволяет рассматривать все удаленные терминалы как стандартные «сетевые виртуальные терминалы».

- *FTP – File Transfer Protocol* (протокол передачи файлов) позволяет пользователю просмотреть каталог удаленного компьютера, скопировать один или несколько файлов.

- *SMTP – Simple Mail Transfer Protocol* (простой протокол передачи почты) поддерживает передачу электронной почты между произвольными узлами сети Internet.

- *ARP – Address Resolution Protocol* (протокол разрешения адресов). Осуществляет преобразование (отображение) IP-адресов (числового составного адреса) в MAC-адреса (аппаратные адреса). Обратное преобразование осуществляется с помощью протокола *RARP*.

- *RARP – Reverse Address Resolution Protocol* (обратный протокол разрешения адресов).

- *DNS – Domain Name System* (система доменных имен).

- *RIP – Routing Information Protocol* (протокол маршрутизации).

- *HTTP – Hypertext Transfer Protocol* (протокол передачи гипертекстов) применяется для передачи гипертекста в сети Internet.

- *SNMP – Simple Network Management Protocol* (протокол управления сетями).

- *Gopher* – распределенная система управления базами данных.

На характеристиках наиболее распространенных протоколов TCP/IP мы остановимся позднее при рассмотрении сети Internet, а сейчас лишь отметим, что стек протоколов TCP/IP – это самый популярный стек протоколов, широко используемый как в глобальных, так и в локальных сетях. Сегодня фактически именно он является стандартом для межсетевого взаимодействия.

### Стек протоколов IPX/SPX

Этот стек протоколов является оригинальным стеком протоколов фирмы Novell, разработанным для сетевой операционной системы NetWare.

Многие особенности стека IPX/SPX обусловлены ориентацией ранних версий ОС NetWare на работу в локальных сетях небольшого размера, состоящих из персональных компьютеров со скромными ресурсами. Понятно, что для таких компьютеров компании Novell нужны были протоколы, на реализацию которых требовалось бы минимальное количество оперативной памяти. В результате протоколы стека IPX/SPX до недавнего времени хорошо работали в локальных сетях. Сейчас стек IPX/SPX реализован не только в NetWare, но и в нескольких других сетевых ОС, например Microsoft Windows NT, Sun Solaris, SCO UNIX.

Чаще всего протоколы устанавливаются автоматически при инсталляции операционной системы. Например, в Microsoft Windows NT Server протоколом по умолчанию является TCP/IP. Однако иногда надо установить новый протокол, изменить порядок следования протоколов в списке привязки или удалить

протокол. Чтобы установить другие протоколы, обычно можно воспользоваться специальными утилитами, встроенными в ОС.

#### **4.3.4. Передача данных по сети**

Для выяснения сущности передачи информации по сети рассмотрим подробнее назначение и процедуру формирования кадров.

##### **Функции пакетов**

Данные обычно содержатся в больших по размерам файлах. Однако сети не будут нормально работать, если компьютер посылает этот блок данных целиком. Существует две причины, замедляющие работу сети при передаче по кабелю больших блоков данных.

1. Такой блок, посылаемый одним компьютером, заполняет кабель и «связывает» работу всей сети.

2. Возникновение ошибок при передаче крупных блоков приведет к повторной передаче всего блока. В случае же повреждения небольшого блока данных требуется повторная передача именно этого небольшого блока, что значительно экономит время.

В связи с этим данные разбиваются на небольшие управляемые блоки. Эти блоки и называются пакетами. *Пакет – группа байтов, передаваемых абонентами сети друг другу.*

Таким образом, пакет – это основная единица информации в компьютерных сетях. При этом скорость их передачи возрастает настолько, что каждый компьютер в сети получает возможность принимать и передавать данные практически одновременно. В компьютере-получателе пакеты накапливаются и выстраиваются в необходимом порядке для восстановления исходного вида данных.

При разбиении данных на пакеты сетевая операционная система добавляет к каждому пакету специальную управляющую информацию, которая обеспечивает:

- передачу исходных данных небольшими блоками;
- сбор данных при их получении в надлежащем порядке;
- после сборки проверку данных на наличие ошибок.

##### **Структура пакетов локальных сетей**

Структура пакета определяется прежде всего аппаратными особенностями данной сети, выбранной топологией и типом среды передачи информации.

Типичный пакет может содержать в себе следующие основные поля:

- *стартовая комбинация* (не обязательная) обеспечивает настройку аппаратуры адаптера или другого сетевого устройства на прием и обработку пакета;

- *сетевой адрес (идентификатор) принимающего абонента* – индивидуальный или групповой номер, присвоенный принимающему абоненту в сети, позволяет приемнику распознать пакет, адресованный ему или всем абонентам сети;
- *сетевой адрес (идентификатор) передающего абонента* – индивидуальный или групповой номер, присвоенный передающему абоненту в сети, информирует принимающего абонента, откуда пришел данный пакет;
- *служебная информация* указывает тип пакета, его номер, а также то, что с ним надо делать, и т.д.;
- *данные* – это информация, ради передачи которой используется данный пакет;
- *коды коррекции ошибок* (синонимы – остаток избыточной циклической суммы, контрольная сумма, числовой код) – это информация, формируемая передатчиком по определенным правилам и содержащая в свернутом виде информацию обо всем пакете. Она используется для проверки правильности передачи пакета на приемном конце;
- *стоповая комбинация* (не обязательна) информирует принимающего абонента об окончании пакета, обеспечивает выход аппаратуры приемника из состояния запроса.

Нередко выделяют:

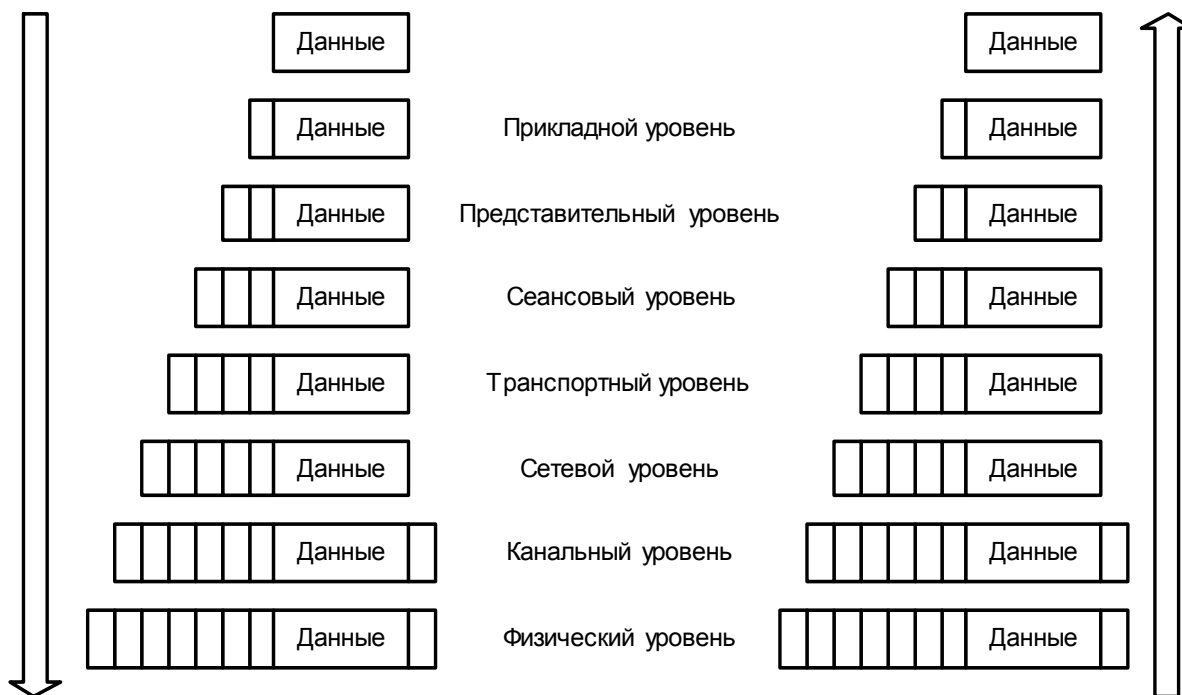
- *начальное управляющее поле пакета*, то есть поле, включающее в себя стартовую комбинацию, сетевые адреса приемника и передатчика, а также служебную информацию. Начальное управляющее поле часто называют заголовком пакета;
- *поле данных* – непосредственно передаваемые данные;
- *конечное управляющее поле*, включающее в себя контрольную сумму и стоповую комбинацию.

Пакеты делятся на два основных типа: *управляющие* (не содержат поля данных) и *информационные* (поле данных присутствует). Управляющие пакеты служат для решения вспомогательных задач по обмену в сети, например для установления соединения между абонентами, для завершения соединения после прекращения сеанса обмена, для подтверждения приема информационного пакета.

Передающий абонент сначала запрашивает с помощью управляющего пакета принимающего абонента о готовности принять данные. Принимающий абонент отвечает управляющим пакетом о своей готовности. Затем следует собственно передача данных, причем на каждый информационный пакет от передатчика приемник отвечает соответствующим пакетом о подтверждении приема. После окончания передачи данных передающий абонент заканчивает сеанс связи управляющим пакетом.

Процедура формирования пакета представлена на **рис. 4.15**. Процесс формирования пакета начинается на прикладном уровне, т.е. там, где рождаются данные. Информация, которую надо переслать по сети, проходит сверху вниз все семь уровней, начиная с прикладного, при этом на каждом уровне к блоку данных добавляется *заголовок*, т.е. информация, предназначенная для соответствующего уровня компьютера-получателя. Отметим еще раз, что заголовок –

служебная информация, которая необходима для адресации сообщений и для некоторых контрольных функций.



*Рис. 4.15. Формирование пакета*

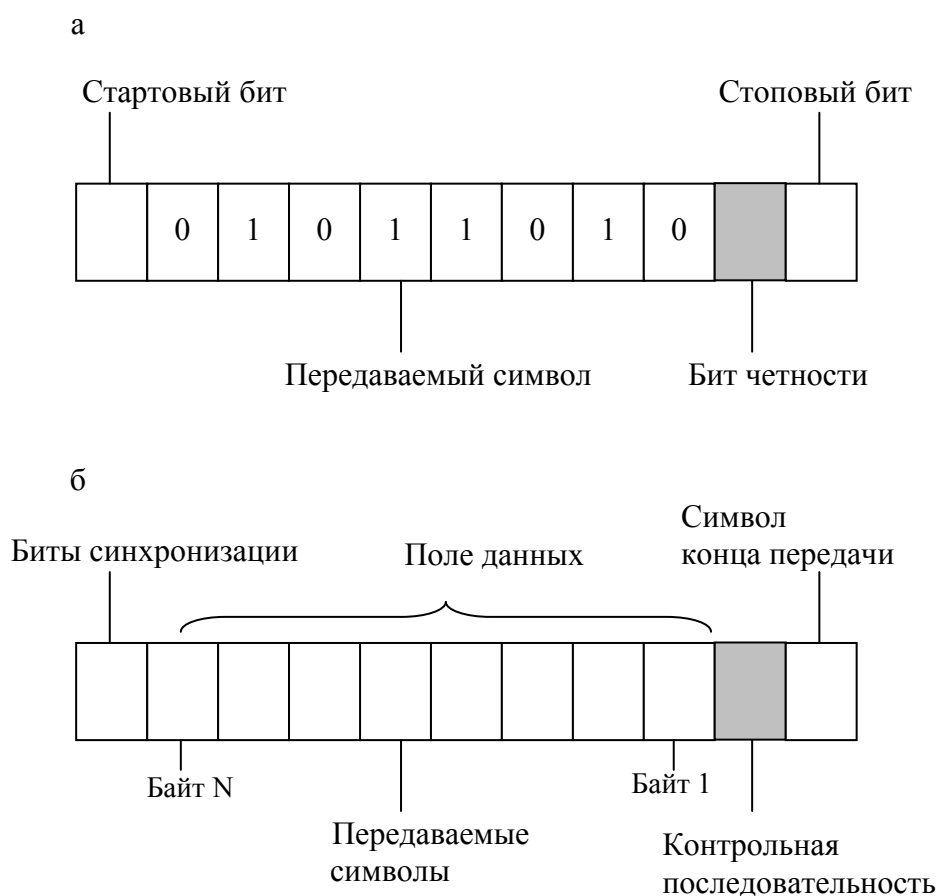
Транспортный уровень разбивает исходный блок данных на пакеты, при этом к каждому пакету добавляется информация, которая поможет компьютеру-получателю восстановить исходные данные из последовательности пакетов. Канальный уровень кроме заголовка добавляет еще и концевик, формируя кадр данных.

### *Типы синхронизации данных*

Синхронизация данных – это согласование различных процессов во времени. При обмене данными на физическом уровне единицей информации является бит, поэтому средства физического уровня поддерживают побитовую синхронизацию между приемником и передатчиком. Канальный уровень оперирует кадрами данных и обеспечивает синхронизацию между приемником и передатчиком на уровне кадров. В обязанности приемника входят распознавание начала первого байта кадра, распознавание границ полей кадра и распознавание признака окончания кадра. Процессы передачи или приема информации в вычислительных сетях могут быть привязаны к определенным временным отметкам, т.е. один из процессов может начаться только после того, как получит полностью данные от другого процесса. Такие процессы называются синхронными.

В то же время существуют процессы, в которых нет такой привязки, и они могут выполняться независимо от степени полноты переданных данных. Такие процессы называются асинхронными.

*Асинхронная связь (старт-стоп связь).* Обычно достаточно обеспечить синхронизацию на указанных двух уровнях – битовом и кадровом, чтобы передатчик и приемник смогли обеспечить устойчивый обмен информацией. Однако при плохом качестве линий связи для удешевления аппаратуры и повышения надежности передачи данных вводят дополнительные средства синхронизации на уровне байт. Этот метод связи используется обычно стандартными телефонными линиями. Каждый символ – буква, число или знак – раскладывается в последовательность битов. Каждая из этих последовательностей отделяется от других *стартовым битом и стоповым битом*, которые позволяют произвести выделение их из потока передачи (рис. 4.16, а). Передающие и принимающие уст-



**Рис. 4.16.** Асинхронная (а) и синхронная (б) передача данных

ройства должны согласовывать комбинацию стартовых и стоповых битов. Принимающий компьютер для управления синхронизацией использует стартовые биты, готовясь тем самым к приему следующего байта данных. Связь этого типа не синхронизируется, т.е. отсутствует синхронизирующее устройство. *Передающий компьютер – просто шлет, а принимающий – просто их получает.* Затем принимающий компьютер проверяет данные, чтобы убедиться в том, что они приняты без ошибок. При асинхронной связи используется специальный бит – *бит четности*. Бит четности используется для контроля правильности передачи данных и принимает такое значение, чтобы в передаваемом символе общее число единиц (или нулей) всегда было четное или нечетное в зависимости от на-

чальной установки регистров устройства оконечного оборудования данных. Приемное устройство заново вычисляет четность поступаемых данных и сравнивает полученный результат с принятым значением бита четности. При несовпадении четности считается, что произошла ошибка в передаче данных. Заметим, что 25 % трафика данных состоит из управляющей и координирующей информации. Максимальная скорость передачи в этом случае – 115 тыс. бит/с.

*Синхронная связь (рис. 4.16, б)* основана на схеме синхронизации, согласованной между двумя устройствами. При этом выделяются биты из группы при передаче их кадрами. Для установки синхронизации и периодической проверки ее правильности используются специальные символы. Передача завершается в конце одного кадра и начинается вновь на следующем кадре. Этот метод более эффективен, чем асинхронная передача. В случае ошибки синхронная схема распознавания и коррекции ошибок просто повторяет передачу кадров. Таким образом, при синхронной передаче информация передается блоками, которые обрамляются специальными управляющими символами. В состав блока включаются также специальные синхросимволы, обеспечивающие контроль состояния физической передающей среды, и символы, позволяющие обнаружить ошибки при обмене информацией. В конце блока данных при синхронной передаче в канал связи выдается контрольная последовательность, сформированная по специальному алгоритму. По этому же алгоритму формируется контрольная последовательность при приеме информации из канала связи. Если обе последовательности совпадают, то ошибок нет. Если повторные передачи не дают положительного результата, то фиксируется состояние аварии. Синхронная связь используется практически во всех цифровых системах связи.

#### **4.3.5. Методы доступа в сетях**

Для рассмотрения процедуры передачи данных по кабелю предварительно рассмотрим методы доступа в сетях. В сети несколько компьютеров должны иметь совместный доступ к кабелю. Однако если два или более компьютеров попытаются одновременно передавать данные, то их пакеты столкнутся друг с другом и будут испорчены, т.е. наступает *коллизия*. Методы доступа служат для предотвращения одновременного доступа к кабелю нескольких компьютеров, упорядочивания передачи и приема данных по сети, гарантируя, что в каждый момент времени только один компьютер может работать на передачу.

Таким образом, *метод доступа* – это набор правил, которые определяют, как компьютер должен отправлять данные по сетевому кабелю.

*Основные методы доступа в компьютерных сетях.*

*Управление обменом в сети "звезда"*

Чаще всего центральный абонент может производить обмен только с одним из периферийных абонентов. Поэтому в любой момент надо выделить только одного абонента, ведущего передачу. Здесь возможны два решения.

Первое из них можно назвать "активный центр". В этом случае центральный абонент посылает запросы (управляющие пакеты) по очереди всем пе-

риферийным абонентам. Периферийный абонент, который хочет передавать (первый из опрошенных), посылает ответ или же сразу начинает передачу. Сеанс обмена проводится именно с ним. После окончания этого сеанса центральный абонент продолжает опрос по кругу. Периферийные абоненты имеют в данном случае так называемые географические приоритеты: максимальный приоритет у того из них, кто ближе расположен к последнему абоненту, закончившему обмен. Если же хочет передать центральный абонент, он передает без всякой очереди.

Второе возможное решение – "пассивный центр". В этом случае центральный абонент по очереди не опрашивает, а слушает всех периферийных абонентов (т.е. принимает пакеты только от одного из них). Те периферийные абоненты, которые хотят передать, периодически посылают запросы и ждут на них ответа. Когда центр принимает запрос, он отвечает запросившему абоненту (разрешает ему передачу), и тот передает. Приоритеты здесь такие же, как и в первом случае.

Преимущества централизованного управления достаточно очевидны: принципиальная невозможность любых конфликтов (коллизий) между абонентами, т.к. все решения принимаются в одном месте. В этом случае гарантировано время доступа, то есть время, проходящее от момента возникновения желания передавать до момента начала передачи. Если интенсивность обмена (нагрузки сети) большая, т.е. все абоненты очень активны, то передача будет осуществляться строго по очереди, и время доступа не превысит заранее известной величины – суммарного времени передачи своих пакетов всем периферийным абонентам.

Недостатки централизованного управления связаны:

- с низкой устойчивостью этого метода к отказам, т.к. если центр выходит из строя, то обмен невозможен;
- с недостаточной гибкостью (центр работает по жесткому заданному алгоритму);
- с низкой скоростью управления (даже если все время передает только один периферийный абонент, ему приходится ждать, пока центр опросит всех).

### *Управление обменом в сети типа "шина"*

В этой топологии возможно точно такое же централизованное управление, как и в "звезде" (т.е. физическая сеть – "шина", но логическая – "звезда"). При этом один из абонентов "центральный" посылает всем остальным "периферийным" запросы, выясняя, кто хочет передать, затем разрешает передачу одному из них. После окончания передачи абонент сообщает "центру", что он закончил, и "центр" снова начинает опрос. Все преимущества и недостатки такого управления – те же, что и в случае "звезды". Единственное отличие состоит в том, что центр здесь не перекачивает информацию от одного абонента к другому, как в "звезде", а только управляет доступом.

Однако гораздо чаще в "шине" реализуется децентрализованное управление, так как аппаратные средства абонентов одинаковы, т.е. осуществляется пассивная технология. При этом все абоненты также имеют равный доступ к се-



ти, и решение, когда можно передавать, принимает каждый абонент на месте, исходя из анализа состояния сети. Возникает конкуренция между абонентами за захват сети, и, следовательно, возможны конфликты между ними.

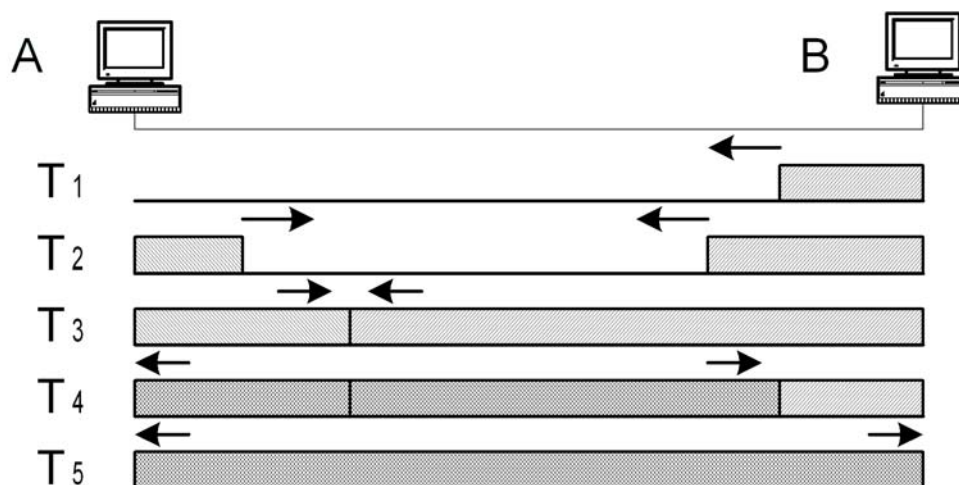
Существует множество алгоритмов доступа или, как еще говорят, сценариев доступа, часто очень сложных. Их выбор зависит от скорости передачи в сети, от длины шины, загруженности сети (интенсивности обмена или трафика сети).

Рассмотрим некоторые методы доступа.

### 1. Множественный доступ с контролем несущей и обнаружением коллизий (Carrier Sense Multiply Access with Collision Detection, CSMA/CD, состязательный метод)

Вдумайтесь в название этого метода доступа. Компьютеры как бы «прослушивают» кабель, отсюда – контроль несущей. Чаще всего сразу несколько компьютеров в сети «хотят» передать данные, отсюда – множественный доступ. Передавая данные, компьютеры «прослушивают» кабель, чтобы, обнаружив коллизии, некоторое время переждать, а затем возобновить передачу; отсюда термин – обнаружение коллизий.

Остановимся более подробно на этом вопросе. На **рис. 4.17** представлен процесс столкновения пакетов. В начальный момент времени  $T_1$  абонентская



**Рис. 4.17.** Столкновение сообщений (коллизия)

система  $B$  начала передавать информацию. В этот же момент абонентская система  $A$  прослушивает передающую среду, однако из-за конечного времени распределения сигнала ей не удастся обнаружить сообщение, посылаемое абонентской системой  $B$ . В следующий момент времени  $T_2$  абонентская система  $A$  начинает передавать информацию, в результате чего в момент времени  $T_3$  сообщения сталкиваются. Дальнейшая передача сообщений теряет смысл. С целью своевременного обнаружения конфликтов абонентская система в процессе передачи информации постоянно контролирует передающую среду и при появлении столкновения прекращает передачу. Так, абонентская система  $A$  прекращает передачу в момент времени  $T_4$ , а абонентская система  $B$  – в момент  $T_5$ . Через некоторый промежуток времени после прекращения передачи конфликтующие абонентские

системы осуществляют повторную попытку передачи информации. Этот метод известен как состязательный метод, поскольку сетевые компьютеры «состязаются», конкурируют между собой за право передавать данные.

В силу ослабления сигнала при расстояниях свыше 2500 м механизм обнаружения коллизий CSMA/CD не эффективен. Коллизию можно не услышать и начать передавать данные, что приведет к разрушению пакетов данных. Однако заметим, что современные CSMA/CD настолько быстры, что пользователи даже не задумываются над тем, что применяется состязательный метод. В то же время сеть с методом доступа CSMA/CD, обслуживающая многих пользователей, которые работают с несколькими системами управления базами данных, может практически зависнуть из-за чрезмерного сетевого трафика.

## **2. Множественный доступ с контролем несущей и предотвращением коллизий (Carrier Sense Multiply Access with Collision Avoidance, CSMA/CA, избежание конфликтов)**

Он не так популярен, как CSMA/CD или передача маркера. Метод состоит в том, что после освобождения сети всеми желающими передавать абонентами передается не пакет, а специальный сигнал, контролируя который они обнаруживают конфликты. В этом случае сталкиваются только эти сигналы, а не пакеты. Следовательно, каждый компьютер перед передачей данных в сеть сигнализирует о своем намерении, поэтому остальные компьютеры «узнают» о готовящейся передаче и могут избежать коллизии. Однако этот метод доступа работает медленнее и имеет меньшую пропускную способность.

### *Управление обменом в сети типа "кольцо"*

Кольцевая топология имеет свои особенности при выборе метода управления обменом. Важным фактором является то, что любой пакет, посланный по кольцу, последовательно пройдя всех абонентов, через некоторое время возвратится в ту же точку (топология замкнута). Отметим, что сети типа "кольцо" бывают однонаправленными и двунаправленными. Мы будем рассматривать только однонаправленные как более распространенные.

**Маркерный метод управления (с передачей полномочий).** Суть доступа с передачей маркера заключается в следующем. Пакет особого типа, *маркер*, циркулирует по кольцу от компьютера к компьютеру (рис. 4.18). Чтобы послать данные в сеть, любой из компьютеров сначала должен дождаться прихода свободного маркера и захватить его. Когда какой-либо компьютер «наполнит» маркер своей информацией и пошлет его по сетевому кабелю, другие компьютеры уже не смогут передавать данные. Так как в каждый момент времени только один компьютер будет использовать маркер, то в сети не возникнет ни состязания, ни коллизий. Таким образом, в данном случае происходит одномоментное использование маркера только одним компьютером.

В сети длиной 200 м маркер циркулирует с огромной частотой, равной 10 тыс. оборотов в секунду, поэтому пользователь визуально не наблюдает задержек в его передаче.

Данный метод похож на метод опроса (централизованный), хотя явно выделенного центра здесь не существует. Однако некий центр все-таки должен

присутствовать: один из абонентов (или специальное устройство) должен следить, чтобы маркер не потерялся (например, из-за помех или сбоя в работе какого-то абонента).

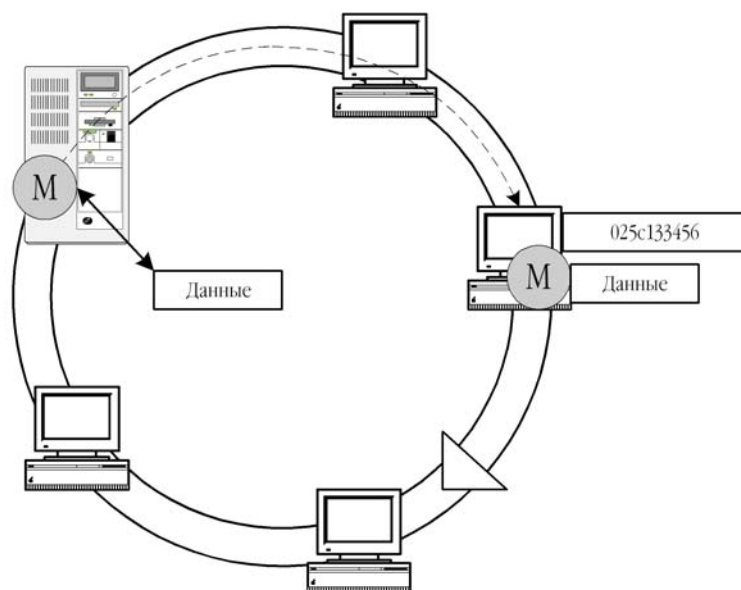


Рис. 4.18. Метод доступа с передачей маркера

**Метод кольцевых сегментов (слотов).** Основное отличие этого метода от маркерного состоит в том, что нескольким абонентам разрешена передача одновременно и в любой момент, в случае же маркерного метода передает только один абонент. Вместо одного маркера в сети используются несколько так называемых слотов (обычно от 2 до 8), которые выполняют, по сути, ту же самую функцию, что и маркер, – функцию временных отметок. Эти слоты идут по "кольцу" довольно часто. Временной интервал между ними невелик, и поэтому между ними может уместиться немного информации (обычно от 8 до 32 байт). При этом каждый слот может находиться в свободном или занятом состоянии. Алгоритм доступа к сети, реализуемый при данном методе, включает в себя следующие этапы.

1. Абонент, желающий передавать, разбивает свою информацию на слоты (маленькие пакеты) установленного размера.

2. Затем ждет прихода свободного слота и загружает его первой частью своей информации, потом ждет следующего свободного слота и помещает в него вторую часть, и так до полной передачи всего объема информации. В каждом слоте существует бит, определяющий, свободен или занят данный слот, поле сетевого адреса приемника и адреса передатчика, а также бит признака конца передачи. Время при данном методе дискретизируется, и поэтому конфликты не происходят.

3. Абонент, которому адресована информация, выбирает слоты, содержащие адресованную ему информацию, и устанавливает в принятом слоте бит подтверждения. Процедура продолжается до последнего адресованного ему слота.

4. Передающий абонент получает свой слот обратно по "кольцу" и освобождает их (помечает как свободный). При этом он сразу же имеет подтверждение приема.

Очевидно, что при данном методе передачу могут вести сразу несколько абонентов. Кстати, совсем не обязательно каждому передающему абоненту занимать соседние слоты: слоты, находящиеся рядом, могут содержать совершенно разную информацию, относящуюся к разным абонентам.

Как и в случае маркерного метода управления, здесь нужен "центр", который следит за прохождением слотов и восстановлением их в случае исчезновения.

#### 4.4. Стандартные сетевые технологии

*Стандартные сетевые технологии* – это комбинация стандартов, топологий, протоколов и реализующих их программно-аппаратных средств, необходимых для создания работоспособной сети.

В настоящее время насчитывается более 2000 сетей, имеющих тот или иной уровень стандартизации, но широкое распространение и всеобщее признание получили не более 10 базовых сетевых технологий. Это связано с тем, что именно эти сети поддерживаются наиболее мощными фирмами и поэтому доведены до уровня международных стандартов. Примерами базовых сетевых технологий могут служить Ethernet, Token Ring, FDDI и др.

Наибольшее распространение среди стандартных сетей получила сеть Ethernet, поэтому рассмотрим ее в качестве примера стандартного решения сетевых проблем. Эта сеть появилась в 1972 году, а в 1980 году она стала международным стандартом. Сеть Ethernet сейчас наиболее популярна в мире. Число сетей, построенных на основе этой технологии, к настоящему моменту оценивается в 5 млн., а количество компьютеров, работающих в таких сетях, в 50 млн., что составляет свыше 100 млн. абонентов в 1998 году или более 80% рынка. Похоже, что сеть Ethernet будет признана в качестве основного сетевого решения на уровне 3 архитектуры информационной системы (см. **рис. 1.2**).

Доступ к сети осуществляется по методу CSMA/CD. Передача осуществляется пакетами переменной длины.

Для сети Ethernet, работающей на скорости 10 Мбит/с, стандарт определяет четыре основных типа среды передачи:

- 10BASE-5 (толстый коаксиальный кабель);
- 10BASE-2 (тонкий коаксиальный кабель);
- 10BASE-T (витая пара);
- 10BASE-F (оптоволоконный кабель).

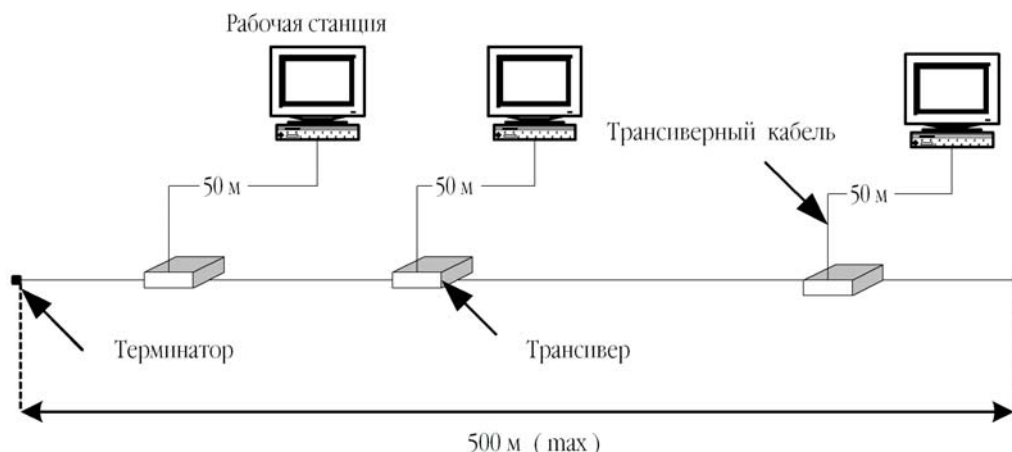
Обозначение среды передачи включает в себя три элемента: цифра "10" означает скорость передачи 10 Мбит/с, слово BASE означает передачу в основной полосе частот 10 МГц, а последний элемент означает допустимую длину сегмента: "5" – 500 метров, "2" – 200 метров (точнее, 185) или тип линии связи: "T" – витая пара, "F" – оптоволокно.

Точно так же для сети Ethernet, работающей на скорости 100 Мбит/с (Fast Ethernet), стандарт определяет три типа среды передачи:

- 100BASE-T4 (счетверенная витая пара);
- 100BASE-TX (сдвоенная витая пара);
- 100BASE-FX (оптоволоконный кабель).

Здесь цифра "100" означает скорость передачи 100 Мбит/с, буква "Т" означает витую пару, буква "F" – оптоволоконный кабель. Типы 100BASE-TX и 100BASE-FX иногда объединяют под именем 100BASE-X.

Рассмотрим подробнее основные виды сети Ethernet.

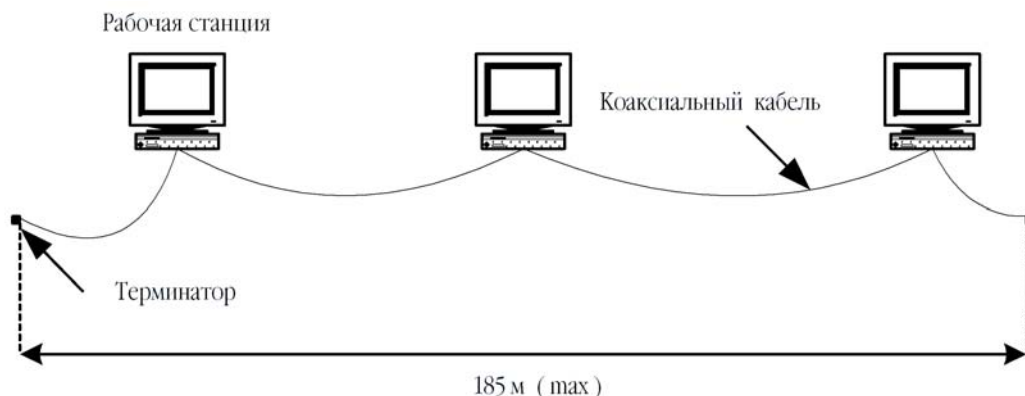


**Рис. 4.19.** Сегмент сети Ethernet 10BASE-5

- 10Base-5 (Thick, "толстый" Ethernet), **рис. 4.19**. Для построения сети Thick Ethernet применяется шинная топология с максимальной длиной сетевого сегмента не более 500 м, общим количеством рабочих станций в сегменте менее 100 и минимальным расстоянием между рабочими станциями не более 2,5 м. Для подключения к основной линии сети рабочей станции можно использовать отводной кабель длиной не более 50 м.

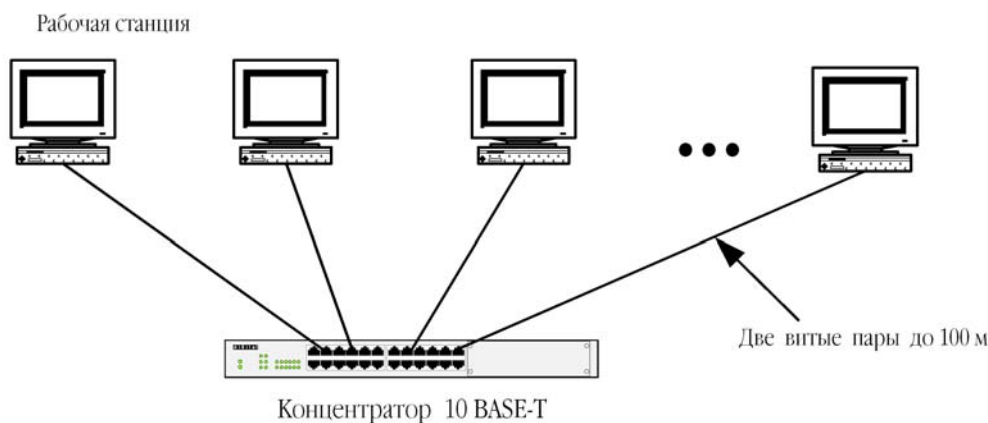
- 10Base-2 (Thin, «тонкий» Ethernet), **рис. 4.20**. Реализация Thin Ethernet также использует топологию «шина». Максимальная длина сегмента составляет 185 м, общее количество рабочих станций в сегменте не более 30, а минимальное расстояние между рабочими станциями 0,5 м. В отличие от сети Thick Ethernet, сети Thin Ethernet не используют отводов – основная линия сети подводится непосредственно к рабочей станции и подключается при помощи разъема к специальному соединителю – коннектору, который, в свою очередь, подключается непосредственно к сетевому адаптеру.

- 10Base-T (10Base-T), **рис. 4.21**. Этот вариант сетевой конфигурации используется в настоящее время наиболее часто. Для построения сети применяются кабели UTP/STP (витая пара или экранированная витая пара), соединяемые в соответствии с топологией звезды. Максимальная длина сегмента составляет 100 м. Число рабочих станций, подключаемых к отдельному сетевому сегменту, ограничивается только производительностью сети. Опыт показывает, что при повседневной рабочей нагрузке на сеть подключение к сети более 30 рабочих станций приводит к ее перегрузке.



**Рис. 4.20.** Сегмент сети Ethernet 10BASE-2

- 10BASE-F. Топология аналогична топологии стандарта 10Base-T. Имеется несколько вариантов этой спецификации – FOIRL (расстояние до 1000 м), 10Base-FL (расстояние до 2000 м), 10Base-FB (расстояние до 2000 м).
- Сеть Fast Ethernet представляет собой дальнейшее развитие сети Ethernet за счет увеличения в 10 раз тактовой частоты, т.е. это высокоскоростная сеть для работы на скорости 100 Мбит/с. При создании сети для работы с мультимедиа или проведения видеоконференций необходимо обеспечить ее высокое быстродействие. Для этой цели можно использовать волоконно-оптические линии связи или линии Fast Ethernet. При этом основные аспекты построения сети Ethernet остались неизменными. Основные отличия касаются физического уровня и связаны с использованием передающей среды. В зависимости от применяемого кабеля имеются три наименования этой сети: 100Base-TX с двумя витыми парами проводов и 100Base-T4 с четырьмя витыми парами и 100Base-FX – для двухжильного оптоволоконного кабеля.



**Рис. 4.21.** Структура сети Ethernet 10BASE-T

тимедиа или проведения видеоконференций необходимо обеспечить ее высокое быстродействие. Для этой цели можно использовать волоконно-оптические линии связи или линии Fast Ethernet. При этом основные аспекты построения сети Ethernet остались неизменными. Основные отличия касаются физического уровня и связаны с использованием передающей среды. В зависимости от применяемого кабеля имеются три наименования этой сети: 100Base-TX с двумя витыми парами проводов и 100Base-T4 с четырьмя витыми парами и 100Base-FX – для двухжильного оптоволоконного кабеля.

Отметим в заключение, что потребности рынка в высокоскоростной технологии передачи информации в локальных сетях привели в последнее время (1998–1999) к созданию новой ступени сетей Ethernet – Gigabit Ethernet со скоростью передачи 1000 Мбит/с. Эта ступень позволяет строить крупные локальные сети, в которых мощные серверы и магистрали нижних уровней сети работают на скоростях 10 Мбит/с, 100 Мбит/с, а магистраль Gigabit Ethernet объединяет их, обеспечивая достаточно большой запас пропускной способности. Разработ-

чики Gigabit Ethernet сохранили большую степень преемственности с технологиями Ethernet и Fast Ethernet. В настоящее время разрабатывается стандарт 10 Гб Ethernet.

#### 4.5. Сетевые операционные системы

Остановимся здесь только на краткой характеристике процедур, характерных для большинства сетевых операционных систем. Компьютер, являющийся частью сети, в действительности работает под управлением и автономной, и сетевой операционной систем одновременно. В современных сетевых операционных системах типа Microsoft Windows 95/98/2000, Microsoft Windows NT Workstation, Microsoft Windows NT Server автономная и сетевая операционные системы скомбинированы в одну операционную систему, которая поддерживает функционирование как автономного компьютера, так и системы в целом.

Сетевая операционная система выполняет следующие основные функции:

- координирует функции всех компьютеров и периферийных устройств в сети;
- обеспечивает защищенный доступ к данным и периферийным устройствам в сети.

Сетевое программное обеспечение состоит из двух важнейших компонентов:

- сетевого программного обеспечения, устанавливаемого на компьютерах-клиентах;
- сетевого программного обеспечения, устанавливаемого на компьютерах-серверах.

Например, в сети два сервера работают под управлением Microsoft Windows NT Server, а у клиентов могут быть установлены ОС Microsoft Windows NT Workstation или Microsoft Windows 98/2000.

*Специфика клиентского программного обеспечения.* В автономных системах, когда пользователь набирает команду запрос к компьютеру на выполнение некой задачи, этот запрос передается через локальную шину процессору компьютера. В сетевой же среде, когда пользователь делает запрос, относящийся к ресурсам удаленного сервера, этот запрос передается из локальной шины компьютера в сеть, к серверу. Программное обеспечение клиента включает в себя *редиректор*, который и осуществляет передачу запросов. Он перехватывает в компьютере запросы и определяет, где они должны выполняться, а также следит за тем, чтобы обозначения устройств соответствовали определенным сетевым ресурсам. Например, в Microsoft Windows NT наиболее простой способ подключения к сетевому диску, к которому необходим доступ, – использование File Manager. Когда на сервере указывается имя совместно используемого каталога, File Manager присваивает одну из букв английского алфавита в качестве обозначения этого каталога, например G. Затем можно обратиться к этому каталогу на удаленном компьютере как к диску G, и редиректор определяет его местонахождение.

Приведем краткие сравнительные характеристики основных сетевых операционных систем, ориентированных на сети на основе сервера.

**Novell NetWare.** Сетевая ОС Novell NetWare представляет собой 32-разрядную многозадачную операционную систему, работающую в защищенном режиме работы процессора. Novell NetWare является операционной системой с централизованным управлением. Это означает, что в сети один или несколько компьютеров используется в качестве файл-серверов. На этих компьютерах работает ОС NetWare. Остальные компьютеры используются в качестве рабочих станций, и на них должна быть загружена сетевая оболочка – специальный компонент NetWare для рабочих станций. NetWare стартует из MS DOS. Как и всякая другая операционная система, Novell NetWare работает с аппаратурой через драйверы. Кроме драйверов можно загружать (и запускать) программы, выполняющие функции обслуживания сервера и сети.

Программные продукты Novell NetWare популярны в силу нескольких причин:

- ОС NetWare способна поддерживать рабочие станции, управляемые MS DOS, Windows, OS/2, Unix, Windows NT и другими ОС;
- LAN NetWare может разрастаться до больших размеров;
- обеспечивает хорошую надежность работы;
- средства защиты данных более чем надежны для большинства организаций.

**Unix.** Она развивалась, ориентируясь на поддержку высокопроизводительных рабочих станций, больших многопроцессорных систем больших ЭВМ, суперкомпьютеров, а также специальных систем, требующих высокой доступности, отказоустойчивости или обработки в реальном масштабе времени. К основным преимуществам Unix можно отнести:

- переносимость от рабочих станций до больших машин и систем с массовым параллелизмом;
- поддержка большинства процессорных технологий;
- наличие реализаций для отказоустойчивых и высокодоступных систем и для приложений реального времени;
- поддержка большого числа пользователей;
- поддержка большого числа файловых систем;
- развитые коммуникационные протоколы и сетевая обработка;
- соответствие главным открытым стандартам;
- способность большого количества приложений работать в среде Unix.

**Windows NT.** Наиболее распространенная операционная система, способная одновременно выполнять много пользовательских и системных программ. Стандартная ОС Windows NT, действующая как сервер, может поддерживать многочисленные прикладные программы. Фактически каждая система Windows NT может действовать как клиент или сервер и даже в обоих вариантах одновременно. Даже есть возможность заставить автономную систему работать в качестве клиента для самой себя.

Приведем некоторые основные преимущества Windows NT по сравнению с рассмотренными выше сетевыми операционными системами:



- объектно-ориентированное проектирование операционной системы. Конкретные функции выполняются отдельными модулями с четко определенными интерфейсами;

- основанная на использовании механизма нитей архитектура, удобная для реализации функций сервера, особенно на многопроцессорной системе;

- наличие очень большого числа установленных систем MS Windows и многочисленных приложений.

В среде Windows NT существует три категории серверов.

1. Главные контроллеры доменов (Primary Domain Controller, PDC). Домен – это логическое объединение компьютеров, которое облегчает администрирование. Первый сервер, устанавливаемый в домене, должен выступать главным контроллером домена, который проверяет права пользователей; может также выступать как сервер файлов, печати и приложений.

2. Резервные контроллеры доменов (Backup Domain Controller, BDC). Хранят резервные копии базы данных пользователей, базы данных домена. Может функционировать как сервер файлов, печати и приложений.

3. Простые серверы (Standart Alone). Серверы, выполняющие функции серверов файлов, печати и приложений.

Устанавливая сервер, необходимо указать: является ли он главным сервером, резервным контроллером или простым сервером.

Во время установки Microsoft Windows NT Server предлагается выбрать или сконфигурировать плату сетевого адаптера, установленного в вашем компьютере. Microsoft Windows NT Server по умолчанию предлагается протокол TCP/IP, хотя можно выбрать и другие из списка предлагаемых.

В дальнейшем производится установка сетевых служб. Сетевые службы – это набор приложений сетевой операционной системы. Эти сетевые службы выполняют различные сетевые задачи, настраиваются на автоматический запуск, однако при желании можно запустить дополнительные службы – электронную почту, почтовый ящик, гипертекст и т.д. [51].

**Управление пользователями.** Сеть, которая может работать сама по себе, еще не придумана. Время от времени нужно подключать новых пользователей, некоторых удалять, необходимо предоставлять новые ресурсы, предоставлять (лишать) соответствующие права на доступ к ним и т.п. Одним словом, *сетью необходимо управлять*. Эти функции и возлагаются на *администратора сети*.

Администратор сети:

- управляет пользователями (управление доступом пользователей к ресурсам);

- управляет ресурсами (установка и поддержание сетевых ресурсов);

- управляет конфигурацией, т.е. осуществляет планирование конфигурации сети, ее расширение и защиту.

Одной из главных задач сетевого администратора – создание *учетных записей пользователей*. Учетные записи создаются и для индивидуальных пользователей, и для групп. Каждому, кто работает в сети, выделяется учетная запись пользователя. Учетная запись состоит из имени пользователя и назначенных ему параметров входа в систему. При попытке пользователя войти в сеть его имя

служит для проверки учетной записи. Имеются утилиты, которые помогают администраторам добавить в базу данных безопасности новые учетные записи.

Учетная запись содержит:

- имя и пароль пользователя;
- права пользователя на доступ к ресурсам сети;
- группы, к которым относятся учетные записи.

Сетевые операционные системы поставляются с заранее созданными пользовательскими учетными записями некоторых групп, которые автоматически активизируются при установке системы. Создание групп в значительной степени упрощает работу администратора.

*Защита информации.* Работая в сети, необходимо быть уверенным в том, что секретные данные таковыми и останутся, поскольку лишь пользователи, которые имеют соответствующие полномочия, смогут получить к ним доступ. В то же время при обеспечении безопасности надо знать меру. Необходимо построить защиту так, чтобы пользователи не испытывали трудности при выполнении работы. Рассмотрим основные методы защиты информации.

*Физическая защита оборудования.* Серверы должны быть физически защищены от случайного или преднамеренного вмешательства. Для этого их целесообразно помещать в специальное помещение, шкаф и т.п. Медный кабель, коаксиальный, подобен радиостанции и излучает электромагнитные сигналы. Эту информацию с помощью специального оборудования легко перехватить, поэтому доступ должен быть ограничен. В частности, целесообразно их прокладывать в специальных стеновых панелях, что затрудняет доступ к ним.

### **Модели защиты**

*Пароль доступа к ресурсу.* В этом случае присваивается пароль каждому общедоступному ресурсу. Таким образом, доступ к ресурсу осуществляется только в том случае, когда пользователь вводит правильный пароль. Во многих системах ресурсы могут быть представлены в совместное использование с различными типами прав доступа: доступ только для чтения, полный доступ, доступ в зависимости от пароля.

*Права доступа.* Защита через права доступа заключается в присвоении каждому пользователю определенного набора прав. При входе в сеть *пользователь вводит пароль*. Сервер, проверяя комбинацию имен пользователя и пароля, т.е. проверяя права пользователя в базе данных безопасности, предоставляет или защищает доступ к сетевым ресурсам. Защита с применением прав доступа обеспечивает более высокий уровень управления доступом к совместно используемым ресурсам, а также более строгий режим безопасности, чем защита паролем. Менее вероятно, что пользователь сообщит свой личный пароль.

*Защита ресурсов.* Проверив и подтвердив имя и пароль пользователя, система безопасности сети предоставляет ему доступ к соответствующим ресурсам. Однако иметь пароль еще недостаточно, для доступа к ресурсам нужны права. Образно говоря, вокруг каждого ресурса возведена ограда безопасности. Эта ограда имеет различные входы, через которые пользователи могут получать доступ к ресурсу. Причем, попав через один вход, пользователь выполнит гораздо больше операций, чем если бы он попал через другой вход. Иными словами,

некоторые входы предоставляют пользователю больше прав на ресурс. Администратор сети определяет, кто какими конкретными входами может пользоваться.

*Права группы.* Наиболее эффективный способ решить задачу – присвоить каждому пользователю соответствующие права – через группу, особенно в больших организациях. Сначала администратор оценивает, какие права необходимы каждому пользователю, а затем включает их в соответствующие группы. Таким образом, администратор сети наделяет конкретными правами целые группы пользователей, а не отдельных пользователей. Права групп реализуются так же, как и права индивидуальных пользователей.

*Дополнительные средства защиты.* Существуют и дополнительные средства защиты, с помощью которых администратор сети может повысить степень сетевой защиты.

*Аудит* – это запись определенных событий в журнал безопасности сервера. Этот процесс отслеживает действия пользователей в сети. В журнале безопасности отражены имена всех пользователей, которые работали с конкретными ресурсами или попытались получить к ним доступ. Аудит фиксирует такие действия, как попытку входа в сеть, подключение к указанным ресурсам и отключение от них; открытие и закрытие файлов; создание, модификацию и удаление каталогов; изменение паролей и т.д.

*Шифрование данных.* Утилита шифрования данных кодирует информацию перед тем, как передать ее по сети, что затрудняет чтение передаваемых по сети данных, если кто-то подсоединился к кабелю. Когда данные поступят на соответствующий компьютер, они будут декодированы в понятную форму с помощью ключа.

#### **4.6. Способы и средства объединения локальных компьютерных сетей**

Очень часто созданная на определенном этапе развития системы локальная вычислительная сеть с течением времени перестает удовлетворять потребностям всех пользователей и возникает необходимость объединения локальных вычислительных сетей. В сетях с небольшим количеством узлов чаще всего используется одна из типовых топологий – общая шина, кольцо, звезда. Все топологии имеют одно общее свойство – *однородность*, которое означает, что все узлы обладают равными правами на доступ к другим компьютерам. Такая однородность структуры делает простой процедуру наращивания числа компьютеров, облегчая обслуживание и эксплуатацию сети. Однако при построении крупных сетей это же достоинство превращается в недостаток, поскольку начинают проявляться ограничения, важнейшими из которых являются:

- ограничения на длину связи между узлами;
- ограничения на количество узлов в сети;
- ограничения на интенсивность трафика, порождаемого узлами сети.

Поскольку каждая из базовых реализаций сети накладывает определенные ограничения, то практически все рабочие сети разделяются на несколько сегментов. Разделение сети на сегменты может способствовать повышению производительности. В большой сети возникает неоднородность информационных

потоков: сеть состоит из множества подсетей рабочих групп, участков, отдельных цехов и других административных образований.

Сети с типовой топологией, в которой все физические сегменты рассматриваются в качестве одной разделяемой среды, оказываются неадекватной структуре информационных потоков в большой сети. Именно здесь в работу включаются *повторители (репитеры), трансиверы, концентраторы, мосты (коммутаторы), маршрутизаторы и шлюзы*.

Следует различать физическую и логическую топологию сети. Под физической топологией сети понимается *конфигурация физических связей между компьютерами, а под логической топологией – конфигурация информационных потоков в сети*.

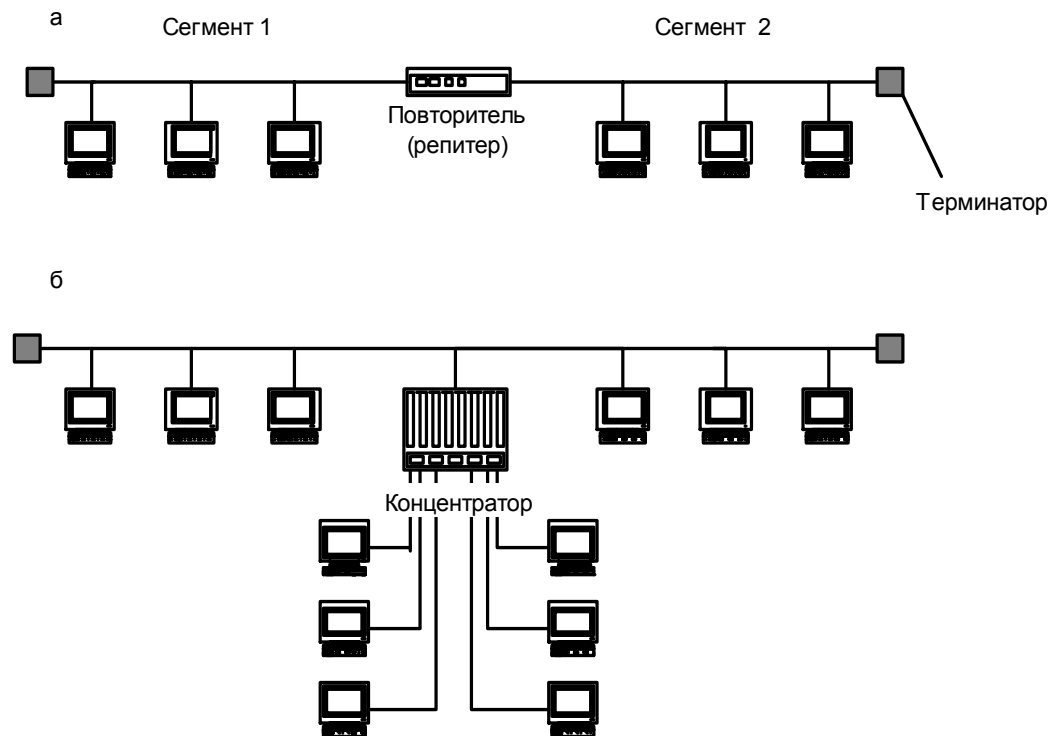
### **Физическая структуризация сети**

**Трансиверы (конвертеры, приемопередатчики, *transceiver*)** служат для двунаправленной передачи между адаптером и сетевым кабелем или между двумя сегментами (отрезками) сетевого кабеля. Основной функцией трансивера является усиление сигналов или преобразование их в другую форму для улучшения характеристик сети, в частности повышения помехоустойчивости и/или увеличения расстояния между абонентами. Например, при подключении к одному оптоволоконному кабелю коаксиального кабеля. В этом случае трансивер выполняет функцию конвертера среды, преобразуя электрические сигналы сети в какие-нибудь другие (оптические, радиосигналы и т.д.).

**Повторители (*репитеры, repeater*)** выполняют более простую задачу, чем трансиверы. Они не преобразуют ни уровни сигналов в сети, ни их физическую природу. Их назначение состоит только в том, чтобы восстановить форму сигнала, искажаемую прохождением в длинной линии. Они служат простыми двунаправленными ретрансляторами сигналов сети и позволяют преодолеть ограничения на длину линий связи за счет улучшения качества передаваемого сигнала – восстановления его мощности и амплитуды, улучшения фронтов и т.п. Основная цель их применения – увеличение длины сети. Они расширяют возможности сети, разделяя ее на сегменты, тем самым уменьшается количество компьютеров на один сегмент. *Трансиверы и повторители не производят абсолютно никакой информационной обработки проходящих через них пакетов. С точки зрения передачи информации они представляют собой абсолютно пассивные устройства*. Поэтому они в принципе не могут хоть как-то изменить основные информационные характеристики сети. Соединяя с их помощью отдельные части сети, мы получаем всего лишь такую же сеть, но только большего объема и с лучшим качеством передачи сигналов.

**Концентратор (*concentrator, hub*)**. Повторитель, который имеет несколько портов и соединяет несколько физических сегментов, называют концентратором (*concentrator*) или хабом (*hub*).

Концентраторы выполняют функцию собранных в одном месте в единый конструктив нескольких повторителей (репитеров) или трансиверов (**рис. 4.22**). Никакой обработки информации они не производят, а только восстанавливают и усиливают сигналы, могут также преобразовывать электрические сигналы в оптические и наоборот. Нужно ли для этого использовать концентратор, не лучше



**Рис. 4.22.** Включение репитера (а) и пассивного концентратора (б)

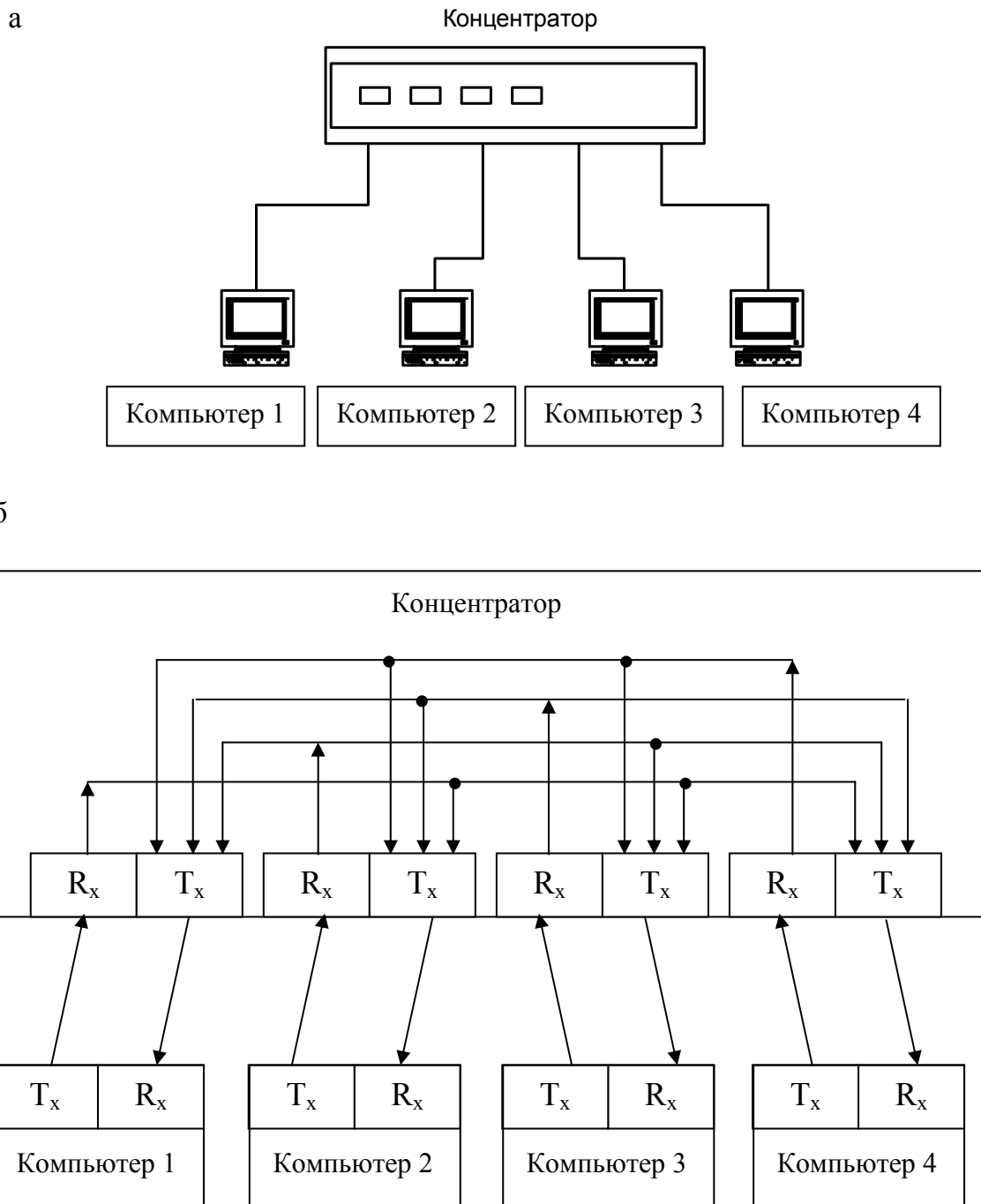
ли обойтись более дешевыми трансиверами и репитерами? Однако концентраторы имеют свои преимущества.

1. Собирая все важные точки в одном месте, мы существенно облегчаем обслуживание сети и контроль за ее работоспособностью, а также облегчаем поиск неисправностей.

2. Концентратор может быть расположен в специальном помещении, куда не имеют доступа лишние люди, что повышает надежность функционирования сети.

К концентратору могут подключаться только части (сегменты) или отдельные абоненты одной и той же сети. Например, сегменты сети, выполненные на тонком кабеле, на толстом кабеле, на оптоволоконном кабеле.

Концентраторы повторяют данные, пришедшие с любого порта, на всех остальных портах, поэтому данные появляются одновременно на всех физических сегментах сети. Пример, представленный на **рис. 4.23**, демонстрирует несовпадение физической и логической топологий сети. Физически компьютеры соединены с помощью концентратора Ethernet по топологии "звезда". Так как концентратор Ethernet повторяет данные, пришедшие с любого порта, на всех остальных портах, то они появляются одновременно на всех компьютерах, так



**Рис. 4.23.** Физическая (а) и логическая (б) топология сети 10BASE-T на базе концентратора Ethernet:  
 $T_x$  – передача сигнала;  $R_x$  – прием сигнала

же как в сети с общей шиной. Таким образом, сеть Ethernet имеет физическую топологию "звезда", а логическую – топологию "шина".

Аналогичны функции концентраторов в сетях с топологией типа "кольцо", где они часто применяются. Концентраторы позволяют более гибко организовывать связь между абонентами, комбинируя кольцевую и звездообразную физическую конфигурации (рис. 4.24). Концентратор просто включается в общее кольцо абонентов, подключенных к нему, т.е. конфигурация информационных потоков при этом не изменяется.

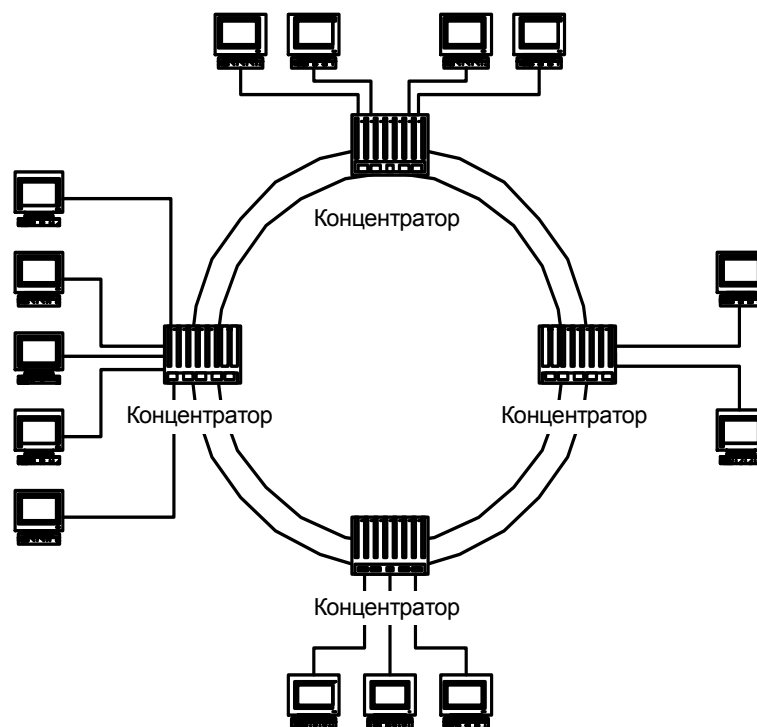


Рис. 4.24. Включение концентраторов в сеть типа "кольцо"

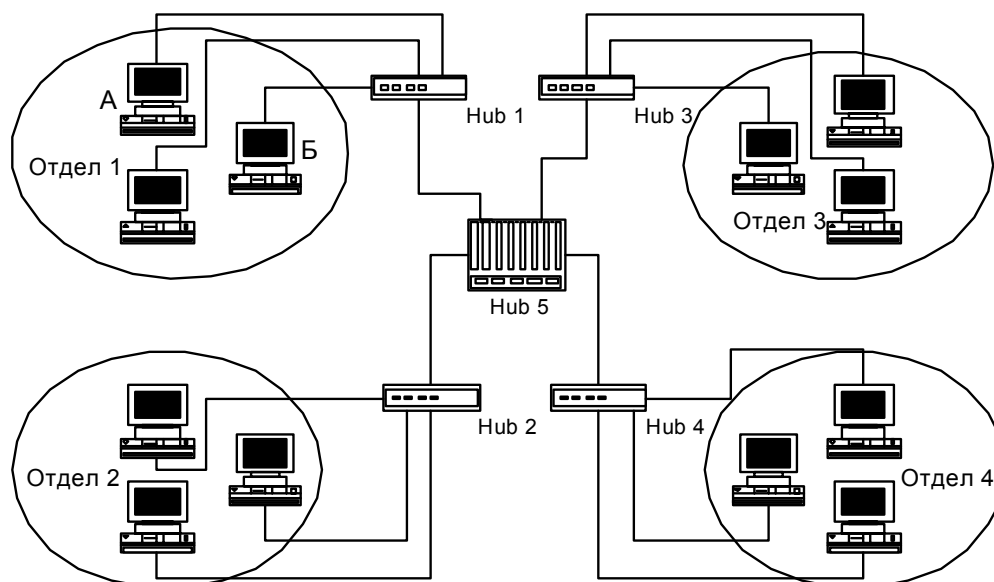
Таким образом, концентраторы всегда изменяют физическую топологию сети, но при этом остается без изменения конфигурация информационных потоков, т.е. логическая конфигурация сети.

### Логическая структуризация сети

В большой сети возникает неоднородность информационных потоков: сеть состоит из множества подсетей рабочих групп, участков, цехов, филиалов предприятия и других административных образований. При этом можно выделить группы узлов, активно общающихся между собой и изредка обменивающихся сообщениями с другими узлами. Проблема перераспределения трафика между различными фрагментами сети не решается с помощью физической структуризации. Сети с типовой топологией чаще всего просто неспособны создавать выделенные, обособленные по интенсивности трафика группы. В качестве примера рассмотрим структуру, представленную на рис. 4.25.

Предположим, в отделе 1 компьютер А захотел передать данные компьютеру Б. При передаче сигнала передаваемые данные поступают на порт концентратора 1, к которому подключен компьютер А, и воспроизводятся без изменения на всех остальных портах. Соответственно, на одном из портов концентра-

тора 5 появится сигнал, т.к. он подключен к одному из портов концентратора 1, который, в свою очередь, воспроизведет его на всех своих портах. Таким образом, все концентраторы данной сети воспроизведут данный сигнал на всех своих портах. Получение сигнала означает, что линия занята. Все компьютеры сети будут ожидать освобождения линии, хотя данные передаются только в пределах одного отдела.



**Рис. 4.25.** Физическая структуризация сети с помощью концентратора

Решение этой проблемы осуществляется с помощью логической структуризации сети. Логические связи представляют собой маршруты передачи данных между узлами сети.

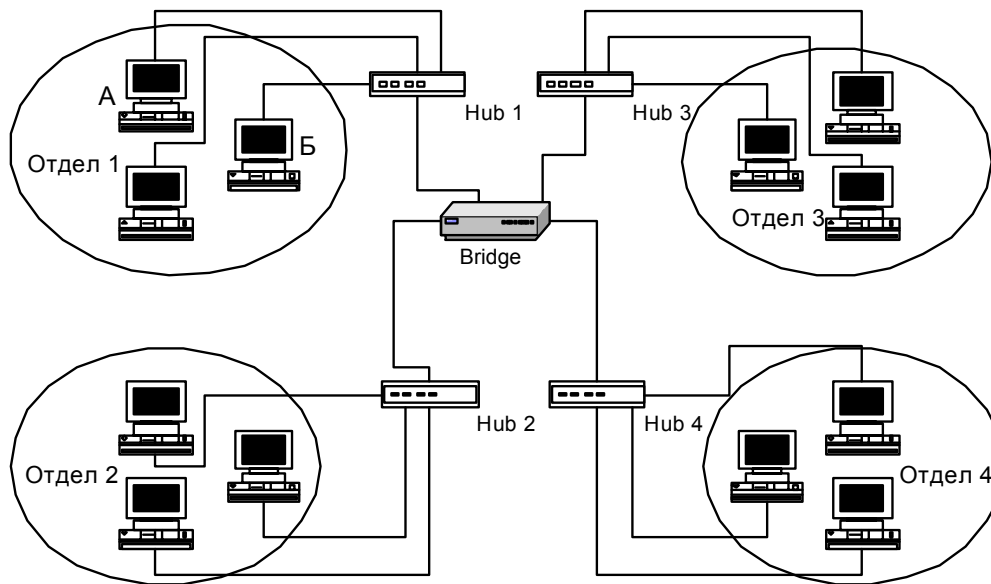
Деление сети на логические сегменты приводит к тому, что нагрузка, приходящаяся на каждый из вновь образованных сегментов, оказывается меньше, чем нагрузка, которую испытывает исходная сеть. Так, решение проблемы сети, представленной на **рис. 4.25**, заключается в том, чтобы данные, направляемые рабочей станции внутри отдела 1, не выходили за пределы этого же отдела. Вариант решения данной проблемы представлен на **рис. 4.26**, где для логической структуризации сети используется мост.

**Мост (Bridge)** делит разделяемую среду передачи данных сети на части, называемые логическим сегментами. Мост изолирует трафик одного сегмента от трафика другого, передавая информацию из одного сегмента в другой только в том случае, когда такая передача действительно необходима. Тем самым *мост изолирует трафик одной подсети от трафика другой*, повышая общую производительность передачи данных с сети.

Основное назначение моста – организация обмена между сетями с разными стандартами обмена, например Ethernet, Arcnet, Token Ring и т.д., а также между несколькими сегментами одной сети (**рис. 4.27**). Особенностью работы моста является то, что система передачи данных реализуется средствами физического и канального уровней. При этом адресация в сети осуществляется на ос-

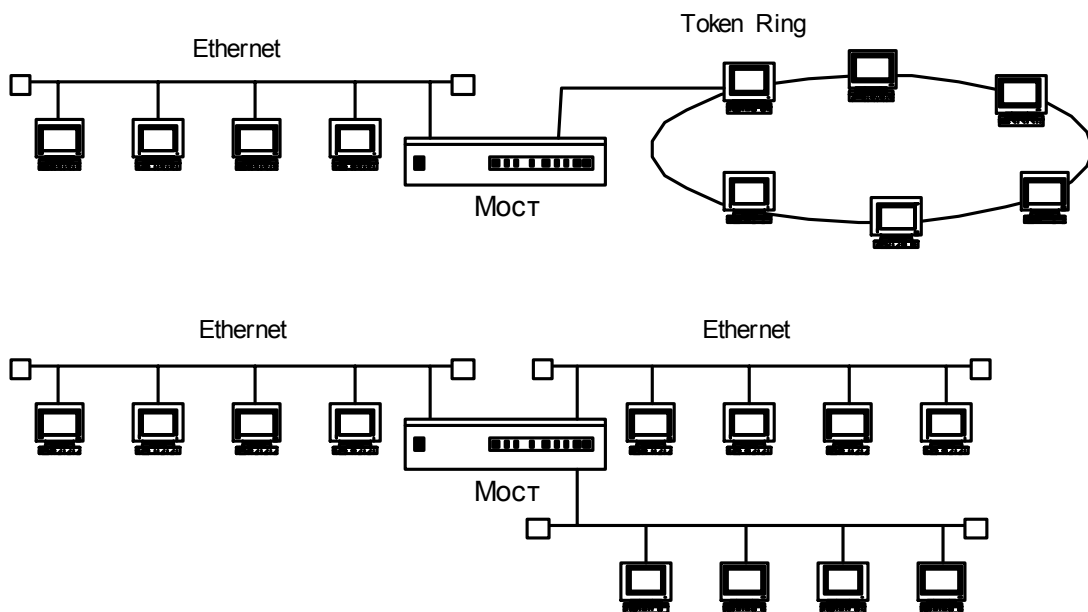


новые аппаратных адресов (MAC-адресов) компьютеров, жестко связанных с сетевым адаптером.



*Рис.4.26. Логическая структуризация сети с помощью моста*

Объединение с помощью моста сетей с одинаковыми протоколами (например, двух Ethernet) отличается от объединения с помощью повторителя или концентратора. В случае каждая из сетей работает со своими собственными пакетами, и только при необходимости через мост приходят пакеты из другой сети, адресованные абонентам данной сети. В результате нагрузка на каждую из сетей оказывается существенно меньше, чем при использовании концентратора. Мост



*Рис. 4.27. Соединение разнородных и однородных сегментов сетей с помощью моста (коммутатора)*

может соединять сети разных топологий.

Передача пакетов осуществляется следующим образом (**рис. 4.28**). Если адреса не указаны в таблице маршрутизации, то мост передает пакеты во все сегменты, если же адресат указан в таблице маршрутизации, мост передает пакет в этот сегмент. Можно сказать, что мосты обладают некоторым «интеллектом», поскольку изучают, куда следует направлять данные. Когда пакеты передаются через мост, данные об адресах компьютеров сохраняются в оперативной памяти моста, который использует эти данные для построения таблицы маршрутизации. Принимая пакет, мост ищет адрес источника в таблице маршрутизации. Если адрес источника не найден, он добавляет его в таблицу. Затем мост сравнивает адреса назначения с базой данных таблицы маршрутизации. В дальнейшем, если мост знает о местонахождении узла-адресата, то он передает пакет ему; если же адресат неизвестен, то мост транслирует пакет во все сегменты и т.д. Таким образом, он просто запоминает адрес отправителя и ставит ему в соответствие порт, а в случае поступления данных для этого адреса, он направляет данные на этот порт.

**Коммутатор (switch)** по принципу работы ничем не отличается от моста.



**Рис. 4.28.** Принцип работы моста

Основное его отличие от моста состоит в том, что он является своего рода коммуникационным мультипроцессором, так как каждый его порт оснащен специализированным процессором, который обрабатывает кадры по алгоритму моста независимо от процессоров других портов. Одним словом, коммутаторы – это мосты нового поколения, которые обрабатывают кадры в параллельном режиме в отличие от классических мостов, обрабатывающих кадры в последовательном режиме. За счет этого общая производительность системы намного выше, чем производительность моста, имеющего один процессорный блок. Посте-

пенно коммутаторы вытесняют из локальных сетей классические однопроцессорные мосты.

Ограничения, связанные с тем, что невозможно создавать замкнутые маршруты, привели к тому, что в ряду коммуникационных устройств появился еще один представитель – маршрутизатор (router).

**Маршрутизаторы (router).** Маршрутизация – это процесс, при помощи которого данные, передаваемые с компьютера в сеть, направляются к компьютеру-адресату в случае, когда последний находится в разных с исходным компьютером сетях. Маршрутизаторы применяются только *в сильно разветвленных сетях, имеющих несколько параллельных маршрутов*. Их функция состоит в том, чтобы выбрать оптимальный путь (маршрут) для каждого пакета. Это делается для избежания чрезмерной нагрузки отдельных сегментов, а также для обхода поврежденных участков. С помощью двух числовых адресов – адреса сети и адреса узла – маршрутизатор однозначно выбирает определенную станцию сети. Он также может выбрать наилучший путь для передачи сообщения абоненту сети, фильтрует информацию, проходящую через него, направляя в одну из сетей только ту информацию, которая ей адресована.

Маршрутизаторы более надежно и более эффективно, чем мосты, изолируют трафик отдельных сегментов сети друг от друга. Маршрутизаторы для выявления принадлежности получателя подсети *используют числовые адреса*, что позволяет структурировать и систематизировать подсети, облегчая возможности администрирования и оптимизации маршрутов прохождения кадров. Маршрутизаторы могут переадресовывать и маршрутизировать пакеты через множество сетей, обмениваясь информацией между различными сетями. Маршрутизатор переправляет или ретранслирует пакеты, основываясь на коммуникационных путях, описанных в его *таблице маршрутизации*. Таблица помогает маршрутизатору определить адреса назначения для поступающих данных. Она включает следующую информацию:

- все известные сетевые адреса;
- способы связи с другими сетями;
- возможные пути между маршрутизаторами;
- стоимость передачи данных по этим путям;
- выбор наилучшего маршрута для данных, сравнивая стоимость и доступность различных вариантов.

Существуют два типа таблиц маршрутизации: статические и динамические. Системные администраторы должны создавать и обновлять статические таблицы маршрутизации вручную, поскольку таблицы не могут измениться без определенного вмешательства. Динамические таблицы маршрутизации создаются и поддерживаются автоматически при помощи протокола маршрутизации.

*Отметим различие между мостами (коммутаторами) и маршрутизаторами.*

1. Мост (коммутатор) распознает только локальные адреса (аппаратные адреса, MAC-адреса), а маршрутизаторы распознают числовые адреса сетей (IP адреса).

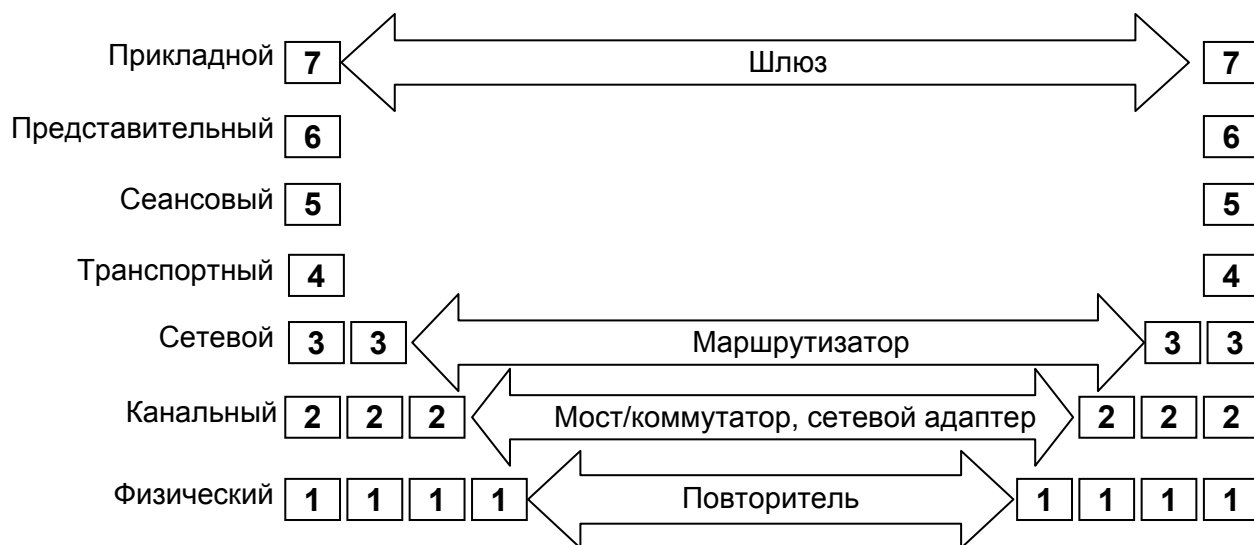
2. Мост (коммутатор) распространяет пакеты с неизвестным ему адресом получателя по всем направлениям, все пакеты с известным адресатом передает только через соответствующий порт, а маршрутизатор фильтрует адреса.

3. Маршрутизаторы более надежно и более эффективно, чем мосты, изолируют трафик отдельных частей сети друг от друга.

**Шлюз (gateway)** – это устройство, служащее для соединения совершенно разных сетей, например локальных сетей с глобальными или локальных сетей с большими ЭВМ, использующими совершенно другие и, что принципиально, абсолютно несовместимые протоколы обмена. При этом приходится преобразовывать весь поток информации, включая код, форматы, методы управления и т.д. Это еще более сложные и, следовательно, гораздо более дорогие устройства, чем маршрутизаторы.

Шлюз выполняет свои функции на уровнях выше сетевого. Он не зависит от используемой передающей среды, но зависит от используемых протоколов обмена данными. Обычно шлюз выполняет преобразования между двумя протоколами. С помощью шлюзов можно подключить локальную вычислительную сеть к главному компьютеру, а также к глобальной сети.

Если вернуться к модели открытых систем, то повторители (репитеры) и концентраторы связывают две сети (или два сегмента сети) на физическом (первом) уровне модели, мосты (коммутаторы) – на канальном (втором), маршрутизаторы – на сетевом (третьем) уровне, а шлюзы – на более высоких уровнях. На **рис. 4.29** показано соответствие функций различных коммуникационных устройств уровням модели OSI.



**Рис. 4.29.** Соответствие функций различных устройств сети уровням модели OSI

#### 4.7. Методы передачи данных в компьютерных сетях

Основными используемыми методами передачи информации являются методы коммутации каналов, сообщений и пакетов.

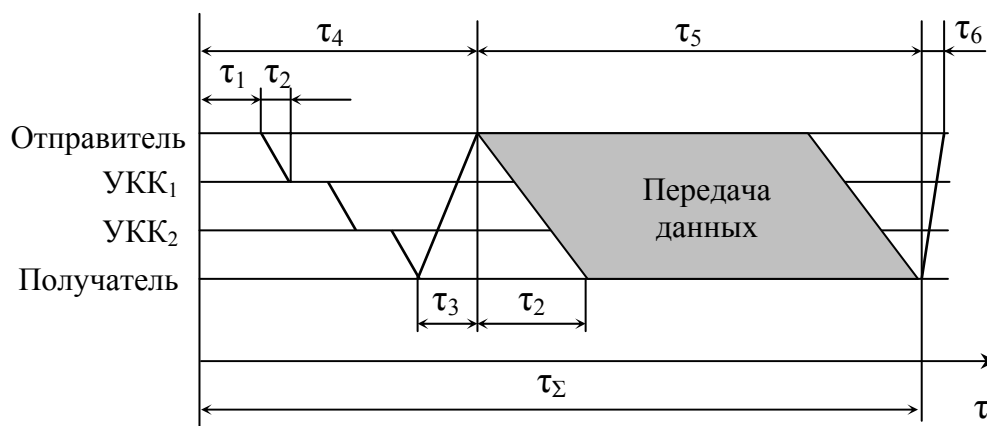
### *Коммутация каналов*

Это наиболее простой и естественный способ передачи данных между двумя абонентскими системами. Он базируется на реализации *физического соединения между ними на время сеанса передачи информации путем образования составного канала из последовательно соединенных каналов.*

По своей сути сети коммутации каналов подобны телефонным сетям коллективного пользования, на базе которых они, как правило, и реализуются. Всемирная сеть, которую использует телефон и которая может быть доступна компьютерам, называется общедоступной коммутируемой телефонной сетью (PSTN, Public Switched Telephone Network). Изначально она была создана для передачи речи, поэтому она обладает низкой скоростью, а для связи по коммутируемым аналоговым линиям необходимы модемы. Модем – это устройство, которое позволяет компьютерам обмениваться данными по телефонной линии. Модем на передающей стороне преобразует цифровые сигналы компьютера в аналоговые и посылает их по телефонной линии. Модем на принимающей стороне преобразует входящие аналоговые сигналы в цифровые и передает их компьютеру-получателю.

При этом связь между отправителем и получателем устанавливается путем отправки отправителем соответствующего сообщения, которое передается по сети передачи данных от одного узла коммутации канала к другому и управляет коммутацией каналов связи, как бы прокладывая путь от отправителя к получателю. После образования физического соединения из пункта назначения отправителю передается ответное сообщение, подтверждающее наличие требуемого соединения. Затем осуществляется передача информации, ради которой был создан канал передачи данных. На время сеанса обмена информацией составной канал полностью оказывается недоступным для других абонентов. После завершения передачи отправитель информации вырабатывает соответствующее управляющее сообщение, которое передается по составному каналу, управляя его разъединением.

Процедура передачи и временных задержек в сетях коммутации каналов на базе телефонных сетей коллективного пользования, коммутации сообщений и пакетов проиллюстрирована на **рис. 4.30**. Общее время сеанса при коммутации каналов складывается из времени коммутации, времени передачи и времени завершения сеанса связи. Следует отметить, что процесс коммутации может оказывать существенное влияние на время сеанса, например, при использовании низкоскоростных узлов коммутации каналов.



**Рис. 4.30.** Передача данных в сетях с коммутацией каналов:

УКК – узел коммутации каналов;

$\tau_1$  – задержка подключения;  $\tau_2$  – распространение сигнала;

$\tau_3$  – передача ответных сообщений;

$\tau_4$  – время коммутации канала;  $\tau_5$  – передача данных;

$\tau_6$  – разъединение канала;  $\tau_\Sigma$  – общее время передачи данных

В рамках сетей коммутации каналов могут организовываться так называемые выделенные каналы (выделенные аналоговые линии), которые постоянно закоммутированы на требуемый срок. В отличие от коммутируемых линий, которые нужно организовать для каждого сеанса, выделенные аналоговые линии обеспечивают готовый к немедленному использованию коммуникационный канал.

**Цифровая связь.** Линия цифровой службы передачи данных (DDS) обеспечивает связь, практически свободную от ошибок. В этом случае используется цифровая связь, она не нуждается в модемах. Вместо этого данные от моста или маршрутизатора DDS передаются через устройство, которое называется устройством обслуживания канала – устройством обработки данных (CSU/DSU). Оно преобразует стандартные цифровые сигналы, генерируемые компьютером, в биполярные цифровые сигналы, применяемые для синхронной связи.

### Коммутация сообщений

Передача информации осуществляется без образования физического соединения между пунктами отправления и получения информации. Между ними устанавливается *виртуальное (логическое) соединение, а физический канал устанавливается локально между смежными узлами коммутации и только на время передачи между ними данных*. При этом информация представляется и передается в виде блока данных, целиком содержащего все сообщение. Заголовок блока данных содержит адреса отправителя и получателя информации, а также другую управляющую информацию, необходимую для достоверной передачи сообщения между абонентами. *Передача данных между абонентами осуществляется с промежуточным запоминанием их в узлах коммутации*. Поступившее в узел коммутации сообщение запоминается в буферном запоминающем устройстве и при наличии свободного канала связи в направлении адресата пе-

редается по этому каналу в следующий узел. Такие узлы, осуществляющие промежуточное хранение и управление передачей сообщений, называются *узлами коммутации сообщений*, а сети передачи данных, использующие данный способ коммутации, получили название *сетей коммутации сообщений*.

Таким образом, сообщение последовательно передается от одного узла коммутации к другому, занимая в каждый период времени только канал передачи данных между смежными узлами. Остальные каналы на пути следования сообщения могут использоваться для других целей.

### *Коммутация пакетов*

Коммутацию пакетов можно рассматривать как дальнейшее развитие коммутации сообщений, при котором сообщение-пакет имеет строго ограниченную длину. Передача данных по сетям с коммутацией пакетов напоминает перевозку огромного количества товара грузовиками вместо транспортировки его на одном поезде. Если опрокинется какой-нибудь грузовик с товаром, навести порядок в этом случае будет проще, чем перегрузить сошедший с рельсов поезд.

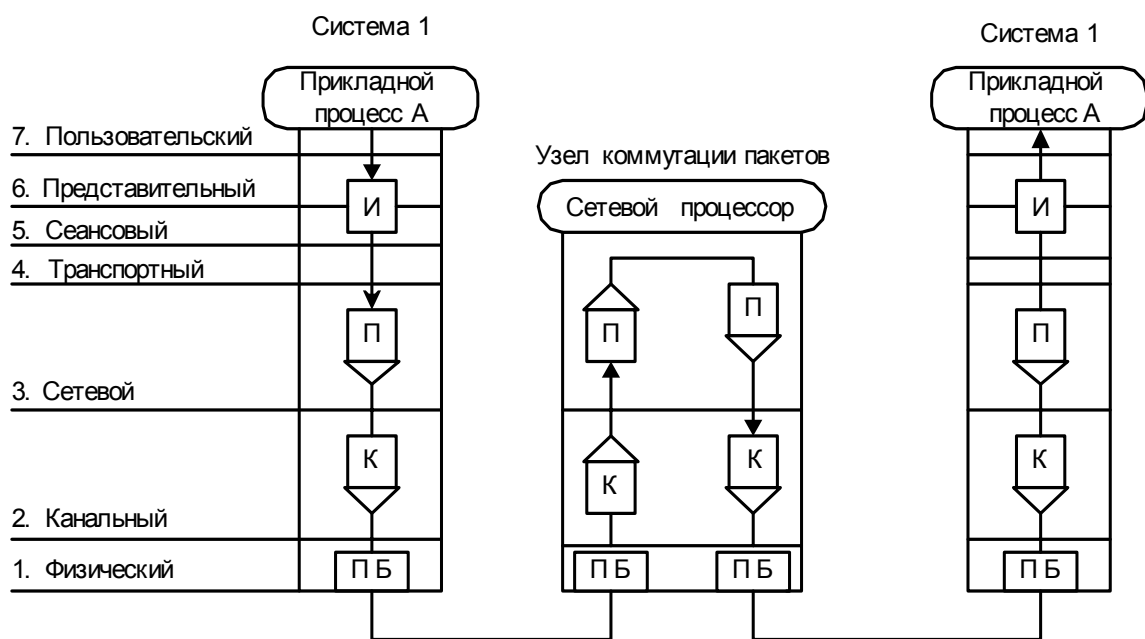
Фиксированная длина пакетов предполагает разбиение длинных сообщений на несколько пакетов. Каждый пакет снабжается адресом отправителя, адресом приемника и номером самого пакета в сообщении. Сформированные пакеты *передаются в сети как независимые сообщения* и, поступая в узел коммутации пакетов, накапливаются в буферах каналов связи. После этого они передаются в выходной буфер, где накапливаются пакеты различных сообщений, выдаваемые на быстрый канал связи для передачи в соседний узел связи, и т.д. В пункте назначения из поступающих пакетов формируется сообщение.

Описанный выше режим передачи пакетов между двумя конечными узлами сети предполагает независимую маршрутизацию каждого пакета. Такой режим работы сети называется *дейтограммным*, и при его использовании коммутатор может изменять маршрут какого-либо пакета в зависимости от состояния сети, в частности работоспособности каналов.

Однако отсутствие логического канала может привести к нарушению порядка поступления пакетов к адресату. Поэтому для передачи больших сообщений используется способ *виртуальных каналов*, при котором все пакеты следуют по каналам и к тому же по заранее установленному маршруту.

В узлах коммутации пакетов реализуются три нижних уровня эталонной модели взаимодействия открытых систем, на которых, соответственно, используются три типа протокольных блоков данных: последовательность бит, кадр и пакет. На **рис. 4.31** показана последовательность преобразования протокольных блоков данных при передаче их по сети коммутации пакетов. На верхних уровнях систем протокольный блок данных рассматривается как некоторый информационный блок, который на сетевом уровне «упаковывается» в пакет. Сформированные на сетевом уровне пакеты передаются на канальный уровень, где к пакету добавляется служебная информация, необходимая для выполнения функций канального уровня, в результате чего формируется кадр. На физическом уровне кадр представляется последовательностью бит, которая в виде физических сигналов поступает в канал передачи данных. При приеме информации

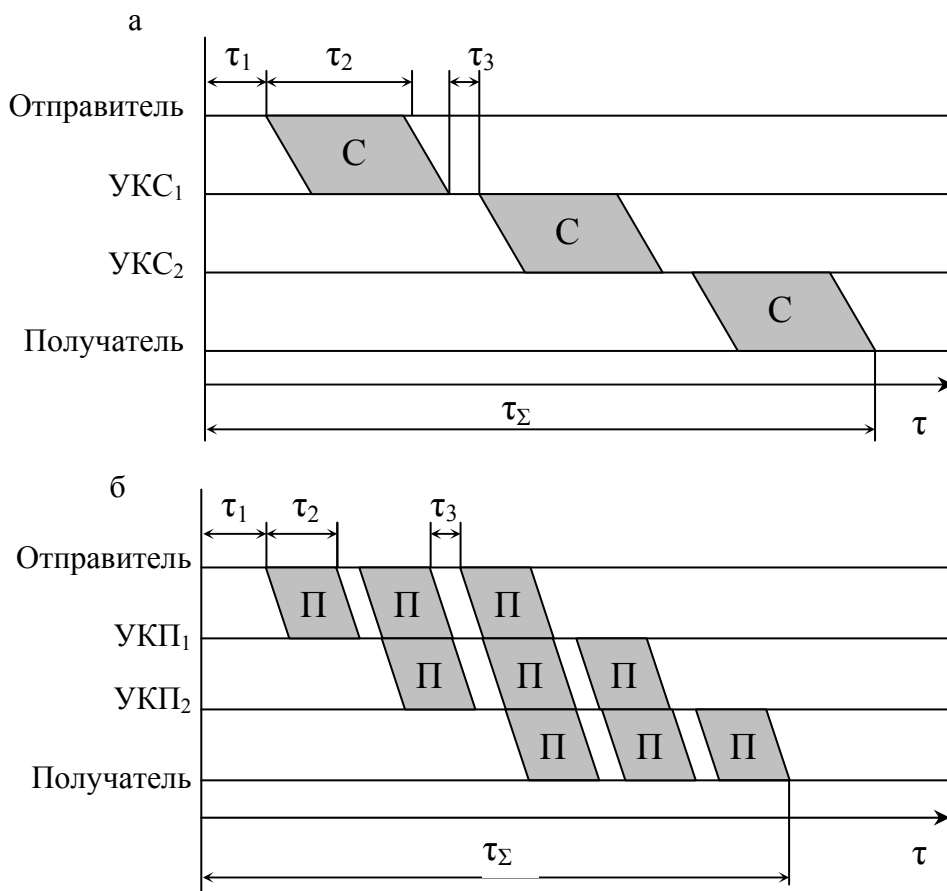
происходит обратный процесс: полученные биты группируются в слова, из которых формируется кадр. На канальном уровне содержимое управляющего поля кадра используется для выполнения процедур канального уровня, а содержимое поля данных в качестве пакета данных передается на сетевой уровень. Управляющее поле пакета формирует сетевой процесс в данном узле коммутации. В дальнейшем пакет преобразуется в кадр, содержащий обновленные адреса и соответствующие значения управляющих полей. Сформированный таким образом кадр данных передается на физический уровень и затем в следующий узел коммутации или абонентскую систему. Таким образом, большие массивы информации передаются несколькими пакетами. Однако, в отличие от коммутации сообщений, операция сборки-разборки осуществляется только в абонентских системах, что упрощает структуру промежуточных узлов сети.



**Рис. 4.31.** Передача данных через сеть коммутации пакетов:  
 ПБ – последовательный бит; К – кадр; П – пакет;  
 И – информационный блок

Естественно, что за счет дублирования управляющей информации в каждом пакете общий объем передаваемого сообщения увеличивается, однако общее время передачи всего сообщения даже сокращается. Это обусловлено тем, что вводимое ограничение на длину пакета позволяет сократить объем запоминающих устройств узлов коммутации и тем самым время пребывания пакетов в узле коммутации, что сокращает общее время передачи сообщения в целом. Как видно из **рис. 4.32**, при коммутации пакетов по сравнению с коммутацией сообщений время передачи информации сокращается и за счет того, что осуществляется одновременная передача нескольких пакетов одного сообщения.





**Рис. 4.32.** Задержка передачи данных в сетях с коммутацией сообщений (а) и пакетов (б):  
 УКС – узел коммутации сообщений;  
 УКП – узел коммутации пакетов;  
 П – пакет; С – сообщение;  
 $\tau_1$  – время задержки;  $\tau_2$  – время передачи;  
 $\tau_3$  – время задержки между сообщениями (пакетами);  
 $\tau_\Sigma$  – время передачи данных по адресу

#### 4.8. Общая характеристика Internet

Internet представляет собой глобальную компьютерную сеть, а само ее название означает «между сетей», т.е. это сеть, соединяющая отдельные сети. Основные вопросы, связанные с технической поддержкой, решаются комитетом Internet Engineering Task Force (IETF Инженерный комитет Internet). Это общественный комитет, который решает технические вопросы, связанные с развитием Internet. Решения и стандарты утверждает общественный комитет Internet Architecture Board – Совет по архитектуре Internet. Существуют национальные и международные сегменты, которые финансируются из различных источников. Internet обеспечивает обмен информацией между всеми компьютерами, которые входят в сеть, будучи подключенными к ней.

Важнейшими компьютерными сетями, которые образуют мировое сетевое пространство вместе с Internet (в том числе совмещаясь с Internet), являются

сети FREENet, NSFNet, BitNet, Ean, USENet, EUNet, NSI, MCIMail, Compu Serve, GLASNet и др. В нашей стране существует и успешно работает несколько региональных компьютерных сетей, среди них можно отметить такие сети, как Relcom, EREEnet, Sprint, Resnet и др.

Любая компьютерная сеть, непосредственно подключенная к Internet и обеспечивающая взаимодействие с другими компьютерными сетями в мире по протоколу TCP/IP, сама является частью Internet. Компьютеры, включенные в сеть Internet и представляющие различные виды сетевого сервиса, называют *хост-компьютерами (узлами)*, или для краткости хостами.

Дадим краткую характеристику существующих возможностей подключения к Internet.

- *Прямое подключение к Internet.* При этом способе для подключения к магистральным каналам Internet используется выделенный компьютер – IP-маршрутизатор. Этот способ обеспечивает полный доступ ко всем услугам, но он достаточно дорог в реализации и поддержке. Такой способ подключения пригоден только для больших организаций. Соединение обычно осуществляется с помощью выделенной телефонной линии. Маршрутизатор становится частью архитектуры Internet и должен оставаться доступным постоянно. Для создания прямого доступа необходимо обратиться в сетевой информационный центр (NIC) с целью регистрации доменного имени и установки маршрутизаторов в магистральную часть Internet.

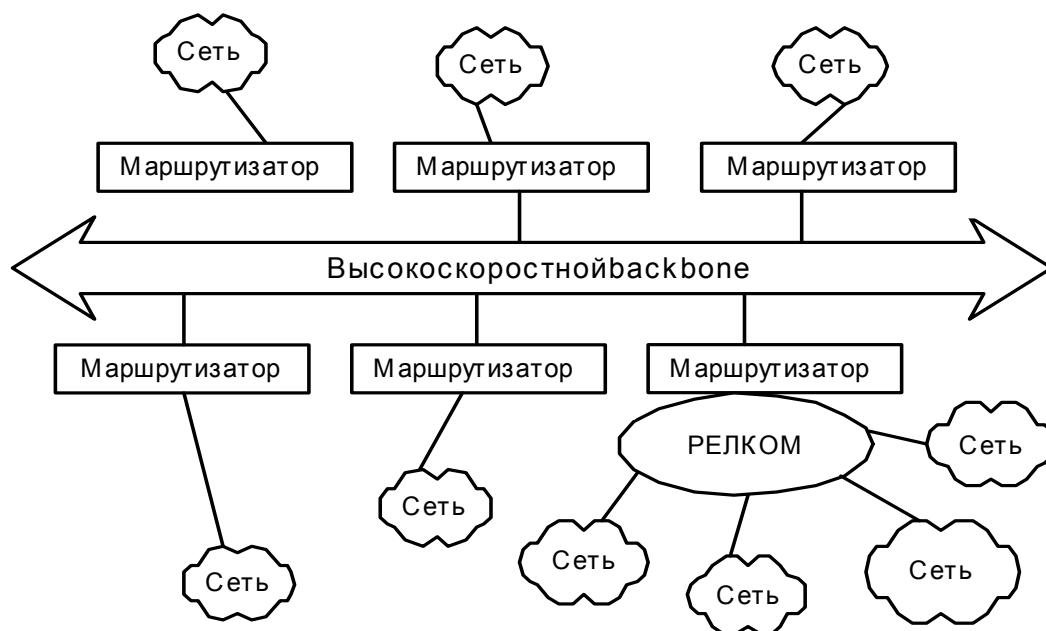
- *Соединение через чей-то маршрутизатор.* Этот способ предполагает получение разрешения на использование чьего-то маршрутизатора для полного доступа ко всем услугам. Такой способ удобен для студентов, но вне университета это может быть не очень удачным решением, т.к. необходимо найти того, кто позволит вам использовать его маршрутизатор.

- *Использование сервис-провайдера.* Ваш компьютер обращается к специальному маршрутизатору компании, которая обеспечивает ограниченный или полный доступ к услугам Internet. Для передачи сигналов по каналам связи вам необходимо заключить договор с организацией, которая осуществит ваше подсоединение. Такую организацию и называют провайдером.

Рассмотрим архитектуру Internet. При этом для наглядности несколько упростим ее структуру и опустим некоторые детали. На **рис. 4.33** схематически представлена структура Internet. В него входит высокоскоростная магистральная сеть передачи данных, называемая backbone. На практике ее образуют выделенные линии, имеющие высокую пропускную способность. На маршрутизаторах установлено сложное программное обеспечение, с помощью которого осуществляется обработка всех сообщений, проходящих через маршрутизатор. Если приходит сообщение, адресованное сети, к которой подключен маршрутизатор, то это сообщение передается в локальную сеть.

Если сообщение предназначено для другой сети, то оно передается следующему маршрутизатору. Сообщения проходят от маршрутизатора к маршрутизатору по скоростным каналам связи, пока не доберутся до своего адресата. Маршрутизаторы "знают" обо всех маршрутизаторах и сетях, с которыми они взаимодействуют, поскольку они обмениваются друг с другом информацией о маршрутизации и состоянии сети, используя специальный протокол маршрути-

зации. Каждый маршрутизатор соединяет магистральную часть Internet с одной или несколькими сетями. Некоторые компании, например Релком, как показано на **рис. 4.33**, выступают в качестве сервис-провайдера для других компаний и пользователей. В большинстве случаев для малых компаний использование сервис-провайдера, подобного Релком, как уже отмечалось, является более быстрым и дешевым способом, чем самостоятельное решение проблем подключения к Internet.



*Рис. 4.33. Архитектура Internet*

Таким образом, такая архитектура в целом вписывается в трехуровневую иерархию информационных сетей:

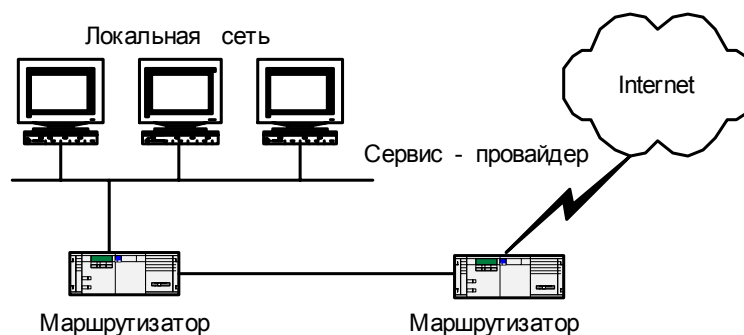
- верхний ключевой уровень глобальных сетей – опорная сеть передачи данных;
- региональные сегменты, оснащенные базовыми шлюзами, в качестве которых используются высокопроизводительные маршрутизаторы, обеспечивающие выбор маршрута передачи данных;
- нижний уровень – локальные сети или отдельные компьютеры на местах.

Обычно подключение локальной сети к Internet заключается в установке маршрутизатора, соединенного с маршрутизатором сервис-провайдера, и в его конфигурировании. Далее к маршрутизатору подключаются компьютеры локальной сети, причем в локальной сети могут работать компьютеры различных платформ с разными операционными системами (**рис. 4.34**).

Заметим, что во многих случаях на маршрутизатор возлагаются функции Firewall (брандмауэр) – фильтрующей системы, защищающей локальную сеть от внешнего мира, что становится сегодня актуальной задачей. С целью повышения безопасности информации можно использовать брандмауэры и прокси-серверы. Подробнее на функциях брандмауэров и прокси-серверов мы остановимся далее.

Передача информации в сетях осуществляется в двух основных режимах: *on line* и *off line*:

- *on line*, или режим «на линии», функционирующий под непосредственным управлением центрального компьютера, т.е. реализующий интерактивный или диалоговый режим работы;
- *off line* – это автономный режим, реализуемый без управления со стороны центрального компьютера сети.



**Рис. 4.34.** Подключение локальной сети к Internet

Самым оперативным и живым способом обмена информацией является работа абонентов в режиме реального времени *on line* (точно так же, как в обычных разговорах по телефону или в селекторных сеансах). Однако по сравнению с речевым обменом по телефону *on line* позволяет транслировать и совместно корректировать таблицы, графическую информацию, массивы цифровых данных, программы для компьютеров и многое другое, причем в объемах, которые не представляются мыслимыми для телефонной, телексной связи.

Следует особо отметить, что в настоящее время даже для небольших, локальных сетей стали очень широко использоваться существующие технологии Internet. Сеть, которая работает по технологии сети Internet, получила название *Intranet* (внутренняя сеть). Таким образом, *Intranet-сети* – это лишь новое применение той же технологии, которая лежит в основе Internet. *Intranet* – это сеть, в которой для распространения информации используются средства, программное обеспечение и протоколы Internet. Благодаря тому, что *Intranet-сети* основаны на открытых стандартах, для них не существует проблем с совместимостью аппаратных и программных платформ, которые заставляли разработчика создавать отдельные приложения для каждого типа операционной системы. Пользователи *Intranet-сети* пользуются стандартными сетевыми браузерами, следовательно, разработчики могут сосредоточиться на построении приложений, работающих на стороне сервера. Сеть *Intranet* можно соединять с Internet, но можно и не делать этого, она может функционировать локально. Следует особо отметить, что именно на основе технологии сети *Intranet* работают современные компьютерные сети на многих крупных металлургических предприятиях России.

## 4.9. Система адресации в Internet (Intranet)

Каждый компьютер, подключенный к Internet, должен быть уникально идентифицирован. С этой целью для каждого компьютера установлено два адреса: цифровой IP-адрес и доменный адрес.

### *Цифровой адрес (IP-адрес)*

Цифровой адрес представляет собой основной тип адресов – это просто число, однозначно определяющее TCP/IP узел в Internet. В сетях TCP/IP узлом называется любой компьютер, имеющий сетевой интерфейс, настроенный на использование TCP/IP.

IP-адрес включает в себя два компонента: *адрес сети и номер узла*. Адрес сети обозначает конкретную сеть (или сегмент), в которой физически находится компьютер (узел). Этот адрес должен быть уникален во всей TCP/IP-сети, вне зависимости от того, является сеть глобальной TCP/IP-сетью или это просто небольшая локальная сеть предприятия, в которой реализован протокол TCP/IP. Адрес сети используется для передачи информации на нужный сетевой интерфейс маршрутизатора.

Компьютер оперирует IP-адресами, представленными в двоичном формате. Поскольку людям сложно манипулировать двоичными числами, обычно предпочитается при записи десятичный формат IP-адресов. Рассмотрим пример IP-адреса. В двоичном коде цифровой адрес записывается следующим образом: 11000000.10101000. 00000001.00000000, в привычном – десятичном коде он имеет вид 192.168.1.0. Когда IP-адрес записан в десятичном формате, он состоит из четырех групп цифр по 8 бит, называемых *октетами*, каждая из которых отделена от соседней октеты точкой и может принимать значение в пределах от 0 до 255. Действительно, если все биты октета равны 0, то его десятичное значение равно 0, а если все биты октета равны 1, его десятичное значение будет равно 255. Если бы использовались все 32 бита в IP-адресе, то получилось бы ( $2^{32}$ ), т.е. свыше четырех миллиардов возможных адресов. Однако некоторые комбинации битов зарезервированы для специальных целей, поэтому реальное число адресов намного меньше. Сетевой адрес класса А 127 зарезервирован для диагностических целей.

В классическом TCP/IP числа 0 (все нули в октете) и 255 (все единицы) не могут использоваться в идентификаторах сетей и узлов.

- Если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет.
- Если в поле номера сети стоят только нули, то, по умолчанию, считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет.
- Если все двоичные разряды IP-адреса равны единице, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета.
- Если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным но-

мером сети. Например, пакет с адресом 194.132.17.255 доставляется всем узлам сети 194.132.17.0. Такая рассылка называется широковещательным сообщением.

В Internet IP-адреса координирует InterNIC (Internet Network Information Center, Сетевой информационный центр Internet, США), который делегирует полномочия по распределению адресов региональным и национальным организациям. Этим организациям InterNIC выделяет большие блоки IP-адресов, из которых они затем выделяют адреса отдельным организациям. Полномочия на адреса в России делегированы нескольким сетевым организациям, среди которых основную роль играет Российский НИИ развития общественных сетей. Для получения IP-адреса для локальной сети организации нужно направить соответствующую заявку.

IP-адреса выделяются в зависимости от размеров предприятия и типа его деятельности. Для обеспечения максимальной гибкости IP-адреса выделяются в зависимости от количества сетей и компьютеров в организации и разделяются на классы *A*, *B* и *C*. Структура IP-адреса для сетей различных классов показана на рис. 4.35.

Класс IP-адреса	Структура IP-адреса		Диапазон десятичных значений первого октета	Доступное количество сетей	Доступное количество узлов
A	0	Адрес сети (7 бит)   Адрес хоста (24 бит)	1 – 126	126	16 777 214
B	1 0	Адрес сети (14 бит)   Адрес хоста (16 бит)	128 – 191	16 384	65 534
C	1 1 0	Адрес сети (21 бит)   Адрес хоста (8 бит)	192 – 223	2 097 152	254

**Рис. 4.35.** Классы IP-адресов и соответствующие им идентификаторы сетей и узлов

Адрес сети класса *A* имеет только 7 бит для сетевого адреса и 24 бита для адреса узла. Старший бит первого октета адреса этого класса всегда равен нулю, что позволяет идентифицировать этот класс. Следовательно, класс *A* использует для идентификации сети только первый октет, а три оставшихся октета – для идентификации узла. Это позволяет идентифицировать  $(2^{24}-2)$ , т.е. более 16 миллионов различных узлов в одной подсети. Однако может существовать только  $(2^7-2)=126$  сетей класса *A*. Таким образом, адреса класса *A* предназначены для использования в больших сетях общего пользования, допускающих большое количество номеров узлов.

Класс *B* использует для идентификации сети первый и второй октеты, а два оставшиеся – для идентификатора узла. Два старших бита первого октета всегда равны 10 (один, ноль), что позволяет идентифицировать этот класс сетей. Адрес сети класса *B* имеет 14 бит для сетевого адреса и 16 бит для адреса узла, что позволяет выделить большее количество сетей класса *B* ( $2^{14}-2$ ) = 16 тыс., но с меньшим количеством узлов. Тем не менее 16 бит позволяют идентифицировать ( $2^{21}-2$ ), т.е. более 65 тыс. узлов. Адреса класса *B* используются в сетях среднего размера, например сетях университетов и крупных компаний.

Класс *C* использует для идентификатора сети первые три октета и оставшийся октет – для идентификатора узла. Три старших бита первого октета адреса этого класса всегда равны 110 (один, один, ноль). Сети класса *C* могут иметь максимум ( $2^8-2$ ) = 254 узла, но таких сетей может быть очень много ( $2^{21}-2$ ). Адреса классов *C* используются в сетях с небольшим числом компьютеров. Большинство сетей относятся к классам *B* и *C*.

Тип класса можно узнать по первому числу IP-адреса. Существуют следующие правила для первого 8-битного числа:

- адреса класса *A* – числа от 0 до 127;
- адреса класса *B* – числа от 128 до 191;
- адреса класса *C* – числа от 192 до 223.

Если адрес вашего компьютера 148.15.86.25, то вы знаете, что ваш компьютер находится в сети класса *B*, сетевой идентификатор – 148.15, а уникальный номер вашего компьютера в этой сети 86.25.

Назначение IP-адресов узлам сети даже при не очень большом размере сети может представлять для администратора сети утомительную процедуру. Протокол *Dynamic Host Configuration Protocol (DHCP)* освобождает администратора от этих проблем, автоматизируя процесс назначения IP-адресов. Протокол DHCP работает в соответствии с моделью клиент-сервер. Компьютер, являющийся DHCP-клиентом, посылает в сеть запрос на получение IP-адреса. DHCP-сервер откликается и посылает ответ, содержащий IP-адрес. При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, *называемое временем аренды*, что дает возможность впоследствии повторно использовать этот IP-адрес. При ручной процедуре назначения адресов осуществляет администратор сети.

Всякий раз, когда посылается сообщение какому-либо компьютеру в Internet, IP-адрес используется для указания адреса отправителя и получателя. Конечно, пользователю не придется самому запоминать все IP-адреса, т.к. для этого используется доменная система имен.

### **Символьные доменные имена (DNS, Domain Name System)**

Символьные доменные имена строятся по иерархическому принципу. Эта структура подобна структуре каталогов в компьютере: есть домены самого верхнего уровня, есть вложенные в них домены, которые, в свою очередь, могут содержать другие имена. Имена доменов самого верхнего уровня строго определены.

Существует два типа классификации доменных имен.

*Имена по типу организации.* Такие домены имеют трехбуквенное сокращенное название, например: com – коммерческие организации; edu – учебные заведения; gov – правительственные учреждения и т.п. Например, tutor.sptu.edu. Здесь edu – общий домен для школ, университетов, tutor – поддомен sptu, который является поддоменом edu. Когда какая-либо организация получает доступ к Internet, она регистрирует свое название в специальных регистрационных службах Internet. Выбор названий, расположенных в домене имени левее организации, является прерогативой самой организации. Для пользователей Internet адресами могут быть просто их регистрационные имена на компьютере, подключенном к сети. За именем следует знак @ – знак "коммерческого АТ". Например, nikola@tutor.sptu.edu – пользователь, зарегистрированный под именем nikola на компьютере, имеющем в Internet имя tutor.sptu.edu. В Internet могут использоваться не только имена отдельных людей, но и имена групп. Для обработки пути поиска в доменах имеются специальные серверы имен, которые преобразуют доменное имя в соответствующий цифровой адрес.

*Имена по стране (географическим регионам)* имеют двухбуквенные обозначения для всех стран мира типа .ru – Россия, .us – США и т.п. Далее, уже в рамках данной страны, провайдеры регистрируют свои группы имен – домены. Имя каждого домена отделяется при написании от другого имени точкой, причем имя домена верхнего уровня пишется справа. Самая левая группа символов является именем данного компьютера. Например, адрес ftp.tim.ustu.ru означает файловый сервер (ftp) кафедры "Теплофизика и информатика в металлургии" (tim) Уральского государственного технического университета (ustu) России (ru).

Если предоставление услуг осуществляется через несколько организаций, то полное имя компьютера может состоять из большого числа символов, хотя на практике редко встретишь имена, включающие в себя больше пяти групп.

Каждый домен администрируется отдельной организацией, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Чтобы получить доменное имя, необходимо зарегистрироваться в какой-либо организации, которой InterNIC делегировал свои полномочия по распределению имен доменов. Так, если вы хотите зарегистрировать, например, домен abcd.ru, то вам следует направить заявку в Российский НИИ развития общественных сетей, которой InterNIC делегировал полномочия по распределению имен в домене ru. Официальная регистрация имен домена производится только после выполнения всей предварительной работы по внесению соответствующих записей в базы данных серверов DNS (Domain Name System).

### *Поиск адреса по доменному имени*

Как же используется эта двойная система адресации компьютеров? Процедура преобразования доменного имени узла в IP-адрес называется определением имен. Для определения имен организуется *так называемый список в базе данных серверов DNS (Domain Name System) server*. Необходимо лишь употре-



бить имя на компьютере, который понимает, как обращаться с *DNS*. Вам никогда не придется самим разыскивать адрес.

Определение имен начинается на рабочих станциях. Когда вы пользуетесь именем, например **x.y.z.ru**, компьютер должен преобразовать его в IP-адрес. Для этого он начинает запрашивать помощь у DNS-серверов. DNS-сервер начинает обработку имени с правого его конца и двигается по нему влево, т.е. сначала производится поиск адреса в самой большой группе (*домене*), потом постепенно сужает поиск. Но для начала опрашивается на предмет наличия у него нужной информации местный узел. Здесь возможны три случая:

1. Местный сервер знает адрес, потому что этот адрес содержится в его базе данных. Например, если вы подсоединены к сети Уральского государственного технического университета, то ваш местный сервер должен обладать информацией о всех компьютерах локальной сети этого института.

2. Местный сервер знает адрес, потому что кто-то недавно уже запрашивал тот же адрес. Когда запрашивается адрес, сервер DNS сохраняет его у себя в памяти некоторое время, как раз на случай, если кто-нибудь еще попозже потребует тот же адрес – это повышает эффективность системы.

3. Местный сервер адрес не знает, но знает, как его выяснить.

Как местный сервер может разузнать запрошенный адрес? В его прикладном или системном программном обеспечении имеется информация о том, как связаться с главным сервером имен (корневым сервером). Этот сервер, который знает адреса серверов имен высшего уровня (самых правых в имени), – это уровень государств (ранга домена **ru**). У него запрашивается адрес компьютера, ответственного за зону **ru**. Если ответ содержит несколько имен доменов, то исходный сервер имен при поиске IP-адреса сначала должен соединиться с главным сервером имен для домена **ru**, затем с сервером имен домена **z.ru**, далее **y.z.ru** и, наконец, с сервером имен для **x.y.z.ru**. Указанный процесс показан на **рис. 4.36**.

На самом деле, для повышения эффективности, поиск начинается не с самого верха, а с наименьшего домена, в который входите и вы, и компьютер, имя которого вы запросили. Например, если ваш компьютер имеет имя **x1.y1.z.ru** и вы запрашиваете компьютер с именем **x.y.z.ru**, то опрос начнется (если имя не выяснится сразу) не со всемирного сервера, чтобы узнать адрес сервера группы **ru**, а сразу с группы **z.ru**, что сразу сокращает поиск и по объему, и по времени.

Этот поиск адреса совершенно аналогичен поиску пути письма без надписанного почтового индекса. Как определяется этот индекс? Все регионы пронумерованы – это первые цифры индекса. Письмо пересылается на центральный почтамт этого региона, где имеется справочник с нумерацией районов этого региона, – это следующие цифры индекса. Теперь письмо идет на центральный почтамт соответствующего района, где уже знают все почтовые отделения в подопечном районе. Таким образом, по географическому адресу определяется почтовый индекс, ему соответствующий. Также определяется и адрес компьютера в Internet, но путешествует не послание, а запрос вашего компьютера об этом адресе. И, в отличие от случая с почтой, информация об адресе доходит до вас, как если бы районный почтамт места назначения отправлял вам письмо, любезно уведомляя вас на будущее об индексе, которого вы не изволили знать.

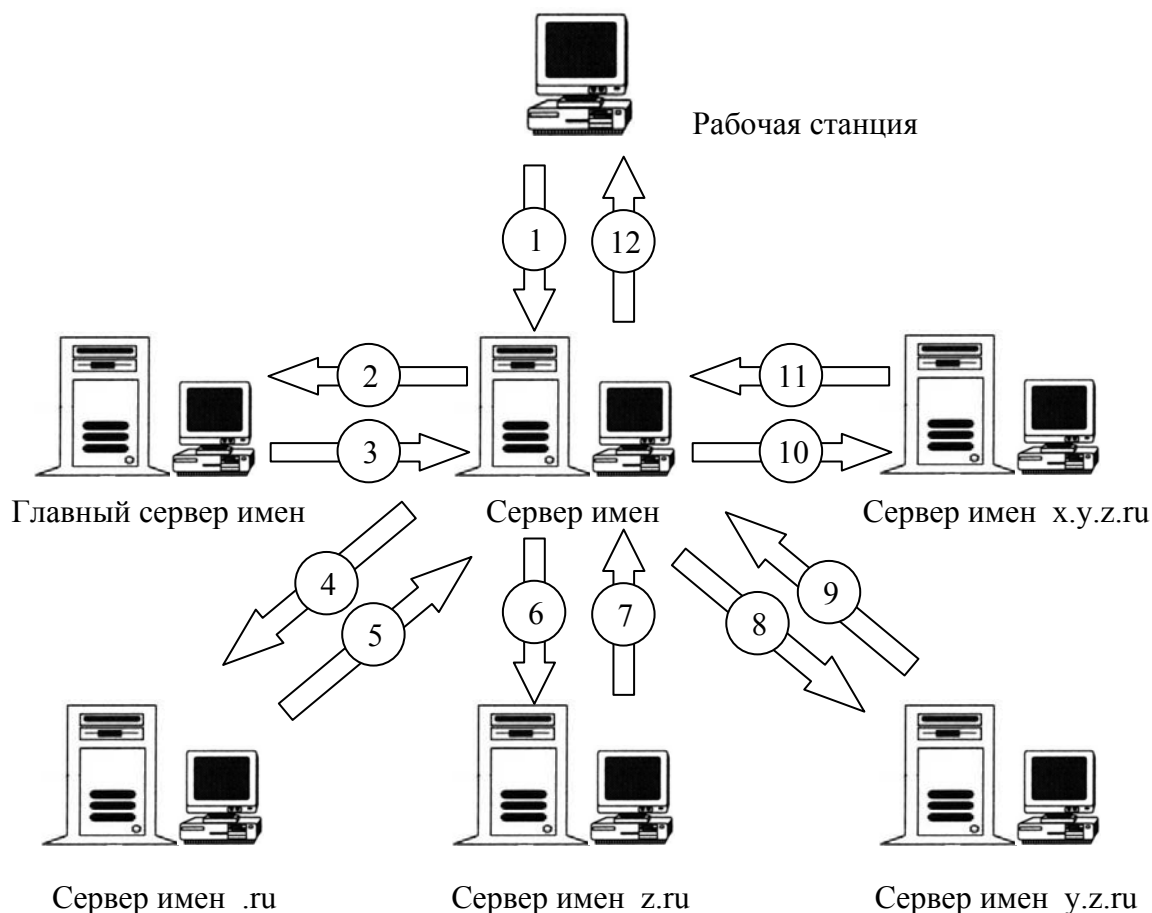


Рис. 4.36. Поиск адресов по доменному имени

### Отображение IP-адресов на локальные (MAC-адреса)

Прежде чем IP-пакет будет отправлен на другой узел, должен быть известен аппаратный (MAC) адрес этого узла. Для определения локального адреса по IP-адресу используется протокол разрешения адреса ARP (*Address Resolution Protocol*). Этот процесс по своей сути очень прост. Для иллюстрации этого процесса рассмотрим два варианта.

**Вариант 1.** Пусть IP-адрес вашего компьютера 192.127.32.25, а IP-адрес получателя (адресата) – 192.127.32.39. Используя полученные вами знания, вы можете определить, что адресат находится в одной локальной сети с вашим компьютером. После ввода запроса к адресату ARP отправляет широковещательный запрос в локальную сеть "Не мог ли владелец IP-адреса 192.127.32.39 сообщить свой MAC-адрес?". Компьютер адресата отправляет ответ, содержащий свой аппаратный MAC-адрес. Когда ваш компьютер получает этот ответ, он добавляет информацию в кэш ARP, чтобы в случае необходимости вновь не заниматься "болтовней", т.е. операцией определения аппаратного адреса получателя. После завершения всего процесса компьютеры готовы к обмену данными.

**Вариант 2.** Пусть IP-адрес получателя (адресата) – 182.123.34.89. В этом случае, как нетрудно заметить, адресат окажется в другой сети. После ввода за-

проса ваш компьютер определяет, что вы пытаетесь обратиться к компьютеру в удаленной сети и поэтому отсылка данных возможна только через маршрутизатор с локальным адресом 192.127.32.1. Маршрутизатор начинает свой собственный процесс преобразования адресов, и так происходит с каждым маршрутизатором по пути следования пакета, пока очередной маршрутизатор не обнаруживает, что для него узел-адресат является локальным.

### *Разделение сетей: подсети и маски подсетей*

После того как мы рассмотрели различные классы доступных IP-адресов, мы установили, что все адресное пространство может быть разделено на три большие группы: классы *A*, *B*, *C*. Однако иногда необходимо дополнительное подразделение блока адресов на подсети, поскольку блоки адресов, выделенные InterNIC организации, могут не соответствовать топологии существующей сети. Как вы помните, каждый сетевой идентификатор соответствует одному физическому сегменту сети. Если вы получаете адрес класса *C*, но в вашей сети два различных физических сегмента, вам желательно далее разделить ваш блок адресов класса *C*. Следовательно, если необходимо провести дальнейшее разделение части адресного пространства, выделенной InterNIC, понадобится передать часть бит, выделенных исходно для идентификатора узла, идентификатору подсети. **Подсеть** – это сеть или идентификатор сети, созданный при помощи переноса нескольких бит из числа IP-адреса, содержащего идентификатор узла, в часть, содержащую идентификатор сети. Конечно, появление дополнительных идентификаторов подсетей, как мы покажем далее, приводит к потере доступных идентификаторов компьютеров, однако снижает трафик сети.

Маскирование сетей позволяет сетевым устройствам и людям определить, какая часть IP-адреса относится к адресу сети и какая – к адресу узла, что позволяет использовать некоторые из идентификаторов узлов для разделения сети на подсети. Маски подсетей по умолчанию для адресов классов *A*, *B* и *C* приведены в табл. 4.1.

Таблица 4.1

**Маски подсетей по умолчанию для адресов классов *A*, *B* и *C***

Класс адреса	Десятичное значение маски	Двоичное значение маски
<i>A</i>	255.0.0.0	11111111.00000000.00000000.00000000
<i>B</i>	255.255.0.0	11111111.11111111.00000000.00000000
<i>C</i>	255.255.255.0	11111111.11111111.11111111.00000000

Маска подсети представляет собой 32-разрядное двоичное число. Единицы в нем обозначают часть, относящуюся к адресу сети, а нули – к адресу узла. Например, маска по умолчанию сети класса *C* 255.255.255.0 представляет число 11111111.11111111.11111111.00000000, поскольку только последний октет используется для идентификации узла. Для адресов класса *A* маской подсети по умолчанию является 255.0.0.0, поскольку только первый октет таких адресов используется для идентификатора подсети. После того как маска сети была определена и установлена, узел использует ее значение для того, чтобы определить,

предназначен пакет для локальной подсети или для удаленной, применяя операцию логического "И". Логическое "И" – это математическая операция, выполняемая над двоичными числами. Результат сравнения двух бит равен 1, если они оба равны 1, и равен 0 в остальных случаях. Пример работы логического "И" представлен в табл. 4.2.

Таблица 4.2

#### Пример работы логического "И"

Двоичная запись IP-адреса	Десятичная запись IP-адреса
11000000.00010010.00101000.01000010	192.18.40.66
Маска сети	Маска сети
11111111.11111111.11111111.11100000	255.255.255.224
Результат логического "И"	Результат логического "И"
11000000.00010010.00101000.01000000	192.18.40.64

Для иллюстрации принципа работы маски сети рассмотрим пример. В аглодоменном производстве существует ряд объектов, территориально распределенных, находящихся на значительном расстоянии друг от друга: доменный цех, агломерационный цех, рудообогатительная фабрика и др. Информация между ними передается по низкоскоростным линиям (типа радиоканал, выделенная телефонная линия, коммутируемые телефонные линии).

Пусть на все нужды информационной системы выделен только один IP адрес сети класса C, например 192.18.48.0 с маской по умолчанию 255.255.255.0. Учитывая, например территориальное расположение объектов, а число компьютеров, необходимо эту сеть разбить на шесть подсетей по 30 адресов. Таким образом, требуется создать как минимум шесть сетевых адресов (подсетей) на основе одного адреса класса C. Вы можете сделать это, расширив маску по умолчанию 255.255.255.0. При этом необходимо использовать для адреса подсетей наименее возможное количество дополнительных бит, так как оставшаяся часть бит необходима для IP-адресов (узлов) компьютеров. Учитывая эти обстоятельства, включим три бита из идентификатора компьютера в идентификатор подсети, получив в результате маску 255.255.255.224.

Число 224 в двоичной записи выглядит как 11100000. При этом имеется восемь возможных комбинаций для адреса подсетей (000, 001, 010, 011, 100, 101, 110, 111). Им соответствуют адреса подсетей, представленные в табл. 4.3. В каждой из этих подсетей можно поместить по 30 адресов узлов. Таким образом, доступное количество адресов составит  $256-16=240$ , так как адреса узлов не могут быть все нули и все единицы. Заметим, что появление дополнительных идентификаторов подсетей приводит к потере допустимых идентификаторов узлов. Так, в нашем случае мы имеем 30 идентификаторов узлов в каждой из 8 подсетей, или всего 240 узлов, вместо 254 узлов при использовании маски по умолчанию.

На доменный цех достаточно отвести две подсети по 30 адресов (IP-адреса 192.18.48.33 – 192.18.48.62; 192.18.48.65 – 192.18.48.94). Все пакеты, относящиеся к этим адресам, "крутятся" только в пределах доменного цеха, и только не относящиеся к этим подсистемам пакеты направляются на маршрутизатор. Аналогичным образом строятся подсети и на других объектах аглодомен-

ного производства: цеха подготовки аглошихты с адресами 192.18.48.97 – 192.18.48.126, аглоцеха с адресами 192.18.48.129 – 192.18.48.158, горнообогажительного производства с адресами 192.18.48.161 – 192.18.48.190. Дополнительно предусмотрена "коммуникационная подсеть" 192.18.48.193 – 192.18.48.222, через которую объединяются все вышеперечисленные подсети (**рис. 4.37**).

Таблица 4.3

**Адреса подсетей сети 192.18.48.0 с маской 255.255.255.224**

Двоичный адрес подсети	Двоичный адрес первого узла подсети	Десятичный адрес подсети	Десятичный адрес первого узла каждой новой подсети
000	00001	192.18.48.0	192.18.48.1
001	00001	192.18.48.32	192.18.48.33
010	00001	192.18.48.64	192.18.48.65
011	00001	192.18.48.96	192.18.48.97
100	00001	192.18.48.128	192.18.48.129
101	00001	192.18.48.160	192.18.48.161
110	00001	192.18.48.192	192.18.48.193
111	00001	192.18.48.224	192.18.48.225

Как же работает маска сети? Как только данный узел, например 192.18.48.56, отправит информацию другому узлу, например 192.18.48.66, он производит операцию логического "И" над маской подсети и IP-адресом узла-адресата, сравнивая полученный результат со своим собственным идентификатором подсети. Если эти два числа совпадают, то узел адресата находится в той же подсети, что и данный узел, и информация может быть отправлена ему непосредственно. Если же эти два числа различаются, то узел адресата находится в другой подсети, и информация пересылается локальному маршрутизатору для отправки в нужную подсеть (сеть). Пример такой процедуры для нескольких IP-адресов представлен в **табл. 4.4**, из которой следует, что адреса 192.128.48.56 и 192.18.48.66 находятся в различных подсетях.

Следует, однако, заметить, что если организация не подключена к Internet, нет необходимости разбивать данный класс адресов на подсети, т.к. можно использовать все адресное пространство ТСП/IP, т.е. можно использовать так много идентификаторов сетей по умолчанию, как необходимо. Так же можно поступать, если сеть соединена с через прокси-сервер или брандмауэр, поскольку эти устройства, как мы увидим далее, скрывают внутреннюю структуру сети, обслуживая все запросы информации через один IP-адрес, т.к. каждый пакет, покидающий сеть, воспринимается как пришедший непосредственно с прокси-сервера, а не с узла, который его отправил.

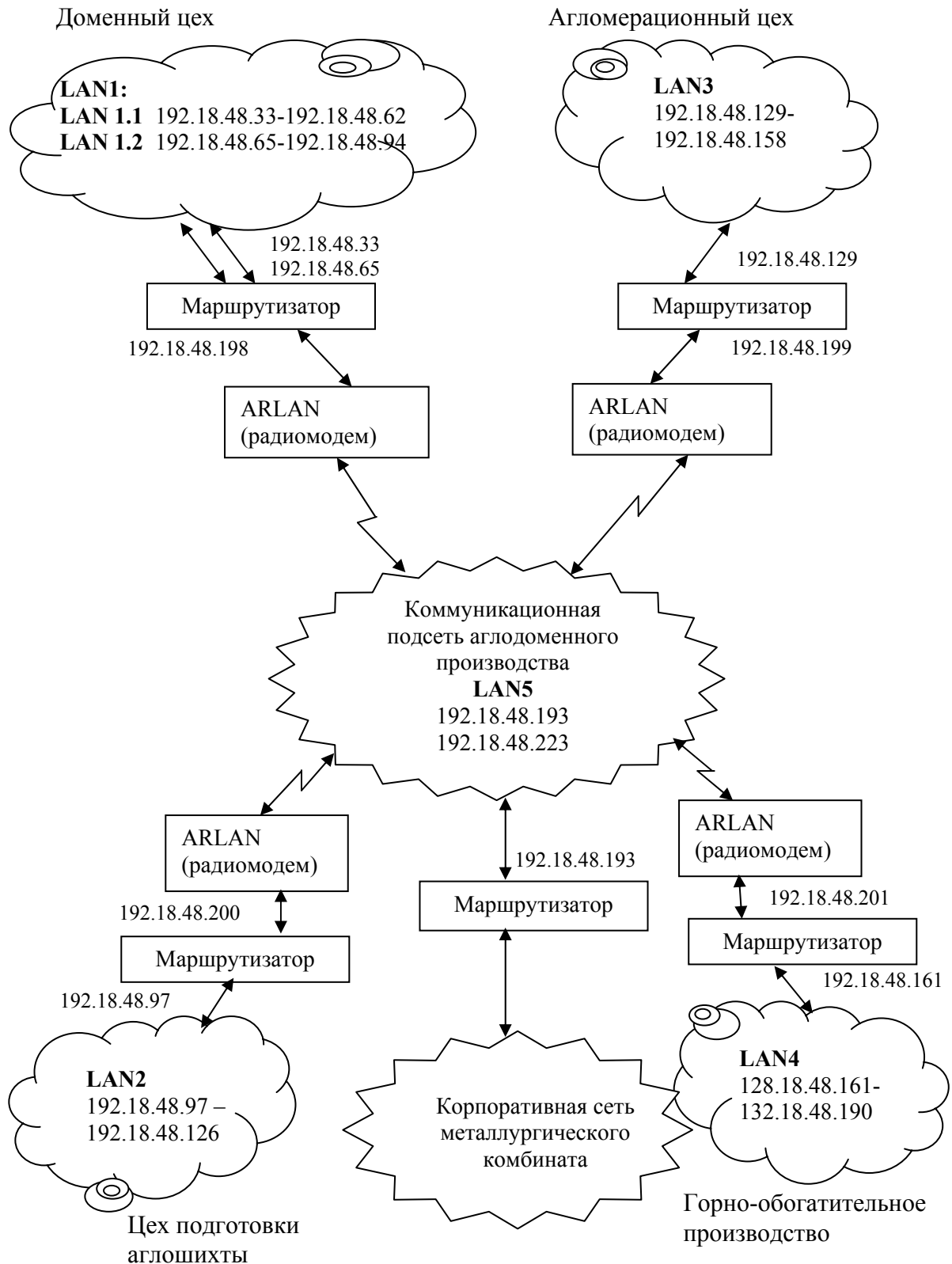


Рис. 4.37. Структура IP-сетей аглодоменного производства

Таблица 4.4

**Работа маски сети 255.255.255.224**

	Десятичный IP-адрес	Двоичный IP-адрес	
		подсети	узла
IP-адрес исходного узла	192.18.48.66	11000000.00010010.00110000	01000010
Маска сети	255.255.255.224	11111111.11111111.11111111	11100000
Результат логического "И"	192.18.48.64	11000000.00010010.00110000	01000000
IP-адрес исходного узла	192.18.48.56	11000000.00010010.00110000	00111000
Маска сети	255.255.255.224	11111111.11111111.11111111	11100000
Результат логического "И"	192.18.48.32	11000000.00010010.00110000	00100000

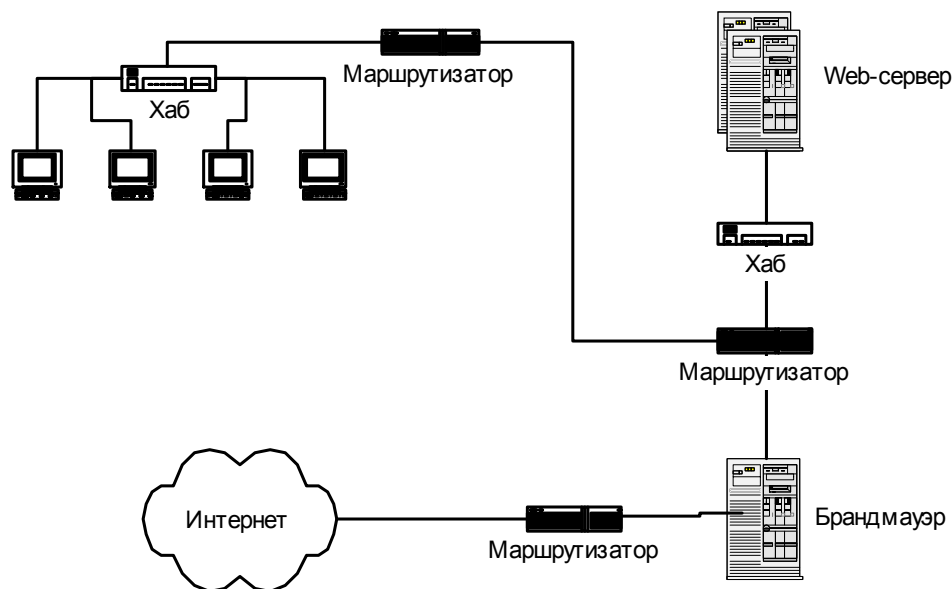
На принципах работы прокси-сервера и брандмауэра мы остановимся позднее, а теперь напомним, что технология Internet, в том числе и маски, широко используется не только в корпоративных сетях крупных металлургических предприятий, но и в локальных сетях (Intranet-сети). В Intranet-сети разработчики могут самостоятельно формировать адреса отдельных сетей, в том числе выбирать и маски подсетей.

#### 4.10. Назначение и функции брандмауэров и прокси-серверов

Если к вашей сети имеет доступ весь мир, кто может гарантировать, что рано или поздно некто не попытается получить кое-что сверх той информации, которую вы сделали общедоступной? Предотвращение несанкционированного доступа внешних пользователей к компьютеру (или ко всей сети) – компетенция прокси-серверов, брандмауэров.

Брандмауэр – это любое устройство или группа устройств, предоставляющее единственную контрольную точку, в которой можно разрешить или запретить дальнейшее перемещение потоков информации между Internet и внутренней сетью (**рис.4.38**). Брандмауэр выполняет практически те же задачи, что и вахтеры на предприятиях. На входе в предприятие вахтер проверяет документы и либо разрешает, либо не разрешает человеку пройти в здание. Если внутренний распорядок предприятия допускает посещения посторонними лицами, вахтер может попросить посетителя расписаться в журнале и предъявить удостоверение личности (например, паспорт или водительские права) и убедиться, что посетитель является именно тем, за кого себя выдает. Эти две функции – ограничение доступа и регистрация доступа – и являются основными функциями брандмауэра в сети.

Брандмауэр ограничивает доступ к сети, используя информацию, заключенную в сетевом пакете. В случае Internet, использующей протокол TCP/IP, эту информацию можно подразделить на следующие компоненты; конечный IP-адрес, конечный IP-порт, исходный IP-адрес и исходный IP-порт.



*Рис. 4.38. Использование брандмауэра для защиты сети*

Уточним понятие IP-порта. Порты – это числа, которые протоколы на транспортном уровне используют для передачи данных соответствующим протоколам более высокого уровня. После того как пакет доставлен в компьютер-получатель, данные необходимо направить конкретному процессу-получателю. Каждый компьютер может выполнять несколько процессов, более того, прикладной процесс может тоже иметь несколько точек входа, выступающих в качестве адреса назначения для пакетов данных. Поступающие пакеты организуются операционной системой в виде множества очередей к точкам входа различных прикладных процессов. Такие системные очереди при использовании протокола TCP называются IP-портами. Следовательно, каждый процесс, использующий протокол TCP, должен иметь номер порта, указывающий расположение определенного приложения или процесса на каждом компьютере. По сути, IP-порт – это расширение IP-адреса, примерно как номер квартиры – расширение номера дома. Сочетание IP-адреса и номера порта называется сокетом.

Номера IP-портов, как правило, соответствуют "хорошо известным протоколам". Например, большинство Web-серверов ожидают запросов на соединение на 80 порту, а большинство почтовых серверов, поддерживающих SMTP протокол, используют 25 порт, номер 21 закреплен за службой удаленного доступа к файлам FTP. Наиболее распространенные TCP/IP приложения и службы используют первые 1023 портов из всех 65535 доступных. Эти применяемые по умолчанию, или "хорошо известные", IP-порты распределены между протоколами, работающими на сервере, в то время как порты, используемые на стороне клиента, выделяются приложению динамически при открытии соединения. Например, при установлении Telnet-сеанса с удаленным узлом пользователь обычно соединяется с "хорошо известным" IP-портом 23. При этом клиент получает динамически выделенный номер порта, который будет использоваться сервером при передаче сообщений на компьютер клиента.



Для сравнения представим себе междугородный телефонный разговор. В данном примере код города, из которого мы звоним, – это исходный адрес, а мы сами – исходный порт. Телефонный номер нашего собеседника соответствует конечному адресу, а сам собеседник – конечному порту. Способ набора номера, то, как мы представляемся и завершаем разговор, будет определяться протоколом. Если телефонная компания нашего абонента предоставляет возможность определения телефонного номера и у абонента имеется соответствующее оборудование, он может узнать номер телефона, с которого мы ему звоним.

Брандмауэр действует таким же образом. Основываясь на информации об исходном и конечном адресах и портах, он может либо принять пакет данных, либо отказаться от его приема. Брандмауэр может также регистрировать выполняемые действия аналогично определителю номера с памятью, чтобы иметь возможность установить, кто предпринял попытку доступа к сети.

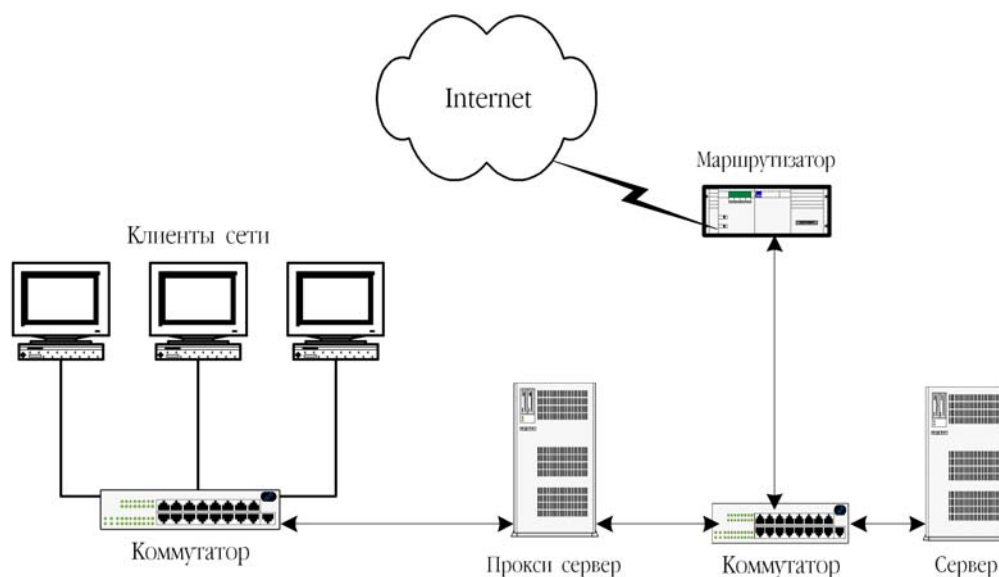
Некоторые брандмауэры не ограничиваются простой фильтрацией пакетов и используют в качестве критерия допуска сетевой сеанс. Для каждого сеанса используется отдельный IP-порт, таким образом, каждому сеансу соответствует уникальный идентификатор. Сеансные брандмауэры способствуют повышению уровня безопасности системы и предоставляют более эффективные возможности по регистрации деятельности клиентов. Все, что может сделать брандмауэр – это снизить риск, связанный с подключением к Internet, но не полностью его исключить. Заметим, что брандмауэр может представлять собой как одно, так и несколько устройств. Так, в качестве брандмауэров используют иногда набор правил фильтрации на маршрутизаторах сети. Одним словом, брандмауэр – это не аппаратное обеспечение, а его настройка (конфигурация).

Прокси-сервер (проху (англ.) – доверенное лицо, посредник) – это сервер, работающий "представителем" клиентов на рабочих станциях в сети и является по сути разновидностью брандмауэра. Однако прокси-сервер обладает одной дополнительной чертой, не присущей сеансным брандмауэрам: он может маскировать адрес и IP-порт конечного компьютера. Иными словами, прокси-сервер – это система, настроенная в качестве посредника между несколькими сетями. Во все пакеты, проходящие через прокси-сервер во внешнюю сеть, будет подставлен адрес прокси-сервера в качестве адреса отправителя. Весь входящий трафик будет поступать на прокси-сервер. Таким образом, прокси-сервер "прячет" друг от друга истинного отправителя и адресата. Это позволяет предотвратить несанкционированный доступ, не препятствуя в то же время доступу к сети внешних пользователей (**рис. 4.39**).

При создании прокси-сервера указывается адрес IP-порта, который он в дальнейшем будет отслеживать. Прокси-сервер будет устанавливать, какой конечный IP-адрес и IP-порт собирается использовать клиент. Если разрешение на соединение отсутствует, подключение не выполняется, если же доступ разрешен, прокси-сервер выполнит подключение к внешнему источнику. Получаемые от внешнего источника данные передаются сетевому клиенту, а любые запросы сетевого клиента передаются внешнему источнику через прокси-сервер.

Чтобы получить доступ к внутренней сети, внешнему пользователю необходимо знать IP-адрес компьютера клиента, IP-адрес прокси-сервера и IP-порт, отслеживаемый прокси-сервером. Кроме того, этот пользователь должен

быть включен в список лиц, которым разрешен доступ к сети, т.е. в IP-адрес внешних клиентов. Если хотя бы один из этих параметров не известен внешнему пользователю, он не сможет подключиться к компьютеру-клиенту, расположенному за прокси-сервером.



*Рис. 4.39. Использование прокси-сервера для защиты сети*

Прокси-серверы можно настроить для работы в обоих направлениях - для доступа внутренних пользователей к внешним и для доступа внешних пользователей к определенным внутренним серверам.

Заметим в заключение, что, кроме обеспечения "барьера безопасности" между сетями, прокси-сервер можно настроить и для кэширования часто запрашиваемой информации.

#### 4.11. Типы сервиса Internet (Intranet)

Существуют несколько основных сервисов компьютерной сети Internet:

- обмен электронной почтой «E-mail»;
- работа во всемирной информационной сети WWW (World Wide Web);
- работа с телеконференциями UseNet;
- сетевые системы передачи информации (копирование и передача файлов по протоколу FTP – File Transfer Protocol);
- распределенная система управления базами данных Gopher;
- WAIS (Wide Area Information Servers Systems) – глобальная информационная служба, предоставляющая распределенную систему обслуживания запросов к нескольким сотням базам данных по различным направлениям науки, техники, образования, культуры;
- справочно-информационные ресурсы LISTSERV;
- справочные книги X.500;

- справочная служба WHOIS;
- информационные ресурсы Mailbase и TRICKLE и др.

Сетевые службы и протоколы основаны на модели клиент-сервер. В этой модели клиентом обозначают процесс или программу, отправляющую запрос на сервер, а сервером – процесс или программу, принимающую запрос, обрабатывающую его и возвращающую результат клиенту. В реальном приложении, использующем архитектуру клиент-сервер, происходит некоторая обработка информации как на стороне клиента, так и на стороне сервера.

Клиент-серверная обработка имеет преимущества перед приложениями на стороне узла или сервера. Во-первых, поскольку сервер выполняет основную обработку запроса и возвращает только результаты, количество передаваемой по сети информации сведено к минимуму. Во-вторых, поскольку клиент и сервер должны поддерживать соответствующие протоколы, то тип операционной системы или аппаратной платформы перестает иметь значение. Сам протокол становится средой общения между клиентом и сервером. Любое программное обеспечение для клиента, например FTP, работающее на любой платформе, может взаимодействовать с любым FTP-сервером, работающим как на той же, так и на совершенно другой аппаратной платформе.

Самой популярной информационной службой Internet является WWW, в сетях Intranet широко используется и служба FTP, поэтому кратко охарактеризуем их возможности.

### *Всемирная информационная сеть WWW (World Wide Web)*

Две основные особенности отличают сеть WWW:

- использование гипертекста;
- возможность клиентов взаимодействовать с другими приложениями Internet.

Название сети WWW (World Wide Web) переводится как "всемирная паутина", под ней принято обозначать совокупность документов во всем мире, которые имеют гипертекстовые ссылки, т.е. ссылки на другие документы, в том числе и на других серверах. Гипертекстовая технология заключается в том, что текст представляется как многомерный, т.е. с иерархической структурой типа сети. Материал текста делится на фрагменты. Каждый видимый на экране компьютера фрагмент, дополненный многочисленными связями с другими фрагментами, позволяет уточнить информацию об изучаемом объекте и двигаться в одном или нескольких направлениях по выбранной связи. При установлении связей можно опираться на различные основания (ключи), но в любом случае речь идет о смысловой, семантической близости связываемых фрагментов. Следуя указанным связям, можно читать или осваивать материал в любом порядке. Объекты не обязательно должны быть текстовыми, они могут быть графическими, музыкальными, с использованием средств мультимедиа и т.п.

Работать с гипертекстами предпочтительнее на рабочей станции клиента, подключенной к одному из Web (WWW)-серверов. Информация на Web-серверах обычно оформлена аналогично страницам "бумажного документа", поэтому принято говорить, что в сети представляется "страница информации".

Отображенный на экране гипертекст представляет собой сочетание алфавитно-цифровой информации в различных форматах и стилях и некоторые графические изображения – картинки.

Для того чтобы пользователь, работающий на компьютере любого типа, видел документ, отформатированный надлежащим образом, разработан универсальный формат *HyperText Markup Language (HTML)* – язык разметки гипертекста. HTML – язык, предназначенный для создания форматированного текста, который насыщен изображениями, звуком, анимацией и гипертекстовыми ссылками на другие документы, разбросанные по всему Web-пространству. HTML-документы представляют собой текстовые поля, в которые встроены специальные команды [56].

Для просмотра страниц HTML-документа необходимы специальные программы – браузеры (WWW browser). WWW browser предназначены для того, чтобы соединиться по Internet с удаленными компьютерами, запрашивать определенные документы, а затем формировать полученные документы для просмотра на локальном компьютере.

Как работают браузеры? Браузеры отправляют запросы и получают информацию, необходимую для того, чтобы изобразить страницу HTML на вашем экране. Полученная информация содержит сам файл HTML, а также все графические, звуковые и видеофайлы, упомянутые в данном файле HTML. Как только данные получены, браузер форматирует текст в соответствии с командами HTML и изображает его на экране вместе с графикой. В тот момент, когда вы щелкаете кнопкой мыши на гипертекстовой ссылке, по каналам Internet отправляется новый запрос на получение другого файла.

Большинство браузеров предоставляют доступ к другим сервисам Internet, в частности FTP-доступа. Это дает возможность считать WWW интегральной службой Internet.

Одной из целей проекта WWW была реализация удобного способа доступа к документам, размещенным на удаленных компьютерах. Для передачи в сети документов, стандартной формой представления которых является гипертекст, был разработан протокол, названный HTTP – HyperText Transfer Protocol (протокол передачи гипертекстов). HTTP – это довольно простой коммуникационный протокол, который учитывает, что передаваемые документы включают гипертекстовые ссылки, т.е. содержат информацию об адресах, по которым могут быть отправлены последние запросы. Заметим, что для просмотра WWW-документов совершенно не обязательно знать что-либо о протоколе HTTP.

HTTP основан на принципе сеансной связи. Каждый сеанс происходит в четыре шага (рис. 4.40):

- 1) клиент устанавливает TCP-соединение с сервером;
- 2) клиент отправляет запрос серверу;
- 3) сервер обрабатывает запрос и возвращает запрашиваемую информацию либо сообщение, что он не может ответить на запрос;
- 4) сервер или клиент закрывает TCP-соединение.

### Сетевые системы передачи информации (копирование и передача файлов по протоколу FTP – File Transfer Protocol)

Для пересылки файлов многих вполне устраивает использование браузера, а не специального приложения – клиента FTP. Можно воспользоваться приложениями – WWW-навигаторами, такими как Windows Internet Explorer, Netscape Navigator. Однако они представляют лишь ограниченные функциональные возможности.

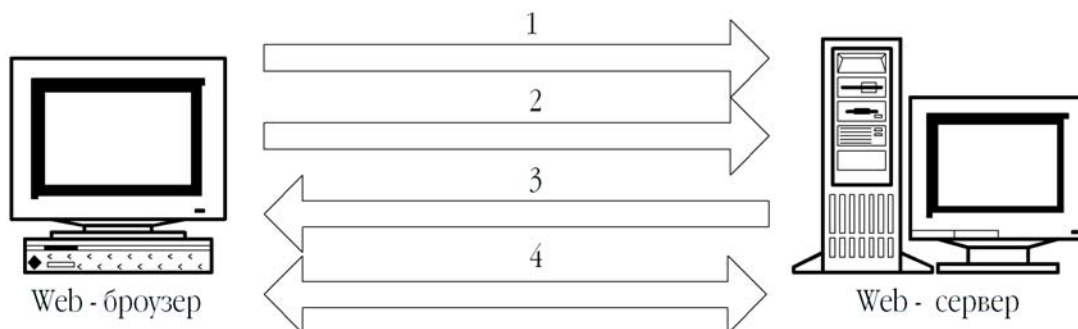


Рис. 4.40. Четыре фазы сеанса HTTP

Сервис FTP специально разработан для того, чтобы позволить вам установить соединение с компьютером в Internet (Intranet), просмотреть файлы, доступные на нем, и скопировать те, которые вам необходимы. На компьютерах, которые используются как серверы FTP, устанавливается специальное программное обеспечение. Оно управляет основными операциями, необходимыми для передачи файлов к пользователю и от него. К таким операциям относятся перемещение, удаление и переименование файлов или каталогов, а также определение, кому именно разрешен доступ к серверу.

Пользователи подключаются к серверу с помощью программ, которые называются *клиентами FTP*. Такие программы-клиенты позволяют использовать *команды FTP*, с помощью которых управляют операциями, выполняемыми сервером, например загрузкой файлов на компьютер пользователя или пересылкой файлов от пользователя на *сервер FTP*.

На **рис. 4.41** показано взаимодействие между клиентом и сервером FTP. Такие программы, как Cnete FTP или WS FTP для компьютеров с операционной системой Windows 95/98/2000, позволяют осуществлять любые операции, начи-

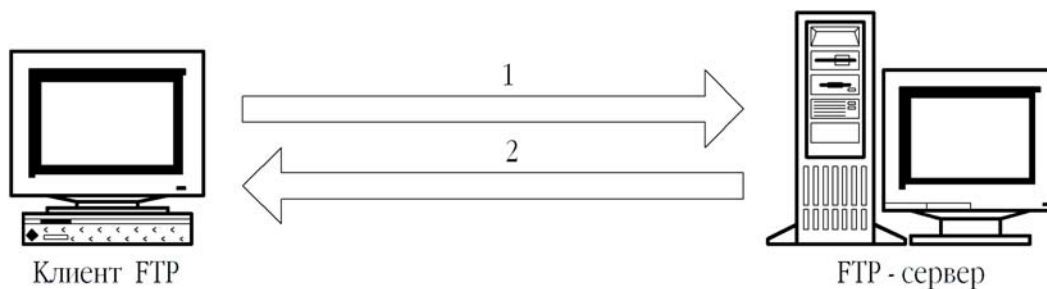


Рис. 4.41. Взаимодействие между клиентом и сервером FTP

ная с загрузки или отправки файлов и заканчивая пересылкой целых каталогов. Отметим в заключение, что большинство FTP-серверов в сетях Internet работает под управлением операционной системы UNIX.

#### 4.12. Контрольные вопросы

1. Что называется компьютерной сетью? Перечислите ее главные компоненты. На какие классы подразделяются компьютерные сети?
2. Какие схемы адресации компьютеров в сети вы знаете? На каких этапах передачи данных используется та или иная схема?
3. В чем заключаются особенности эксплуатации основных типов локальных компьютерных сетей?
4. Перечислите и кратко охарактеризуйте базовые топологии построения компьютерных сетей. Какие топологии используются в локальных и глобальных сетях?
5. С помощью каких физических средств передается информация в компьютерных сетях?
6. Что такое эталонная модель взаимодействия открытых систем (OSI) и какие функции она выполняет? Перечислите основные уровни модели OSI.
7. Для каких целей используют протоколы в компьютерных сетях? Перечислите наиболее распространенные протоколы и задачи, которые они выполняют.
8. С какой целью производится деление данных на пакеты при передаче в компьютерных сетях?
9. Чем отличается асинхронная передача данных от синхронной?
10. Какие методы доступа используются при управлении обменом в сетях типа «звезда», «шина» и «кольцо»?
11. Что называется «стандартной сетевой технологией» и какие их виды наиболее распространены?
12. В чем заключаются особенности работы сетевых операционных систем?
13. Какие средства используют для объединения локальных компьютерных сетей?
14. С помощью каких средств осуществляется защита информации в глобальных компьютерных сетях?
15. Что такое цифровой IP-адрес и на какие классы они подразделяются?
16. Как осуществляется поиск адреса компьютера по доменному имени?

---

ГЛАВА 5. АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ	5-2
5.1. ОБОБЩЕННАЯ СХЕМА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ	5-2
5.1.1. <i>Нижний уровень</i>	5-3
5.1.2. <i>Средний уровень</i>	5-8
5.1.3. <i>Верхний уровень</i>	5-9
5.2. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОМЫШЛЕННЫХ СЕТЕЙ	5-10
5.3. ПРИМЕРЫ КОМПЛЕКТОВАНИЯ НИЖНИХ УРОВНЕЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ	5-23
5.3.1. <i>Программируемые логические контроллеры серии Smart фирмы «PEP Modular Computers»</i>	5-23
5.3.2. <i>Модули серии ADAM 4000 фирмы «Advantech»</i>	5-27
5.3.3. <i>Комплекс технических средств МИК и программных средств MIK\$Sys</i>	5-31
5.3.4. <i>Программируемые логические контроллеры SIMATIC фирмы SIEMENS</i>	5-33
5.4. КОНТРОЛЬНЫЕ ВОПРОСЫ	5-37

## Глава 5. АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

Архитектура современных информационных систем (систем компьютерной автоматизации) технологических процессов представляет собой многоуровневую систему и строится на основе открытых технологий с использованием стандартных модулей, объединенных в сеть помехозащищенным интерфейсом.

Архитектура этих систем строится на принципах:

- стандартности;
- модульности;
- распределенности;
- открытости;
- обеспечения требуемыми метрологическими характеристиками измерительных каналов во всем диапазоне внешних воздействий окружающей среды;
- способности работать в широком диапазоне температур, вибраций, электромагнитных электрических и сетевых помех;
- обеспечения высокой надежности (время наработки на отказ более 100000 часов);
- наличия набора измерительных средств, сопрягаемых с обширным парком датчиков и исполнительных механизмов.

Выбор конкретной структурной схемы информационной системы зависит от множества факторов, в частности от сложности системы, количества датчиков и исполнительных механизмов, скорости обновления информации, требований к надежности, резервированию, противоаварийной защите и т.д. Так, системы, имеющие менее 600–1000 входов-выходов, можно считать малыми, от 600 до 3000 – средними, более 2000–3000 – большими. Очевидно, что такая классификация достаточно условна. В связи с этим дать какой-либо единый рецепт по созданию структурной схемы информационной системы не представляется возможным. В то же время можно представить обобщенную структурную схему современной информационной системы технологических процессов.

### 5.1. Обобщенная схема автоматизированной информационной системы

Обобщенная схема информационной системы представлена на **рис. 5.1**. Это трехуровневая, иерархическая система управления. Для каждого из уровней предполагает индивидуальный подход к разработке программного обеспечения, зависящий от решаемых задач. Заметим, что при выборе базовых программно-технических компонентов необходимо обеспечить, с одной стороны, простоту реализации отдельных составных частей и комплекса системы в целом, а с другой – технологичность сопровождения, модернизации и развития комплекса в процессе эксплуатации.



### 5.1.1. Нижний уровень

Основным функциональным назначением этого уровня является ввод электрических сигналов от датчиков, их первичная обработка, программная реализация законов автоматического регулирования, вывод сигналов управления

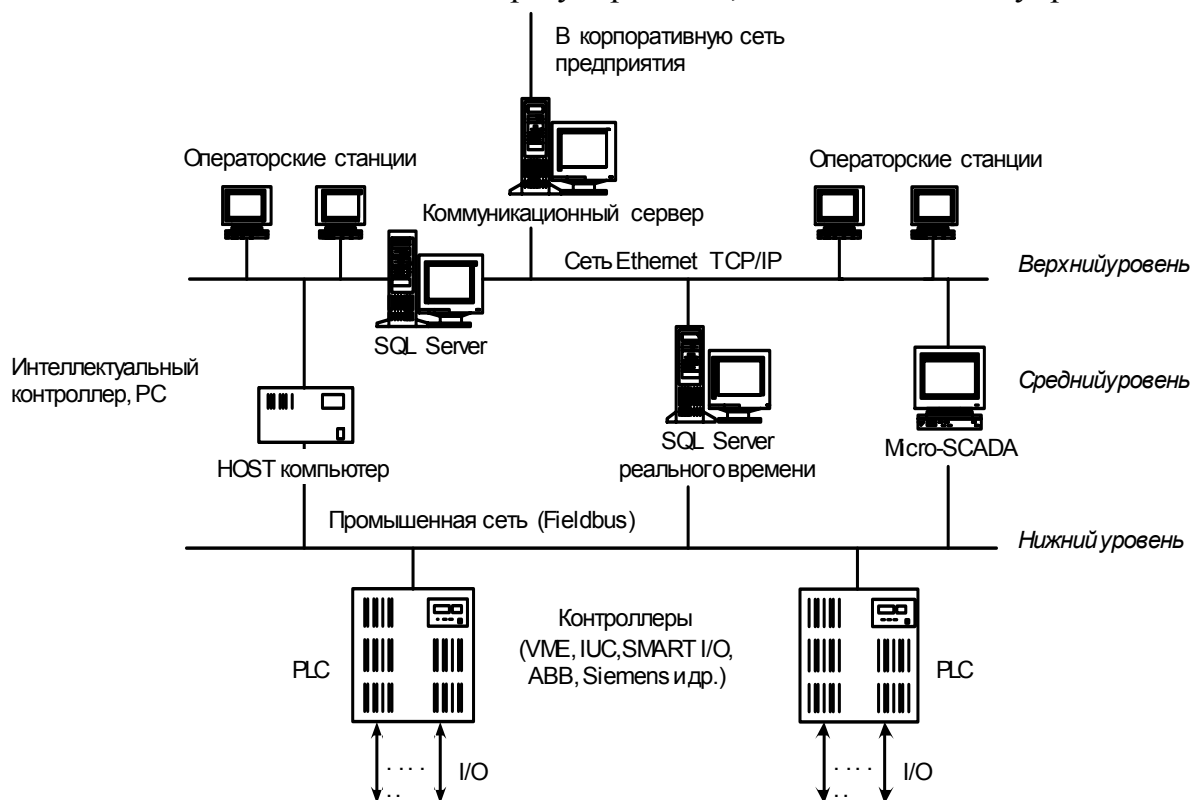


Рис. 5.1. Обобщенная схема информационной системы технологического процесса

исполнительными механизмами и информационное взаимодействие со средним (промежуточным) уровнем.

Заметим, что для реализации этих задач в промышленных системах существуют, в принципе, два конкурирующих направления в технике: промышленные (промышленные) компьютеры PC и программируемые логические контроллеры PLC.

**Индустриальные (промышленные) компьютеры (PC)**, как правило, программно совместимы с обычными офисными PC, но адаптированы для жестких условий промышленной эксплуатации. Они должны работать круглосуточно, подвергаясь воздействию окружающей среды, влаги, пыли, вибрации. В качестве устройств сопряжения с объектом управления данные системы комплектуются дополнительными платами (адаптерами) расширения, которые на рынке представлены в большом разнообразии. В качестве операционной системы в промышленных PC, чаще всего, применяется Windows 95/98/NT/2000, в том числе различные расширения реального времени, специально разработанные для этой операционной системы Windows NT.

**Промышленные контроллеры (PLC – Programmable Logic Controller)** представляют собой специализированные вычислительные устройства, предназначенные для управления процессом (объектом) в реальном времени.

В роли управляющих электронно-вычислительных машин традиционные универсальные РС малопригодны по следующим причинам:

- недостаточным спектром периферийного оборудования и устройств;
- время решения задач в системах управления должно быть строго ограничено, реальным временем, что в настоящее время сложно реализуемо для многих задач на РС;
- работа в необслуживаемом круглосуточном режиме РС невозможна;
- как правило, РС не обладают возможностью обеспечить их высокую технологичность ремонта и обслуживания, что необходимо в АСУ ТП;
- гибкость и необходимость модернизации в процессе эксплуатации, наращивание каналов ввода/вывода, без потерь быстродействия на РС проблематичны;
- принцип распределенности систем на РС реализовать достаточно сложно;
- в универсальных РС большое количество ресурсов тратится на пользовательский интерфейс, который в системах АСУ ТП не всегда необходим;
- в управляющих электронно-вычислительных машинах необходимо специализированное программное обеспечение: ОС и прикладные программы, т.к. целью информационной среды является обеспечение заданной реакции на воздействия, ОС реального времени;
- сменяемость универсальных РС быстрая, а управляющие электронно-вычислительные машины мало меняются, преемственность достигается тем, что в состав магистральных шин закладывается заведомая избыточность.

Промышленные контроллеры имеют вычислительное ядро и модули ввода-вывода, принимающие информацию (сигналы) с датчиков, переключателей, преобразователей, других устройств и контроллеров и осуществляющие управление процессом или объектом выдачей управляющих сигналов на приводы, клапаны, переключатели и другие исполнительные устройства. Современные PLC часто объединяются в сеть, а программные средства, разрабатываемые для них, позволяют в удобной для оператора форме программировать и управлять ими через Host-компьютер, находящийся на более высоком уровне.

Много материалов и исследований по промышленной автоматизации посвящено конкуренции двух направлений.

*Основная тенденция в создании современных автоматизированных информационных систем заключается в следующем: там, где требуется повышенная надежность и управление в реальном масштабе времени, применяют PLC, которые по сравнению с РС общего назначения более устойчивы к зависаниям, обладают детерминированностью поведения, детерминированными реакциями на события реального времени и не подвержены атакам программ-вирусов.*

PLC являются ответственными компонентами и, в значительной степени, определяют свойства всей информационной системы: открытость, модульность, наращиваемость, гибкость, надежность функционирования.

Исследования рынка PLC показывают, что наиболее развитой архитектурой, программным обеспечением и функциональными возможностями обладают контроллеры Siemens, Allen-Bradley, Fanuc Automation, Mitsubishi, SMART, IUC 9300, VME-9300, Omron, МИК, МФК, ТЕКОН и др.

Заметим, однако, что реализация системы на основе оборудования и программных средств, поставляемых "под ключ" приборостроительными фирмами-гигантами, которые применяют собственную уникальную архитектуру измерительно-вычислительных комплексов (Siemens, Allen-Bradley и ряд других), а в дальнейшем монополизируют ее сопровождение, ремонт и модернизацию, не всегда рациональна. Использование закрытых внутрифирменных стандартов, отсутствие исчерпывающей документации на нее и программные средства навсегда привязывают заказчика к фирме-поставщику и, в конечном счете, требуют существенно больших, чем в любом другом варианте, суммарных затрат на создание и эксплуатацию информационных систем.

К аппаратно-программным средствам этого уровня обычно предъявляются жесткие требования по надежности, времени реакции на исполнительные устройства, датчики и т.д. Работоспособность указанных устройств в промышленности должна обеспечиваться в относительно жестких условиях эксплуатации (например, повышенные температуры, сильная загрязненность помещений, токопроводящая пыль, наличие агрессивных газов) при непрерывном круглосуточном режиме работы.

Для обеспечения надежности работы аппаратуры этого уровня обычно применяются специальные технологические решения. В частности, в ней должны отсутствовать механические элементы – вращающиеся диски, вентиляторы и т.п., ухудшающие надежность работы аппаратуры. Для охлаждения элементов используются пассивные системы охлаждения – радиаторы естественной конвекции, а ПЗУ реализовано на флэш (Flash)-дисках. Что такое флэш-ПЗУ, реализованное на флэш-дисках? Функциональное назначение флэш-дисков ничем не отличается от всех остальных «дисков» в компьютерном смысле этого слова – устройство для долговременного энергонезависимого хранения информации с возможностью многократной перезаписи. Их самое главное отличие от тривиальных винчестеров и флоппи-дисков заключается в отсутствии каких-либо подвижных механических деталей конструкций. Для хранения информации во флэш-дисках используются специальные микросхемы памяти.

В отличие от любого винчестера, флэш-диск практически невосприимчив к ударам и вибрациям, он выдерживает удары до 1000g. Диапазон рабочих температур флэш-дисков может составлять от  $-40\text{ }^{\circ}\text{C}$  до  $+85\text{ }^{\circ}\text{C}$ , что является просто недостижимым для традиционных механических дисков. Они все-таки вертятся, а на морозе или жаре это не так-то просто. Кроме того, при изменении температуры механические части конструкции подвергаются сжатию и расширению. Причем для различных материалов в различной степени. Все это приводит к тому, что типичный рабочий диапазон температур для механических дисковых накопителей составляет  $+5\text{ }^{\circ}\text{C} \dots +40\text{ }^{\circ}\text{C}$ . Флэш-диски, в отличие от механических, не затрачивают времени на позиционирование считывающей головки над необходимым цилиндром или сектором вращающегося магнитного диска. С этой точки зрения доступ к любым данным на флэш-диске происходит практи-

чески мгновенно. В настоящее время за пальму лидерства в разработке и производстве флэш-дисков борются две фирмы: SanDisk (США), M-Systems (Израиль), которые разрабатывают микросхемы флэш-памяти емкостью до 3,5 Гбайт.

Выбор операционной системы (ОС) этого уровня зависит от жесткости требований реального времени. Так, для достаточно большого спектра задач можно использовать ОС OS-9, QNX, VxWorks и другие, а также расширение реального времени для Windows NT – RTX (Real Time Extension) (см. главу 2).

Прикладное программное обеспечение современных PLC, имеющих встроенную операционную систему, может быть разработано как с использованием традиционных инструментальных средств (компиляторы языков С, С++, Паскаль, Фортран и т.д.), так и на основе специализированных языковых средств. Традиционная технология требует от разработчика знаний не только в области использования языков программирования, но и особенностей операционной системы, а также аппаратных возможностей данного контроллера и организации системы ввода-вывода. При этом разработанное программное обеспечение будет привязано только к данному типу контроллера и не может быть перенесено на другую аппаратно-программную платформу.

Потребность в специальных платформи-независимых языках программирования возникла давно. Она послужила причиной объединения усилий ведущих производителей контроллеров по разработке под эгидой Международной Электротехнической Комиссии (IEC, МЭК) стандарта на такие языки программирования PLC. В результате появился стандарт МЭК 61131-3, специфицирующий синтаксис и мнемонику языков программирования, которые могут применяться для любого логического контроллера.

Стандартом МЭК 61131-3 определяется пять языков: три графических (SFC, FBD, LD) и два текстовых (ST, IL).

**SFC (Sequential Function Charts)** – *графический язык последовательных функциональных схем*. Язык SFC предназначен для использования на этапе проектирования программного обеспечения и позволяет описать "скелет" программы – логику ее работы на уровне последовательных шагов и условий переходов (транзакций). Это, по своей сути, аналог блок-схем программ.

**FBD (Function Block Diagram)** – *графический язык функциональных блок-овых диаграмм*. Язык FBD применяется для построения комплексных процедур, состоящих из различных функциональных библиотечных блоков – арифметических, регуляторов, тригонометрических и т.д.). Аналог – принципиальные схемы электронной аппаратуры.

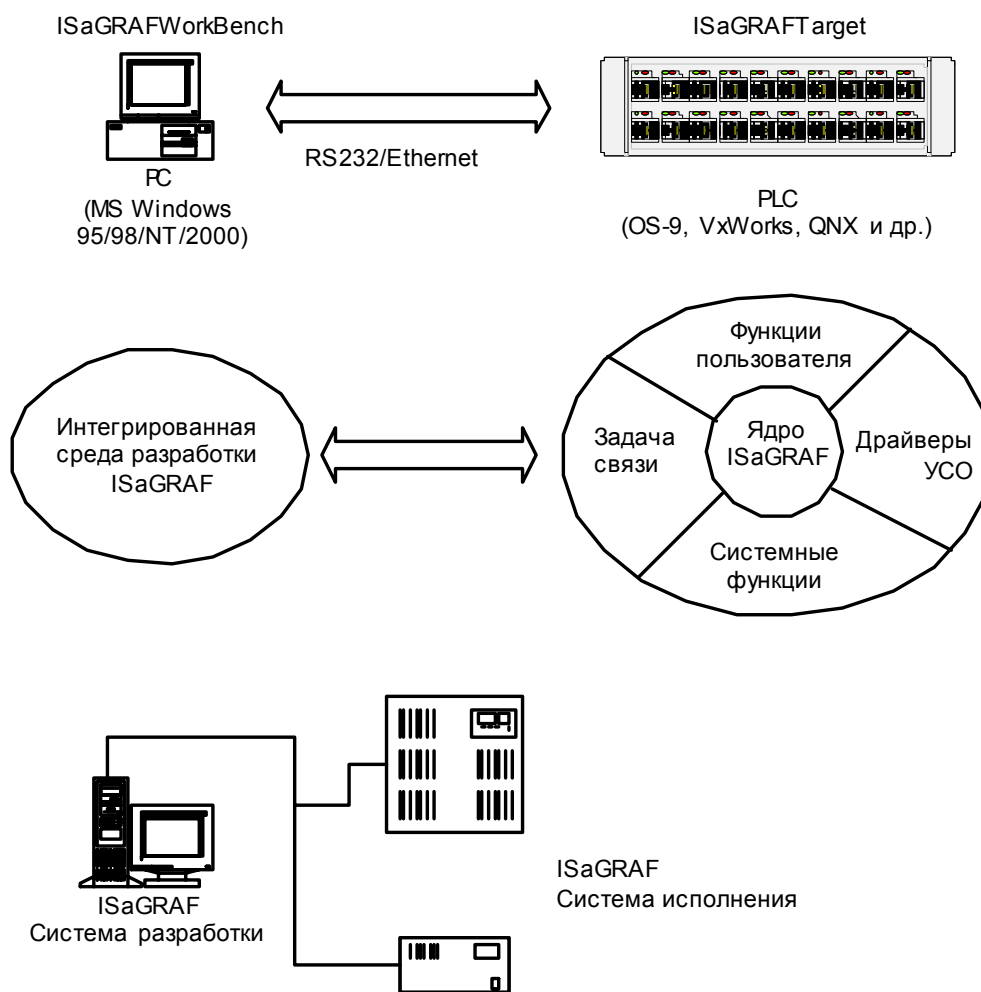
**LD (Ladder Diagrams)** – *графический язык релейной логики*. Язык LD применяется для описания логических выражений различного уровня сложности. Позволяет моделировать различные схемы релейной логики.

**ST (Structured Text)** – *язык структурированного текста*. Это язык высокого уровня, по мнемонике похож на Паскаль и применяется для разработки процедур со сложной логикой, обработки данных.

**IL (Instruction List)** – *язык инструкций*. Это язык низкого уровня класса ассемблера и применяется для программирования эффективных, оптимизированных процедур.

Одной из первых реализаций этого стандарта стала инструментальная система ISaGRAF разработки фирмы CJ International (Франция), позволяющая разрабатывать программное обеспечение для интеллектуальных PLC. ISaGRAF предоставляет возможность разрабатывать процедуры с использованием языка C/C++, причем эти процедуры можно вызывать из любого, описанного выше языка.

Система ISaGRAF состоит из двух частей: системы разработки ISaGRAF Workbench и системы исполнения ISaGRAF Target (**рис. 5.2**).



**Рис. 5.2.** Структура системы ISaGRAF

Система разработки (ISaGRAF Workbench) представляет собой набор Windows-приложений, интегрированных в единую инструментальную среду и работающих под управлением ОС Windows 95/98/NT/2000.

Основу системы исполнения (ISaGRAF Target) составляет набор программных модулей, выполняющих самостоятельные задачи под управлением ядра ISaGRAF. Ядро ISaGRAF реализует поддержку стандартных языков программирования, типового набора функций и функциональных блоков и драйверов ввода-вывода. Система исполнения ISaGRAF Target загружается в ПЗУ при изготовлении контроллера или загружается пользователем. Задача связи обеспечивает поддержку процедуры загрузки пользовательского ISaGRAF-приложения

со стороны программируемого контроллера, а также доступ к рабочим переменным этого приложения со стороны отладчика системы разработки ядра ISaGRAF.

Общий алгоритм работы с пакетом ISaGRAF можно представить следующим образом:

- создание проекта в рамках ISaGRAF Workbench;
- написание программ на языках ISaGRAF в соответствующих редакторах;
- описание переменных в редакторе переменных;
- описание соединений переменных с физическими и логическими устройствами ввода-вывода;
- компилирование проекта в исполняемый код для ISaGRAF Target;
- отладка программы в ISaGRAF Workbench;
- отладка программы в ISaGRAF Target.

### 5.1.2. Средний уровень

Основная задача этого уровня заключается в сборе информации с различных подсистем и/или контроллеров, их обработка и передача на верхний уровень. Средний уровень чаще всего представляет собой станции контроля и регистрации технологических параметров и предназначен для замены регистрирующих приборов. Современный подход к этому уровню – это SCADA-система. Кроме того, средний уровень является связующим звеном нижнего и верхнего уровней системы.

Аппаратная база этого уровня должна быть более мощной, предусматривающей возможность обмена через промышленные (Fieldbus) и офисные сети (например, Ethernet) с нижним и верхним уровнем. Предусмотрена организация логических и горизонтальных соединений с базами данных реального времени. *Обычно используют промышленные компьютеры PC совместимой архитектуры, а чаще всего – интеллектуальные контроллеры.*

Спектр задач интеллектуальных контроллеров и промышленных компьютеров более разнообразен и включает:

- сбор данных с контроллерного уровня;
- обработку данных, включая масштабирование;
- синхронизацию работы подсистем;
- организацию архивов по выбранным параметрам;
- резервирование каналов передачи данных.

*База данных реального времени.* До появления IndustrialSQL Server (Wonderware, США) задача регистрации информации в реальном времени могла быть решена либо на уровне программного обеспечения интеллектуального контроллера, либо на уровне SCADA-системы. Сейчас с появлением IndustrialSQL Server появилась дополнительная возможность обеспечить высокоскоростное хранение информации в БД, архивировать данные с целью экономии места на диске, обеспечивать стандартный доступ к данным. IndustrialSQL Server пред-

ставляет собой расширение Microsoft SQL Server и обеспечивает сбор данных в сотни раз быстрее, чем любые другие реляционные базы данных. Он может использоваться как в небольших цехах с сотней регистрируемых технологических параметров, так и на крупных промышленных предприятиях с сотнями тысяч параметров.

*Системы Micro-SCADA* решают задачи, аналогичные традиционным SCADA-системам. Отличие лишь в ориентации программного обеспечения на определенную отрасль (часть объекта, например систему загрузки или воздухо-нагреватель доменной печи и т.п.). Выбор ориентации определяет спектр драйверов или серверов ввода-вывода для подключаемого специфичного контроллера оборудования, набор графических объектов.

### 5.1.3. Верхний уровень

Основным средством коммуникации на этом уровне обычно является компьютерная сеть Ethernet с протоколами TCP/IP.

Компьютеры этого уровня решают следующие задачи:

- диспетчерский сбор, сигнализация управления и отображение всего технологического процесса или его отдельных подсистем (SCADA-системы), реализованные на операторских станциях;
- хранение технологической информации в базе данных;
- компьютерная поддержка принятия решений (комплекс математических моделей технологического процесса, интеллектуальная поддержка принятия решений), реализованная на расчетных станциях;
- интеграция локальной сети в корпоративную сеть всего предприятия (коммуникационный сервер).

Выбор базовых компонентов этого уровня, предназначенных для организации комфортного человеко-машинного интерфейса, во многом определяется сложившейся привычной ориентацией на персональные компьютеры и имеющийся подготовленный персонал для работы на них. В связи с этим, как правило, выбор сводится лишь к определенной требуемой конфигурации PC-совместимых компьютеров и операционных систем Windows 95/98/NT/2000.

В качестве программного обеспечения этого уровня можно использовать достаточно широкий спектр традиционных SCADA-систем (Genesis, InTouch, WinCC, MIKSSys и др.), СУБД (Access, SQL Server, SQL Base, Oracle Server, Informix, SyBase и др.), коммуникационных программ Internet Information Server (IIS) и др. (см. главы 2, 3).

Представленная на **рис. 5.1** обобщенная схема в ряде случаев является избыточной. Естественно, что возможна и различная компоновка информационных систем. Так, для некоторых, достаточно инерционных металлургических процессов (агломерационное, доменное производство) нет необходимости формирования отдельной базы данных реального времени, и для решения технологических задач можно ограничиться базой данных верхнего уровня (SQL Server). Может отсутствовать и необходимость в системе Micro-SCADA. Однако, чаще всего, для относительно сложных информационных систем в металлургии ми-

нимальный набор необходимых элементов таких систем включает комплекс датчиков, исполнительной и сигнализирующей аппаратуры, сеть PLC, интеллектуальные контроллеры или PC, операторские (SCADA-системы), расчетные станции и базы данных (SQL Server). В качестве примера на **рис. 5.3** представлен один из вариантов простой архитектуры системы контроля и управления, когда информация, поступающая от интеллектуальных контроллеров или PC, хранится в базе данных. При этом клиент-серверная организация взаимодействия расчетных и операторских (клиентских) станций обеспечивает возможность этим станциям с использованием языка структурированных запросов SQL взаимодействовать с сервером баз данных. Естественно, что такая архитектура информационно-управляющей системы обеспечивает меньшую скорость отражения измененной первичной технологической информации на мониторах операторских станций и расчетных станций.

Заметим в заключение, что в простейших случаях возможно совмещение нижнего и среднего уровней информационной системы и реализация их на промышленных компьютерах PC, имеющих собственные УСО (**рис. 5.4**). Номенклатура таких УСО для промышленных компьютеров отличается большим разнообразием. Такая конфигурация системы по принципу работы фактически ничем не отличается от централизованной системы и обычно она используется, когда отсутствует необходимость в применении принципа распределенности (децентрализации) информационно-управляющей системы технологического процесса.

Очевидно, что для многих задач такой подход не обеспечивает управление в реальном времени. Связано это с тем, что современное программное обеспечение операторского интерфейса в своей массе предназначено для работы под управлением операционной системы Windows, которая пока еще слабо оптимизирована для работы в режиме жесткого реального времени. Кроме того, операторская станция обычно отличается нестабильным программным окружением (оператор может, например, запустить зараженную вирусом программу), что в сочетании со сложностью самой операционной системы может нарушить функционирование приложения, отвечающего за управление технологическим процессом вплоть до полного "зависания" компьютера.

## 5.2. Общая характеристика промышленных сетей

Каналы связи контроллеров с интеллектуальными контроллерами и промышленными компьютерами отличаются большим разнообразием. Выбор конкретного решения зависит от архитектуры системы, расстояния, числа контролируемых точек, требований к пропускной способности и надежности канала, наличия допустимых линий связи. Задача промышленных сетей (Fieldbus) состоит в организации физической и логической связи различных компонентов автоматизации таким образом, чтобы информация с этого уровня была доступна общезаводской информационной системе.



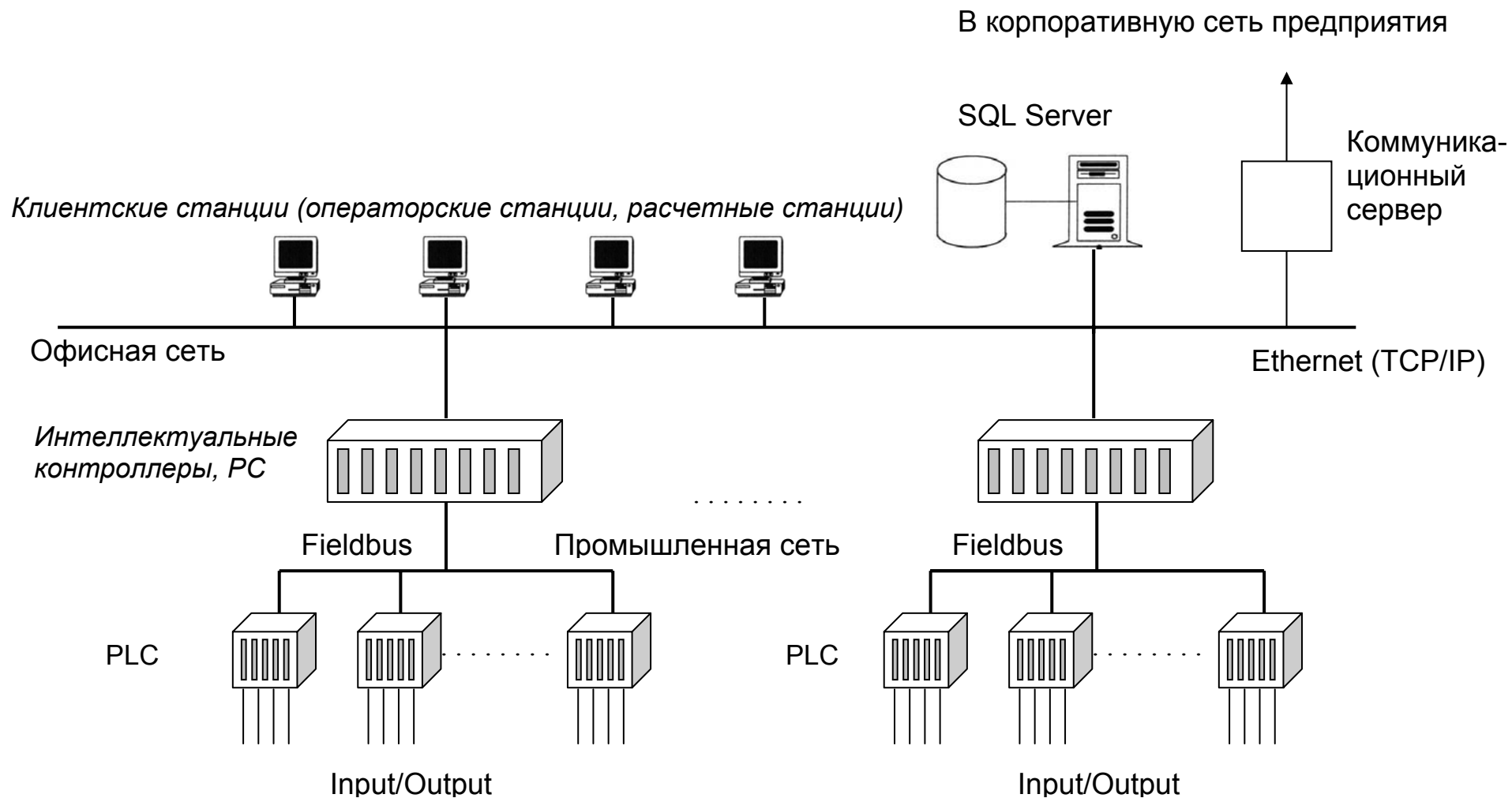
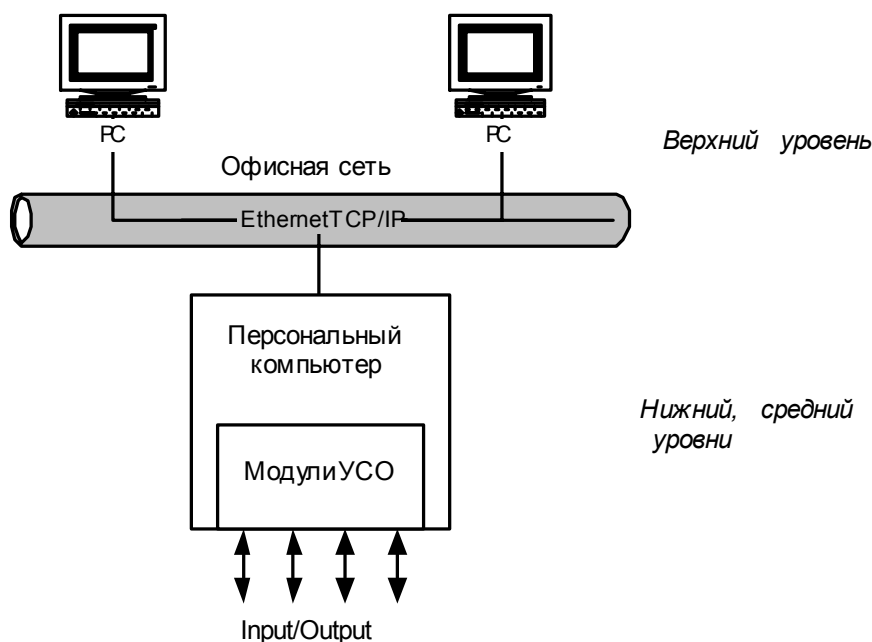


Рис. 5.3. К структуре информационной системы



**Рис. 5.4.** Простейший вариант комплектования компьютерной информационной системы

Таким образом, **промышленные сети *Fieldbus*** – это специализированные быстродействующие каналы связи, позволяющие эффективно решать задачи надежных и помехоустойчивых соединений с компонентами автоматизации различных разновидностей, в частности программируемых логических контроллеров, промышленных персональных компьютеров, удаленных модулей УСО и др.

Сформулируем лишь некоторые основные требования, которые можно предъявить к "идеальной" промышленной сети:

- 1) предсказуемость времени доставки информации;
- 2) помехоустойчивость;
- 3) доступность и простота организации физического канала передачи данных;
- 4) максимальный сервис для приложений верхнего уровня;
- 5) минимальная стоимость устройств аппаратной реализации, особенно на уровне контроллеров;
- 6) возможность получения «распределенного интеллекта», путем представления максимального доступа к каналу нескольким ведущим узлам.
- 7) управляемость и самовосстановление в случае возникновения нештатных ситуаций.

Следовательно, промышленные сети должны полностью удовлетворять запросам потребителей по модульности, надежности, защите от внешних помех, простоте в построении, монтаже и программировании.

Нетрудно заметить, что требования противоречивы, так второе требование противоречит третьему. В целом следует отметить, что промышленная сеть, по своей сути, – один большой компромисс. От того, как расставлены акценты в этом компромиссе, зависит успешность решения задач, стоящих перед данной сетевой архитектурой.

Сегодня говорить о некой универсальной промышленной сети еще рано. Однако известны требования к ней и понятны прикладные задачи, которые необходимо решать с ее помощью:

- *автоматизация на уровне периферийного оборудования.* Здесь необходимы следующие качества: высокая скорость передачи, короткое время реакции на события, длина линий связи обычно до 300 метров;
- *автоматизация на уровне управления технологическими процессами (уровень производства).*

Этот уровень требует следующих качеств: среднее время опроса датчиков до 100 мс, длина линий связи до 1500 м, с возможностью работы устройств во взрывоопасных средах.

Как правило, способность передавать большие объемы данных не столь важна в промышленных условиях, как оперативность и предсказуемость времени передачи информации. Скорости передачи по промышленным шинам обычно колеблются от 9,6 кбит/с до 12 Мбит/с при «медных» средах передачи и до 500 Мбит/с на оптоволокне. Однако в промышленных системах в последнее время стали использовать высокоскоростные офисные сети Ethernet.

На каком бы уровне сеть не работала, она должна быть *стандартной*. Соблюдение всех семи уровней эталонной модели OSI/ISO в системах управления обычно не требуется. Кроме того, наличие дополнительных изолирующих межуровневых интерфейсов привело бы к потере производительности системы в режиме реального времени и сделало бы существенно менее предсказуемыми задержки прохождения сообщений в сети. Большинство промышленных сетей имеют сжатую трехуровневую архитектуру и поддерживают 1-, 2- и 7-й уровни OSI модели (см. главу 4). Сервисные функции остальных уровней либо отсутствуют, либо включены в прикладной уровень. Напомним, что

- *физический уровень – уровень 1 (Physical Layer)* обеспечивает необходимые механические, функциональные и электрические характеристики для установления, поддержания и размыкания физического соединения;
- *уровень передачи данных – уровень 2 (Data Link Layer)* гарантирует передачу данных между устройствами. Этот уровень управляет не только сетевым доступом, но также механизмами защиты и восстановления данных в случае ошибок при передаче;
- *прикладной уровень – уровень 7 (Application Layer Interface)* обеспечивает непосредственную поддержку прикладных процессов и программ конечного пользователя и управление взаимодействием этих программ с различными объектами сети передачи данных.

Основными достоинствами промышленных сетей являются недорогие линии и надежность передачи данных. Данные передаются последовательно бит за битом, как правило, по одному физическому каналу (одному проводнику). Такой режим передачи не только экономит кабельное оборудование, но и позволяет решать задачи по надежной передаче данных на большие расстояния.

В настоящее время используются ряд физических интерфейсов соединения, определяющих электрические и механические характеристики интерфейса в промышленных сетях, а также номенклатуру цепей стыка и процедуру их взаи-

модействия на аппаратном уровне: токовая петля (ИРПС), RS-232C, RS-485, RS-422. Заметим, что сокращения RS означают Recommended Standard (рекомендательный стандарт). Однако если производитель желает, чтобы его устройства были совместимы с другими аналогичными устройствами, он должен придерживаться определенных правил, которые и закреплены этими стандартами. Основными из них являются RS-232C, RS-485. Важное отличие RS-232C от RS-485 заключается в том, что RS-232C использует небалансный (unbalanced) сигнал, в то время как RS-485 использует балансный (balanced) сигнал.

Небалансный сигнал передается по несбалансированной линии, представляющей собой сигнальную землю и одиночный сигнальный провод, уровень напряжения на котором используется, чтобы передать или получить двоичные 1 или 0. Напротив, балансный сигнал передается по сбалансированной линии, которая представлена сигнальной землей и парой проводов, разница напряжений между которыми используется для передачи/приема бинарной информации (а все вместе составляют экранированную витую пару). Не вдаваясь в подробности, можно сказать, что сбалансированный сигнал передается быстрее и дальше, чем несбалансированный.

Сравнительные характеристики RS-232C, RS-485 приведены в **табл. 5.1**.

Таблица 5.1

### Сравнительные характеристики стандартных физических интерфейсов

Сравнительные характеристики	RS-232C	RS-485
Максимальное число приемников /передатчиков на линии	1/1	32/32
Максимальная длина линий (без повторителей)	15 м	1220 м
Максимальная скорость передачи	38,4 кбод	90 – 500 кбод

**Интерфейс RS-232C.** Это широко используемый стандартный последовательный интерфейс обеспечивает работу стандартного оборудования передачи данных между компьютерами и терминалами. Тип соединения – «точка-точка». Электрическая система основана на импульсах  $\pm 12$  В, кодирующих последовательности "0" и "1". Механически этот стандарт определяет 8-, 9- и 25-контактные разъемы. Основные сигналы передаются по линиям "передача/прием" данных. Остальные линии передают информацию о состоянии коммутируемых устройств.

**Интерфейс RS-485.** Это последовательный тип интерфейса, он используется для высокоскоростной передачи данных. Максимальная длина варьируется от 1,2 км на скорости до 90 кбод и до 200м – на скорости до 500 кбод. Стандарт на RS-485 предусматривает только 32 пары передатчиков/приемников, но производители расширили возможности RS-485, так что теперь он поддерживает от 128 до 255 устройств на одной линии, а используя репитеры, можно продлевать максимальную длину линии. Также стандарт на RS-485 предусматривает использование двухжильной экранированной витой пары, но возможно использование и четырехпроводной витой пары.

В промышленных сетях используются те же методы доступа, что и в локальных сетях – модель CSMA/CD (метод множественного доступа с контролем несущей и обнаружением конфликтов) и The Token Passing Method (метод с передачей маркера), описанные ранее.

Кроме этих методов используется метод MASTER-SLAVE – ведущий (запрос) – ведомый (ответ). В этом методе право инициировать циклы чтения-записи имеет только MASTER – ведущий узел. Он адресует каждого пассивного участника как SLAVE – ведомый, обеспечивает их данными и запрашивает у них данные. С тем чтобы увеличить пропускную способность шины, команды протокола должны быть как можно проще.

Сегодня существуют десятки разнообразных промышленных сетей. Наиболее популярными являются Industrial Ethernet, Profibus, Controller Area Network bus (CANbus), Factory Instrumentation Protocol (FIP), Interbus-S, Bitbus, MODbus, Foundation Fieldbus (FF), LONWorks и др. Каждая из перечисленных систем имеет свои особенности и области применения. Отметим, что практически все эти протоколы имеют поддержку в виде сильных компаний и практически все опираются на те или иные стандарты.

Большие усилия сегодня направлены на разработку аппаратно-программных шлюзов (мостов) из одного протокола в другой. Заметим, что в настоящее время ведется разработка общеевропейского стандарта, объединяющего Profibus и FIP.

*В зависимости от области применения весь спектр промышленных сетей можно разделить на два уровня:*

- *sensor/actuator level* (датчиковый уровень) – задачи сетей этого уровня сводятся к опросу датчиков и управлению работой разнообразных исполнительных механизмов. Здесь основным параметром является время и частота опроса каналов датчиков. Типичные представители этого уровня Profibus-DP/PA, Interbus-S;
- *field level* (системный уровень) – промышленные сети этого уровня решают задачи по управлению процессом производства, сбором и обработкой данных на уровне промышленных контроллеров. На этом уровне на первый план выдвигается задача передачи больших объемов данных, при этом схема потоков данных может быть достаточно сложной. Типичные представители этого уровня Profibus-FMS, Bitbus.

Другими словами, необходимо различать промышленные сети для системного уровня (Field busses) и датчикового уровня (Sensor/actuator busses).

Сравнение этих двух классов в самом общем виде можно получить по критериям, представленным в **табл. 5.2**.

Заметим, что существуют сети для обеих уровней применения, типичными представителями которых являются CANbus, FIP и др.

На **рис. 5.5** представлена обобщенная сетевая структура, показывающая в общем виде возможное использование того или иного вида сетей на отдельных уровнях условного промышленного предприятия.

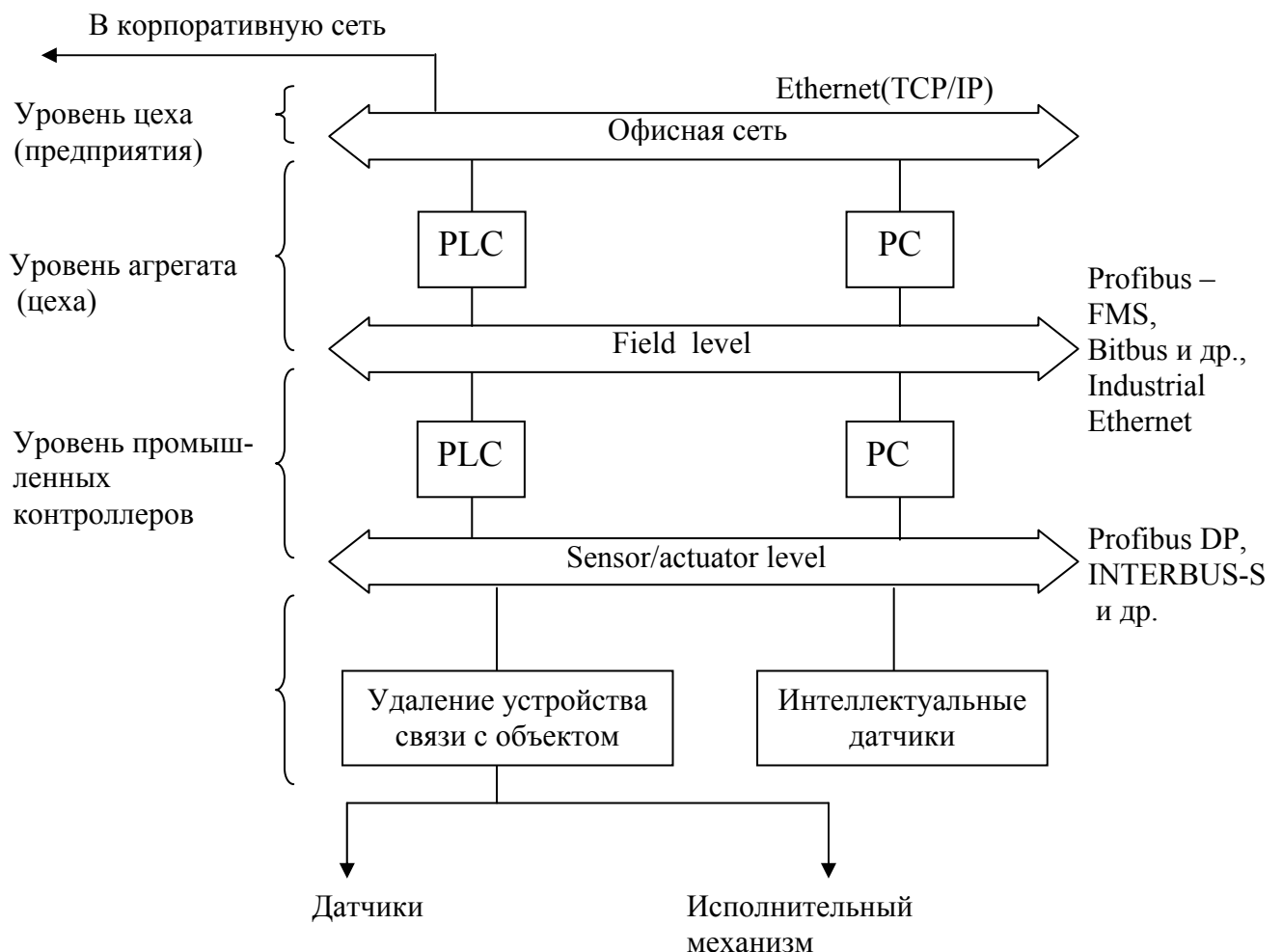
В распределенных промышленных системах между собой связаны сетевые узлы самых разных типов, с самыми разными скоростями и расстояниями передачи данных. Поэтому решить все задачи при помощи промышленной шины

одного типа невозможно. Однако все вместе они удовлетворяют требованиям практически любой системы управления, имеющей распределенную архитектуру. Характеристики некоторых промышленных сетей приведены в табл. 5.3. Рассмотрим их подробнее.

Таблица 5.2

**Сравнительные критерии промышленных сетей  
типов Fieldbus и Sensorbus**

Основные критерии	Fieldbus	Sensorbus
Расширение сети	От 100 м до 1500 м	До 100 м
Время цикла	От 10 мс до 10 с	От 1 мс до 1 с
Объем передаваемых данных за цикл	От 8 байт до сотен байт	От 1 до 8 байт
Цена среды передачи	Низкая	Очень низкая



**Рис. 5.5.** Уровни связи и типовые компьютерные сети в информационных системах технологических процессов

**Profibus (Process Field Bus).** Из всего многообразия промышленных шин следует выделить Profibus как наиболее популярный в настоящее время в Европе и США. В мире работают свыше 100 тыс. систем на основе этой технологии, а число работающих узлов достигает 2 млн. Этот стандарт определяет обмен информацией между компонентами автоматизации и многими разновидностями PC, PLC и рассматривается как кандидат на обретение международного IEC-стандарта.

Таблица 5.3

### Характеристики некоторых промышленных сетей

Протокол	Страна	Среда передачи	Число узлов	Скорость передачи данных, кбод	Длина линий	Топология сети	Принцип доступа к сети
PROFIBUS	Германия	Витая пара, оптокабель	До 126	500–1500 (FMS); 1500–12000 (DP); 31 (PA)	1200 м; 4800 м с повторителем; до 23 км оптокабелем	Звезда; шина	MASTER/ SLAVE/ TOKEN
CANbus	Германия	Витая пара	До 30	50–1000	До 1 км – 2 кбод, До 40 м – 1 Мбод	Шина	Произвольный доступ по приоритету CSMA/CR
BITbus	США	Витая пара, радиоканал, оптокабель	До 250	500–1400	До 1200 м	Шина	MASTER/ SLAVE
MODBUS	США	Не специфицированы	1 MASTER 247 SLAVE	Нет данных	15 м RS-232C, 1200 м RS-485	Звезда	MASTER/ SLAVE/

Под общим названием Profibus понимается совокупность трех различных вариантов этой сети: Profibus-FMS, Profibus-DP, Profibus-PA, каждый из которых ориентирован на свою прикладную область. Архитектура составляющих частей протокола Profibus приведена на **рис. 5.6**.

Все три варианта протокола используют общий канальный уровень (протокол доступа к передающей среде).

**Profibus FMS (FMS — Fieldbus Message Specification, Спецификация сообщений шины полевого уровня).** Это универсальное решение для задач взаимодействия на верхнем уровне иерархии промышленных связей и предназначено для взаимодействия между контроллерами. На этом уровне требуется высокая степень функциональности, и этот критерий более важен, чем критерий

скорости. Основное назначение – передача больших объемов данных. Следовательно, если требуется объединить в детерминированную сеть несколько контроллеров, оптимальным вариантом будет PROFIBUS-FMS. Протокол FMS по стандарту определяет уровни 1, 2 и 7, может работать со скоростью до 1,5 Мбод и предоставляет пользователю широкий выбор коммуникационных функций.

PROFIBUS			
Profibus-FMS		Profibus-DP	PROFIBUS-PA
		Функции пользовательского уровня	
7	Fieldbus Message Specification (FMS)	Нет	Нет
3...6	Отсутствуют		
2	Fieldbus Data Link (FDL) –промышленный канал данных		
1	RS-485/ Fibre Optic	IEC1158-2	

Рис. 5.6. Архитектура протокола PROFIBUS

**Profibus-DP (DP – Decentral Periphery, Децентрализованные периферийные устройства).** Это оптимизированная по производительности версия Profibus, предназначенная специально для критичных по времени взаимодействий между системами автоматизации и распределенным периферийным оборудованием, где на первое место выходит такой параметр, как продолжительность цикла шины. Profibus-DP был разработан с целью упрощения подключения простых и недорогих устройств ввода-вывода в системах с повышенными временными требованиями. Profibus-DP использует уровни 1 и 2 и реализует набор функций, обеспечивающих интерфейс к прикладной задаче пользователя. Отсутствие уровня 7 обеспечивает быструю и эффективную технологию передачи за счет прямого доступа к функциям канального уровня. Обычно такие системы состоят из одного главного устройства, периодически обращающегося к подчиненным устройствам ввода-вывода. Так, для передачи 512 бит данных, распределенных по 32 станциям, требуется всего 6 мс. Таким образом, Profibus-DP используется для организации скоростного обмена небольшими порциями данных с распределенными входами и выходами системы.

**Profibus-PA (PA – Process Automation, Автоматизация процессов).** Проект Profibus-PA применяется в системах автоматизации взрывоопасных производств, где для обеспечения требований безопасности необходима низковольтная аппаратура. Фактически это аналог Profibus-DP, но в иной физической реализации. При этом на физическом уровне (уровень 1 OSI-модели) технология передачи реализована не с помощью RS-485, а согласно стандарту IEC61158-2



для организации технологии передачи во взрывоопасных средах с внутренней защитой данных. По сигнальным проводам, в соответствии со стандартом, по витой паре идет не только информационный обмен с датчиками и исполнительными механизмами, но и происходит подача питания к ним. Причем энергия в кабеле недостаточна, чтобы образовать искру, что соответствует искробезопасной цепи.

Передающая среда в Profibus, как правило, строится на основе RS-485 интерфейса и "медных" средах передачи. Существует возможность использования и оптического кабеля в качестве физической среды передачи информации. Эта технология находит применение в случае присутствия сильного электромагнитного влияния и при необходимости в высокой скорости передачи информации на большие расстояния по сравнению с RS-485. Переход с медного кабеля на оптоволоконный производится с помощью преобразователей RS-485/Fiber Optic. Для увеличения длины сети линейной конфигурации можно ставить до четырех повторителей. В этом случае общая длина сети может достигать шести километров. Для формирования различных структур сети (древовидной, звездообразной и т.п.) используются активные повторители – разветвители сети.

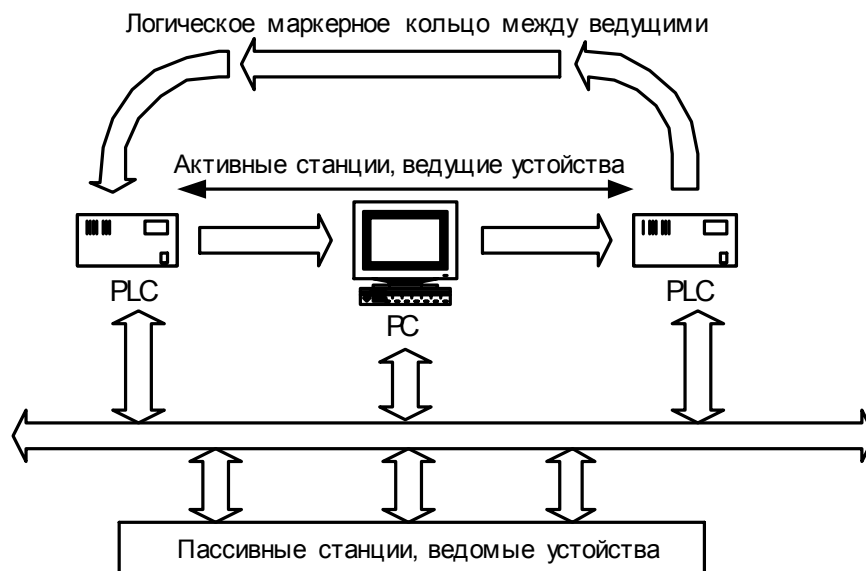
Для взаимодействия между ведущими – контроллерами (компьютерами) – протокол включает метод передачи маркера, а для взаимодействия между ведомыми – периферийными устройствами – метод MASTER-SLAVE (ведущий-ведомый). Таким образом, в Profibus применяется комбинированный метод, называемый гибридным доступом к передающей среде.

Метод передачи маркера обеспечивает присвоение права доступа к шине в пределах точно определенного временного интервала и используется в Profibus только для нескольких ведущих устройств, при этом маркер циркулирует между всеми ведущими устройствами в пределах максимального времени оборота маркера.

Метод ведущий-ведомый позволяет ведущему устройству (активной станции), которому в настоящий момент принадлежит право передачи информации, взаимодействовать с ведомыми устройствами (пассивными станциями). Каждое ведущее устройство имеет возможность получать данные от ведомых устройств и передавать данные ведомым устройствам.

Ведомые устройства – это обычные периферийные устройства. Типичными ведомыми устройствами являются датчики, исполнительные устройства и передаточные механизмы. У них нет прав доступа к шине – то есть они могут только подтверждать принимаемые сообщения или передавать сообщения ведущему устройству по его запросу. Ведомые устройства называются также "пассивными станциями".

На **рис. 5.7** показана структура Profibus с тремя активными станциями (ведущими устройствами) и с несколькими пассивными станциями (ведомыми устройствами). Три ведущих устройства образуют логическое маркерное кольцо. После того как активная станция получает маркер, ей на определенное время дается разрешение выполнять на шине функции ведущего устройства. Она может взаимодействовать со всеми ведомыми станциями по методу MASTER-SLAVE и со всеми ведущими станциями, участвующими во взаимоотношениях по методу передачи маркера.



**Рис. 5.7.** Гибридный метод управления доступом

**CANbus (Controller Area Network bus).** Возрастающим признанием пользуется интерфейс CANbus, разработанный фирмой Bosch и Intel. Он представляет собой последовательный интерфейс, специально созданный для соединения между собой датчиков, исполнительных устройств и интеллектуальных контроллеров. Преимущества интерфейса CAN – обеспечение режима обмена в реальном масштабе времени, благодаря возможности инициативной передачи данных при изменении состояния входных сигналов, высокая помехоустойчивость. Особая привлекательность интерфейса CAN заключается в его устойчивости к электрическим и электромагнитным помехам, характерным для цеховых условий эксплуатации.

CANbus – это последовательная шина, механизм работы которой описывается моделью децентрализованного контроля за доступом к шине. Эта модель представляет собой модернизированный вариант модели CSMA/CD. Отличие заключается в механизме разрешения коллизий.

Протокол CANbus относится к классу CSMA/CR (Carrier Sense Multiple Access with Collision Resolution, метод коллективного доступа с опознаванием несущей и разрешением коллизий). Разрешение коллизий производится аппаратурой по принципу побитового сравнения сетевых адресов конфликтующих устройств (**рис. 5.8**). Станция, пытающаяся передать очередную "единицу" из своего адреса, видя, что реально в канале передается "ноль", понимает, что кон-



Скорость передачи изменяется от 500 до 2400 кбод. Синхронный режим передачи предполагает использование двух дифференциальных сигнальных пар: одной для данных, другой для синхронизации.

2. *Режим с самосинхронизацией.* Использование этого режима позволяет значительно удлинить шину. Стандартом определяется две скорости передачи: 375 кбод (до 300 м) и 62,5 кбод (до 1200 м). Используя репитеры, можно объединить последовательно несколько шинных сегментов (до 28 узлов на сегменте). Тогда общее число узлов можно довести до 250, длину общей шины – до нескольких километров. При этом режиме используются две дифференциальные пары: одна для данных и одна для управления репитером.

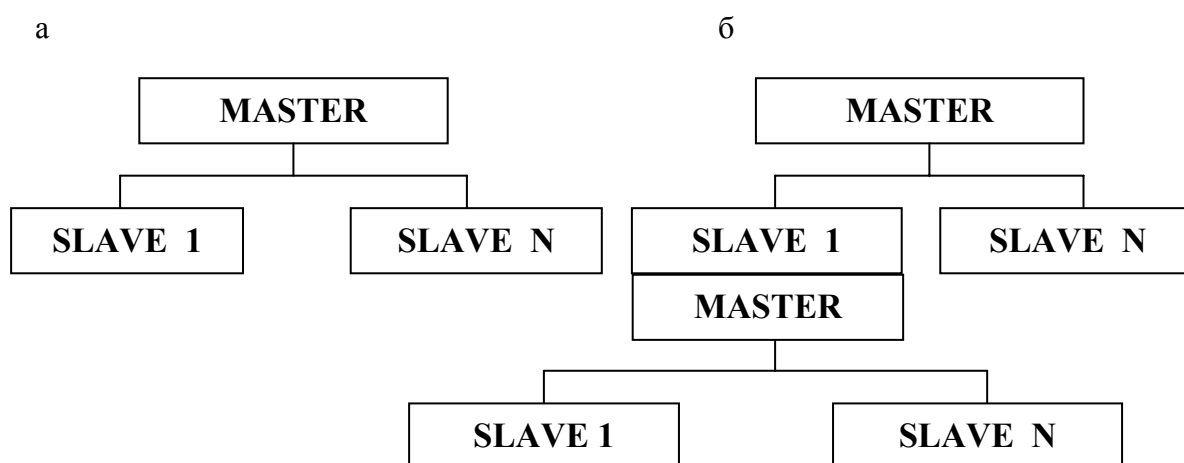


Рис. 5.9. Примеры топологии сети BITBUS:

а – одноуровневая; б – многоуровневая

**MODbus.** Этот протокол разработан фирмой Gould Inc. для построения промышленных распределенных систем управления. Специальный физический интерфейс для него не определен. Эта возможность предоставлена самому пользователю, можно использовать, в частности, RS-232C, RS-485.

Протокол MODbus работает по принципу MASTER/SLAVE, или "ведущий"–"ведомый". Конфигурация на основе этого протокола предполагает наличие одного MASTER-узла и до 247 SLAVE-узлов. При этом только MASTER инициирует циклы обмена данными. Существует два типа запросов:

- запрос-ответ (адресуется только один из SLAVE-узлов);
- широковещательная передача (MASTER через выставленные адреса 0 обращается ко всем остальным узлам сети одновременно).

Каждый запрос со стороны ведущего узла включает код команды (чтение, запись и т.д.), адрес абонента (адрес 0 используется, как уже отмечалось, для широковещательной передачи), размер поля данных, собственно данные или буфер под данные и контрольный CRC-код. Следует, однако, отметить, что MODbus значительно проще, но работает значительно медленнее, чем CANbus и Profibus. Схема функционирования протокола MODbus представлена на рис.5.10.

Использование промышленных сетей позволяет реализовать самые различные структурные схемы информационных систем. Так, в самом простом случае, если количество сигналов ввода-вывода и время решения задач (цикла) контроллера позволяют использовать один PLC, то выгоднее по протоколу, например CANbus, через интерфейс, например RS-232C, подключить PLC к PC. При этом процессорный модуль можно брать и без промышленной шины. Для мощ-



Рис. 5.10. Общая схема работы сети MODBUS

ных распределенных информационных систем приходится разбивать, например, сеть Profibus на несколько сегментов по горизонтали и организовывать многоуровневую промышленную сеть. В итоге получается мощная система управления технологическим процессом с большими возможностями по скорости и количеству сигналов ввода-вывода, позволяющая производить резервирование любой степени сложности на любом уровне.

### 5.3. Примеры комплектования нижних уровней информационной системы

Системы АСУ ТП любой сложности можно построить на основе аппаратуры как отечественных, так и зарубежных фирм, в частности MicroPC Octagon Systems, SIMATIC Win, Allen-Bradley, Omron, МИК CM9107, МФК и многих других.

Проиллюстрируем идеологию построения таких систем на примере некоторых аппаратно-программных средств зарубежных и отечественных фирм.

#### 5.3.1. Программируемые логические контроллеры серии Smart фирмы «PEP Modular Computers»

Основным требованиям построения современных информационных систем технологических процессов удовлетворяют, например, аппаратно-программные средства фирмы "PEP Modular Computers" (PEP MC Computer GmbH, Германия).

Контроллеры SMART I/O и SMART2 представляют собой открытые системы для промышленной автоматизации и обработки данных. Программируемые контроллеры могут быть использованы как программируемый микроконтроллер и как компьютерная система, работающая в реальном времени.

Контроллеры SMART I/O и SMART2 являются двумя параллельными линиями PLC, взаимодополняющими друг друга, использующими общую элементную базу как на уровне микросхем, так и на уровне модулей ввода-вывода.

Основными свойствами контроллеров SMART I/O и SMART2 являются:

- возможность потребителям самим выбрать необходимую комплектацию контроллера, как в аппаратном, так и в программном плане, что позволяет создать оптимальную для конкретной задачи систему с минимальными затратами;
- унификация модулей ввода-вывода позволяет снизить их стоимость;
- совместимость программного обеспечения с контроллерами среднего и верхнего уровня (IUC, VME) позволяет переносить программное обеспечение, в случае необходимости, на более мощные контроллеры и создавать многоуровневые системы без переучивания персонала предприятия – системного интегратора;
- достаточно мощный, 32-разрядный процессор (в то же время достаточно дешевый из-за массового применения в телекоммуникациях), и достаточная память трех видов позволяют решать относительно сложные задачи;
- устойчивая к сбоям операционная система реального времени OS-9 и пакет ISaGRAF позволяют легко создавать потребителям мощное, устойчивое прикладное программное обеспечение систем автоматизации;
- флэш-память (Flash-ROM) – до 4 Мбайт, ОЗУ – до 1 Мбайт;
- мощные сетевые возможности (MODbus, PROFIBUS-FMS, PROFIBUS-DP, CAN) позволяют создать как любой степени сложности и разветвленности децентрализованную систему автоматизации, так и простую одноконтроллерную систему, имеющую возможность, в случае необходимости, дальнейшего расширения;
- широкая номенклатура модулей ввода-вывода позволяет подключать практически любые датчики и исполнительные механизмы.

Контроллеры SMART I/O и SMART2 спроектированы на базе процессора MC68302 фирмы Motorola, микросхема которого содержит два микропроцессора. Один из них является стандартным процессором 68H000 с частотой 20 МГц, а второй – коммуникационным процессором с сокращенным набором команд (RISC). RISC-процессор используется для реализации протоколов промышленных шин и связи с модулями ввода-вывода. Тем самым основной процессор освобождается для решения задач логического управления и измерений. Контроллеры SMART I/O и SMART2 предназначены для использования в промышленной автоматике везде, где необходим компьютер и/или программируемый логический контроллер для работы в реальном времени.

Дадим краткую характеристику основным модулям SMART.

Основой контроллеров SMART I/O является модуль SMART-BASE, в состав которого входят преобразователь постоянного тока, центральный процес-

сор, таймер, полевая шина PROFIBUS, схема управления модулями ввода-вывода и гнезда для трех SMART-модулей ввода-вывода.

Модуль-носитель SMART-EXT предназначен для расширения ввода-вывода контроллеров типа SMART. К базовому модулю контроллера можно подключить каскадом (цепочкой) до четырех таких модулей. При этом в каждый модуль-носитель могут быть вставлены по два модуля ввода-вывода.

В контроллере SMART2 принцип модульности расширен, так как ядро контроллера также собирается из модулей. В качестве носителя используется SMART2-BASE, как для ядра, так и для модулей расширения. Предусмотрен интерфейс с офисной сетью Ethernet с поддержкой протокола TCP/IP.

Все модули ввода-вывода контроллеров SMART выполнены в едином конструктиве и все гальванически развязаны от системы.

Приведем характеристики некоторых модулей ввода-вывода.

*Модуль SM-DINI* предназначен для ввода восьми дискретных сигналов 24 В постоянного напряжения. Ток входа 3–4 мА при +24 В.

*Модуль SM-DOUT1* предназначен для вывода 8 дискретных сигналов. Каждая выходная линия может управлять нагрузкой до 500 мА, в том числе и индуктивной.

*Модуль SM – ADC1* предназначен для ввода 6 аналоговых сигналов тока или напряжения. Имеет следующие варианты исполнения:

- входные сигналы  $\pm 10$  В;
- входные сигналы 0 – 20 мА;
- входные сигналы  $\pm 5$  мА;
- входные сигналы 0–5 мА.

*Модуль SM-PT100* предназначен для подключения термометров сопротивления типа ТСП-100П (PT100). Имеет 16-разрядный АЦП и позволяет проводить измерения с абсолютной точностью до 0,2 °С.

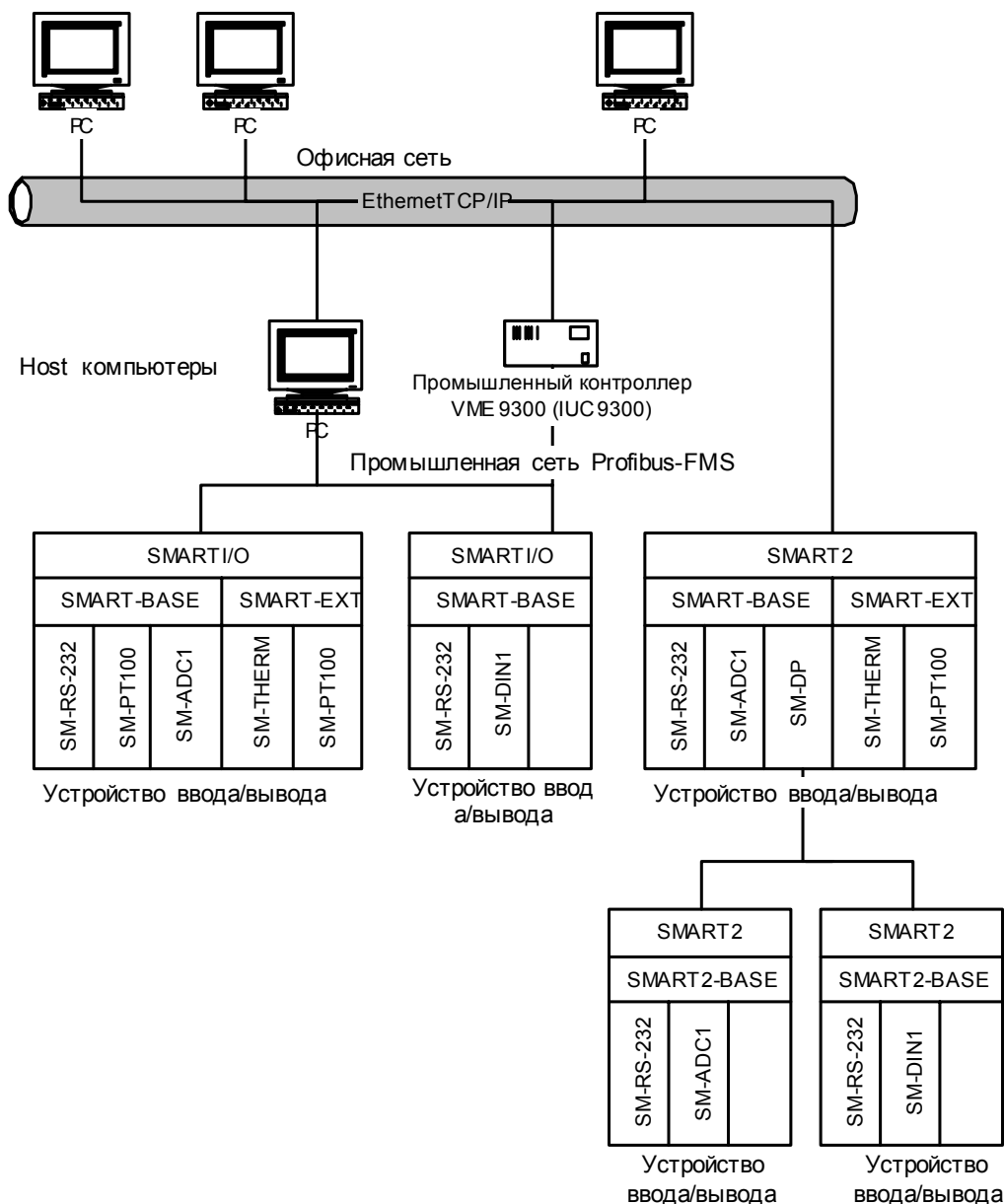
*Модуль SM-THERM* позволяет подсоединить до четырех датчиков – термопар. Термокомпенсация холодного спая проводится на этом же модуле, внешним подсоединением термосопротивления ТСП-100П (PT100). *Модуль SM-DP* предназначен для подключения контроллера SMART2 к сети PROFIBUS-DP в качестве подчиненного устройства (Slave).

*Модуль SM-RS232*, установленный в первое место SMART-BASE, позволяет получить пользователю дополнительный интерфейс для обмена информацией. В частности, можно реализовать протокол MODBUS.

Пример комплектования информационной системы на базе аппаратных средств этой фирмы представлен на **рис. 5.11**.

На нижнем уровне находятся программируемые логические контроллеры серии SMART со средствами связи с объектом, которые осуществляют сбор и предварительную обработку информации, автоматическое управление технологическими параметрами агрегатов, а также вывод управляющих сигналов на объект. Широкая номенклатура модулей УСО позволяет производить сбор дискретных, импульсных и аналоговых датчиков, а также выдавать команды управления на различные исполнительные устройства. Благодаря гибкости и модульности конструкции контроллеров серии Smart, можно конфигурировать как простые системы (на единицы и десятки каналов ввода-вывода), так и большие

системы (сотни каналов ввода-вывода). Микропроцессор контроллера поддерживает различные последовательные каналы со стыками RS-232C и RS-485 промышленной сети. Прикладное программное обеспечение контроллеров разрабатывается с использованием инструментального пакета ISaGRAF. Программируемые контроллеры оснащены ОС реального времени OS9. На уровне объектов все контроллеры объединены промышленной сетью Profibus-FMS.



*Рис. 5.11. Пример реализации информационной системы на базе контроллеров SMART*

На следующем уровне находятся мощные контроллеры верхнего уровня IUC – средней производительности или VME – высокой производительности фирмы «PEP Modular Computers» – VME9300, IUC9300 или PC, оснащенные ОС OS9. В них производится последовательная обработка и обмен информацией между верхним и нижним уровнями.



На верхнем уровне через локальную компьютерную сеть Ethernet реализуется дальнейшая обработка информации. В частности, осуществляется отображение информации (SCADA), доступ к базам данных, знаний, компьютерным системам поддержки принятия решений и т.п.

Заметим, что использование мощных контроллеров верхнего уровня предпочтительнее, чем PC. Основной недостаток офисных PC – недостаточная надежность. Кроме того, этот персональный компьютер будет перегружен задачами SCADA, Profibus-FMS, Ethernet, что приводит к ограничению решаемых задач по количеству и скорости.

Высокая надежность мощных контроллеров, с большими коммуникационными возможностями, позволяет подготавливать и оперативно хранить информацию для передачи на верхний уровень.

Современные информационно-управляющие системы, созданные на контролерах серии SMART и прикладном программном обеспечении ISaGRAF, успешно работают на ряде металлургических предприятий, в частности в системах АСУ ТП глиноземного производства на Богословском и Уральском алюминиевых заводах.

### **5.3.2. Модули серии ADAM 4000 фирмы «Advantech»**

Все большее распространение на предприятиях России при использовании современной идеологии автоматизации различных технологических процессов получают модули ADAM фирмы «Advantech».

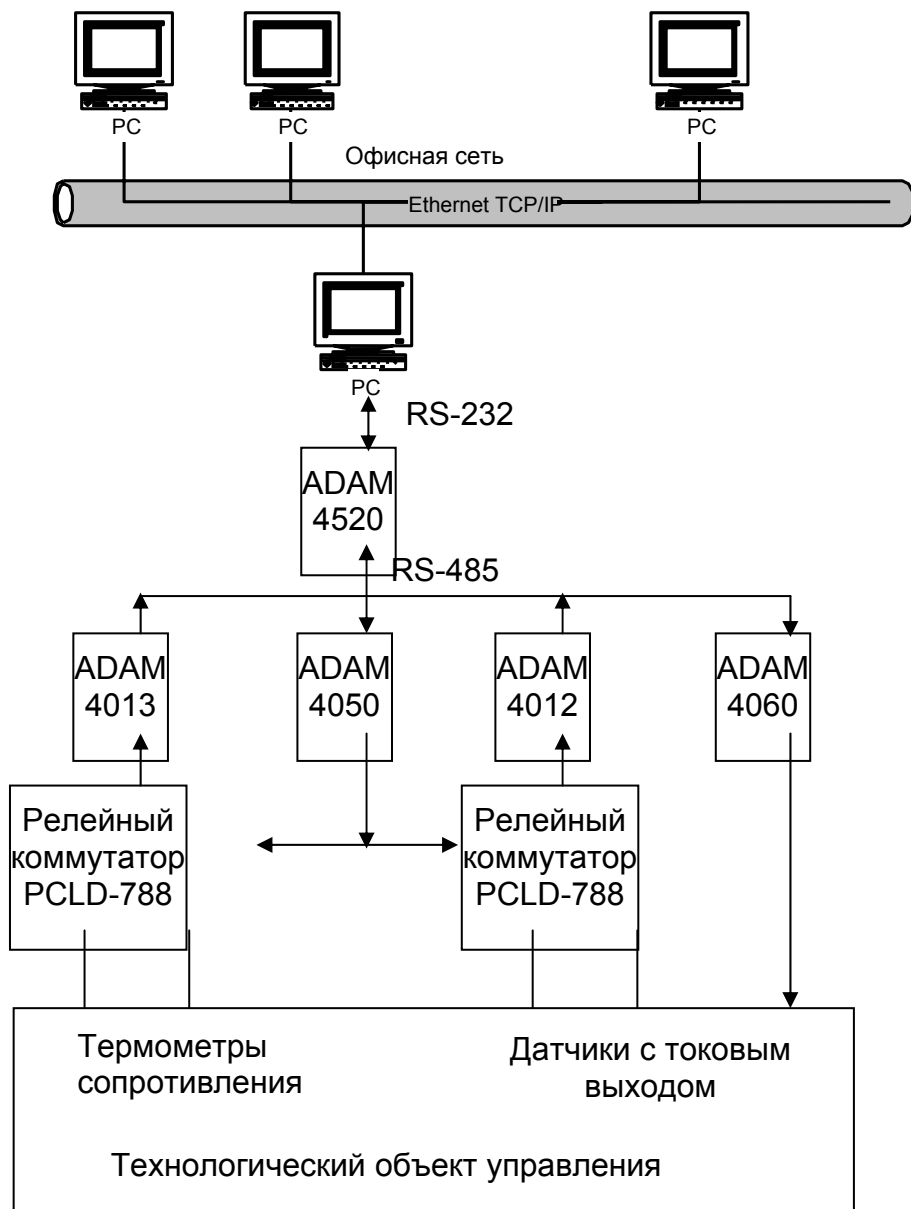
Модули серии *ADAM 4000* фирмы «Advantech» – модули для распределенных систем сбора данных и управления на базе интерфейса RS-485 на витой паре. Эти модули являются интеллектуальными модулями УСО на относительно небольшое число каналов ввода-вывода, предназначены для построения распределенных систем сбора данных и управления и представляют собой компактные и интеллектуальные устройства обработки сигналов датчиков, специально разработанные для применения в промышленности. Наличие встроенных микропроцессоров позволяет им осуществлять нормализацию сигналов, операции аналогового и дискретного ввода-вывода и передачу данных по интерфейсу RS-485.

Набор модулей этой серии позволяет подключать термометры сопротивления (ADAM 4013), термопары (ADAM 4011, ADAM 4018), датчики с аналоговыми сигналами (мВ, мА, В – одноканальный ADAM 4012, 8-канальный ADAM 4017) и др.

Пример комплектования информационной подсистемы на базе модулей серии ADAM 4000 представлен на **рис. 5.12**.

В типовых схемах каждый контролируемый параметр на стадии его определения (измерения или вычисления) подвергается стандартной математической обработке, которая включает:

- масштабирование измеряемых сигналов;
- контроль достоверности путем сравнения с установками «больше/меньше», «предельная скорость», а также путем логического анализа значений взаимосвязанных между собой параметров;



**Рис. 5.12.** Вариант комплектования информационных систем на базе модулей ADAM 40XX:

ADAM 4520 – модуль связи с компьютером (конвертер) RS 232 / RS485;

ADAM 4050 – модуль цифрового ввода-вывода, предусматривающий возможность работы с электронными реле;

ADAM 4013 – модуль аналогового ввода, в том числе и с термометров сопротивления;

ADAM 4060 – модуль релейного цифрового вывода;

ADAM 4012 – модуль аналогового ввода.

- контроль нарушений регламентируемых границ. Для каждого параметра могут быть заданы предупредительные и аварийные установки на нижней и верхней границах.

Модули серии *ADAM 5000 (ADAM 5000/CAN)* – это модули распределенных систем сбора данных и управления. Они предназначены для построения территориально-распределенных систем сбора данных и управления, обеспечивают выполнение следующих функций:

- аналоговый ввод-вывод;
- дискретный ввод-вывод;
- первичное преобразование информации;
- ADAM 5000 осуществляет прием команд от удаленной вычислительной системы и передачи в ее адрес преобразованных данных с использованием интерфейса RS-485. Модуль ADAM 5000/CAN использует интерфейс CAN.

*Модули расширения* (набор модулей ввода-вывода дискретных и аналоговых сигналов, модули приема сигналов термопар и термометров сопротивления).

Например,

- ADAM-5013 – 3-канальный модуль ввода для подключения термометров сопротивления;
- ADAM-5018 – 8-канальный модуль ввода для подключения термопар и т.д.

В начале 1998 года фирма «Advantech» приступила к выпуску новой модели интеллектуальных УСО серии ADAM 5000 – модели ADAM 5510, которая, кроме всего прочего, является свободно программируемым IBM-совместимым микроконтроллером.

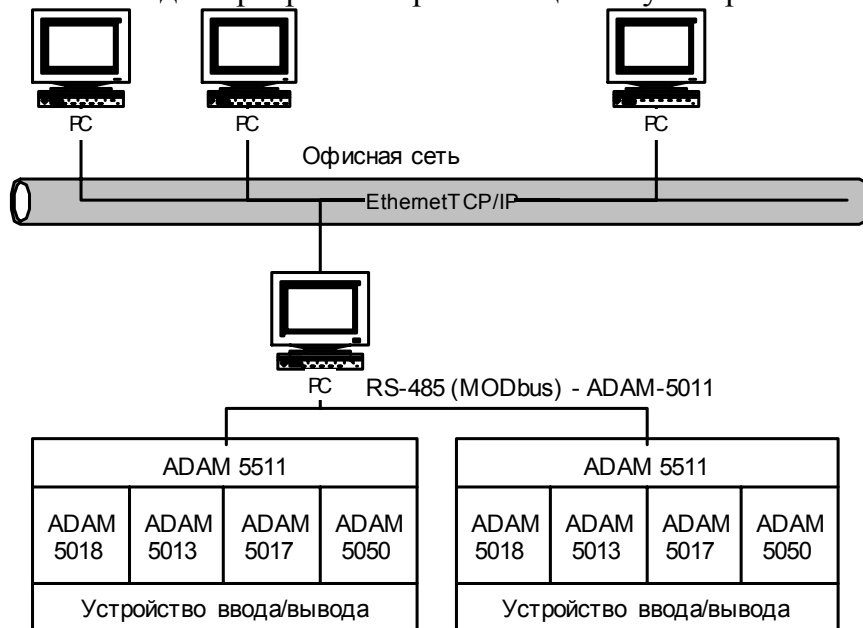
Программируемый микроконтроллер ADAM 5510 предназначен для использования в локальных и распределенных системах автоматизации в качестве автономного контроллера. Он обеспечивает прием и выдачу аналоговых и дискретных сигналов, первичное преобразование сигналов по запрограммированным пользователем алгоритмам и обмен информацией по последовательным каналам связи на базе интерфейса RS-485. Пример комплектования системы на базе модулей серии ADAM 50xx представлен на **рис. 5.13**.

Появление модуля ADAM 5510 позволило расположить управляющую программу пользователя непосредственно в памяти контроллера, что сократило время реакции в контуре управления виртуально до 5–10 мкс. Этот факт позволил применить системы на базе ADAM 5510 для решения задач жесткого реального времени. Контроллер имеет открытую архитектуру и может программироваться с помощью традиционных языков программирования (С, ассемблер), а также функциональных блок-диаграмм. Для этого имеется возможность поддержки пакета Ultralogik (FBD – Function Block Diagram, язык функциональных блок-диаграмм, стандарт МЭК 61131-3). Ultralogik – это система разработки программного обеспечения сбора данных и управления для промышленных контроллеров. Программирующий на Ultralogik может не знать ничего о внутреннем устройстве ADAM 5510. Для него важно понимать какие модули ввода-вывода присутствуют в его контроллере, и на какие именно линии в них

приходят те или иные сигналы. В Ultralogik входят библиотеки для всех модулей ввода-вывода ADAM 5510, библиотеки алгоритмов сбора данных и управления (ПИ-, ПИД-законы и т.д.), предусмотрена возможность подключения функций на языках С, С++, Паскаль, Ассемблер.

Программирование на Ultralogik предусматривает выполнение четырех простых операций:

- определение списка переменных;
- конфигурирование контроллера;
- реализация алгоритмов в виде функциональных блоков;
- отладка программы при помощи симулятора сигналов процесса.



**Рис. 5.13.** Комплектование информационных систем на базе модулей серии ADAM 50XX: ADAM 5018. 8-канальный модуль ввода для подключения терморпар; ADAM 5013. 3-канальный модуль ввода для подключения термометров сопротивления; ADAM 5017. 8-канальный модуль аналогового ввода; ADAM 5050. 16-канальный универсальный модуль дискретного ввода-вывода

Скорость создания работоспособных приложений при использовании Ultralogik может быть выше, чем при использовании на языках класса С или Pascal.

Таким образом, ADAM 5510 удачно сочетает в себе качества программируемого логического контроллера с простой и открытой архитектурой IBM PC-совместимых компьютеров и имеет следующие характеристики:

- совместимый программируемый 16-разрядный микропроцессорный контроллер;
- модуль процессора 80188, 40 МГц;
- флэш-ПЗУ (Flash-ROM) до 256 кбайт;
- объем оперативного запоминающего устройства (ОЗУ) – 256 кбайт;
- операционная система, совместимая с MS-DOS 6.22;

- количество модулей расширения ввода-вывода – 4 серии ADAM 50xx (до 8 в модуле ADAM 5000E).

### **5.3.3. Комплекс технических средств МИК и программных средств MIK\$Sys**

Разработчики – Институт электронных управляющих машин, ТОО «Электронмашсистем» и кафедра автоматики Инженерно-физического института (МИФИ), г. Москва.

Комплекс технических средств МИК предназначен для построения малых, средних и больших распределенных АСУ ТП, использующих локальные сети типа *Вitbus*. В состав технических средств МИК входит программируемый логический контроллер (PLC) МИК СМ9107, моноблоки МБ СМ9207 – регулирующие и широкий набор модулей УСО.

Программируемый логический контроллер МИК СМ9107 представляет собой открытую иерархическую структуру технических и программных средств для построения распределенных (до 250 узлов на расстоянии 12 км) систем управления и сбора информации различного назначения и производительности.

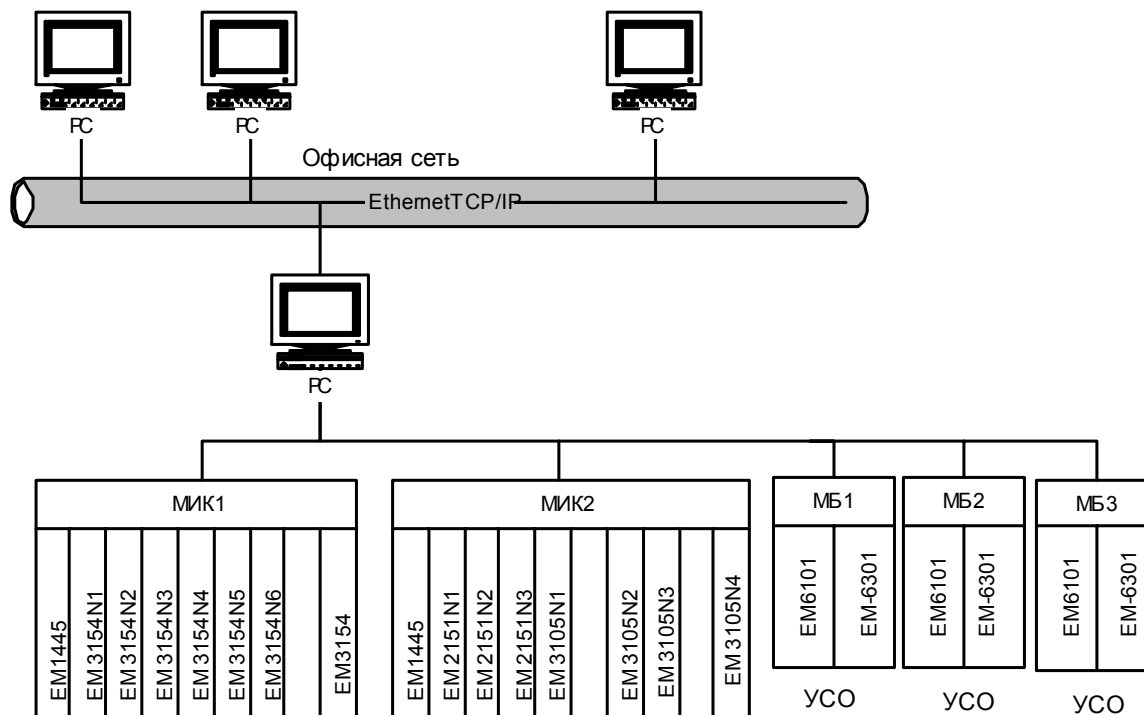
На самом верхнем уровне иерархии используется классическая для промышленных систем схема: «host» (хозяин)–компьютер, в качестве которого обычно используется РС, и множество починенных PLC.

МИК состоит из центрального процессора (ЦП) и набора модулей ввода-вывода, связанных с ЦП быстрым параллельным интерфейсом. Производительность центрального процессора при использовании сравнительно дешевых микропроцессоров фирмы «Dallas Semiconductor» может достигать 8 млн. операций в секунду.

Кроме программируемых логических контроллеров МИК СМ9107, в состав комплекса аппаратуры входят и удаленные регулирующие моноблоки МБ СМ9207. Моноблоки представляют собой систему дешевых микроконтроллеров с малым числом каналов ввода-вывода, которая полностью совместима с контроллерами СМ9107. В моноблок встроен процессорный блок, плата УСО. Использование этих моноблоков позволяет создавать относительно дешевые распределенные информационные системы.

Широкий набор модулей УСО серии ЕМ обеспечивает ввод-вывод сигналов Государственной системы приборов (ГСП) и имеет, как правило, собственный интеллект, что позволяет освободить ресурсы центрального процессора от функций управления ввода-вывода сигналов и первичной обработки информации. В стандартную номенклатуру входят многоканальные (до 48 каналов) модули ввода токовых сигналов 0–5 мА, 4–20 мА, ввода сигналов терморезисторов и термометров сопротивления, ввода дискретных сигналов, вывода аналоговых сигналов по току и напряжению, а также вывода дискретных сигналов (коммутаторы постоянного и переменного тока до 220 В). Такт опроса, отображения и управления в зависимости от сложности системы составляет от 0,05 до 5 сек. Количество входных/выходных или моделируемых аналоговых и дискретных

сигналов – от нескольких десятков до нескольких тысяч. Вариант комплектования системы на базе контроллеров МИК представлен на **рис. 5.14**.



**Рис. 5.14.** Структурная схема информационной системы на контроллерах МИК:  
 МИК – программируемый логический контроллер СМ9107, в составе:  
 центральный процессор (EM1445), модуль ввода сигналов термонар  
 48-канальный (EM3154), модуль ввода дискретных сигналов (EM2151),  
 модуль ввода аналоговых сигналов 0 – 5 мА, 4 – 20 мА (EM3105), модуль  
 источника питания (EM5405);  
 МБ – моноблок регулирующий, удаленный, в составе:  
 процессор (EM6101), модуль УСО (EM6301)

Для работы с контроллерами МИК разработано программное обеспечение MIKSSys, реализованное на языке С и построенное на основе коммерческой версии пакета MIKSSys. В состав программного обеспечения входят:

- встроенный диспетчер задач, поддерживающий до 8 задач в реальном масштабе времени;
- сетевой пакет, обеспечивающий связь удаленных узлов с ПЭВМ;
- система разработки и поддержки графического интерфейса оператора на ПЭВМ (SCADA-система);
- система организации многозадачного режима работы в реальном времени на ПЭВМ;
- программное и алгоритмическое обеспечение первичной обработки информации фильтрации входных данных;
- программное и алгоритмическое обеспечение цифрового управления и регулирования, текущей идентификации параметров объектов управления и оптимальной настройки параметров регуляторов;

- программное обеспечение компактных быстродействующих реляционных СУБД для управления данными на диске и в ОЗУ;
- программное и алгоритмическое обеспечение моделирования тепловых, гидравлических и других динамических объектов в реальном масштабе времени.

Заметим, что отечественные контроллеры МИК и комплекс программного обеспечения МИК\$Sys хорошо зарекомендовали себя в металлургии. В частности, они успешно эксплуатируются на ОАО ММК, где на их основе созданы современные информационные системы на ряде доменных печей, вращающихся и шахтных печах известково-доломитового производства и других агрегатов.

Пример реализации управления режимом работы доменной печи и управления на ОАО ММК на базе комплекса технических средств МИК и программных средств МИК\$Sys мы рассмотрим в следующей главе.

#### **5.3.4. Программируемые логические контроллеры SIMATIC фирмы SIEMENS**

В прошлом название SIMATIC было широко известно во всем мире как синоним программируемых логических контроллеров. Сегодня SIMATIC представляет мощную, полностью интегрированную систему автоматизации (Totally Integrated Automation).

SIMATIC Totally Integrated Automation объединяет целый набор технических и программных средств:

- программируемые логические контроллеры SIMATIC;
- станции распределенного ввода-вывода SIMATIC DP;
- промышленное программное обеспечение SIMATIC;
- человеко-машинный интерфейс SIMATIC HMI;
- промышленные компьютеры SIMATIC PC;
- программаторы SIMATIC PG;
- мощные средства связи SIMATIC NET и другие элементы.

Для иллюстрации принципа построения систем остановимся на краткой характеристике только первых трех групп этой мощной интегрированной системы.

#### **Программируемые логические контроллеры SIMATIC**

Семейство программируемых контроллеров SIMATIC S7-200 (центральные процессоры CPU 2xx), SIMATIC S7-300 (центральные процессоры CPU 3xx), SIMATIC S7-400 (центральные процессоры CPU 4xx) имеют широкие возможности.

Наиболее мощными являются контроллеры SIMATIC S7-400, они предназначены для решения задач автоматического управления средней и высокой степени сложности. Высокая надежность и быстродействие, широкий спектр характеристик и сервисных функций, удобство программирования обеспечивают

успешное использование этих контроллеров в промышленности. Приведем некоторые характеристики контроллера SIMATIC S7-400:

- объем оперативной памяти до 4 Мбайт;
- количество дискретных входов до 32 тыс. и выходов до 65 тыс.;
- количество аналоговых входов/выходов до 8 тыс.;
- контроллеры SIMATIC S7-400 могут подключаться к сетям Industrial Ethernet и Profibus.

Программируемые контроллеры SIMATIC S7-400 способны выполнять функции ведущих устройств сети Profibus-DP. В качестве ведомых устройств Profibus-DP обычно используются станции распределенного ввода-вывода ET200 и контроллеры SIMATIC S7-300.

### *Станции распределенного ввода-вывода SIMATIC DP*

Программируемые логические контроллеры SIMATIC имеют широкие возможности расширения ввода-вывода. Для реализации этих возможностей в распределенных системах предназначена широкая гамма станции распределенного ввода-вывода SIMATIC DP, отличающаяся конструкцией, степенью защиты и своими функциональными возможностями. Так, в распределенных (децентрализованных) системах с интенсивным обменом данными для SIMATIC S7-400 в качестве устройств расширения рекомендуется использовать дешевые системы связи с распределенными системами ввода-вывода ET200, подключаемые к центральному контроллеру по сети Profibus-DP. По этой сети к центральному контроллеру может быть подключено до 96 устройств ET200. При использовании волоконно-оптической линии связи максимальное расстояние между центральным контроллером и последним устройством ET200 может достигать 23 км. Каждый модуль ET200 имеет широкий набор интеллектуальных модулей УСО ввода-вывода дискретных, аналоговых и других сигналов, позволяет использовать в своем составе силовые коммуникационные модули, производящие коммутацию цепей трехфазного переменного тока с нагрузкой мощностью до 5,5 кВт.

### *Промышленное программное обеспечение SIMATIC*

Промышленное программное обеспечение SIMATIC – это комплекс полностью совместимых сервисных программ, позволяющих выполнять проектирование, программирование, настройку, наладку, запуск, диагностирование и эксплуатацию систем автоматического управления, построенных на основе SIMATIC. Все промышленное программное обеспечение SIMATIC можно разделить на четыре класса:

- *Стандартные инструментальные средства*, являющиеся основным программным обеспечением для программирования аппаратуры SIMATIC. Они включают в свой состав пакет STEP 7, отвечающий требованиям международного стандарта МЭК (IEC) 61131-3. Он может работать на персональных компьютерах под управлением Windows 95/98/2000/NT. В состав базового



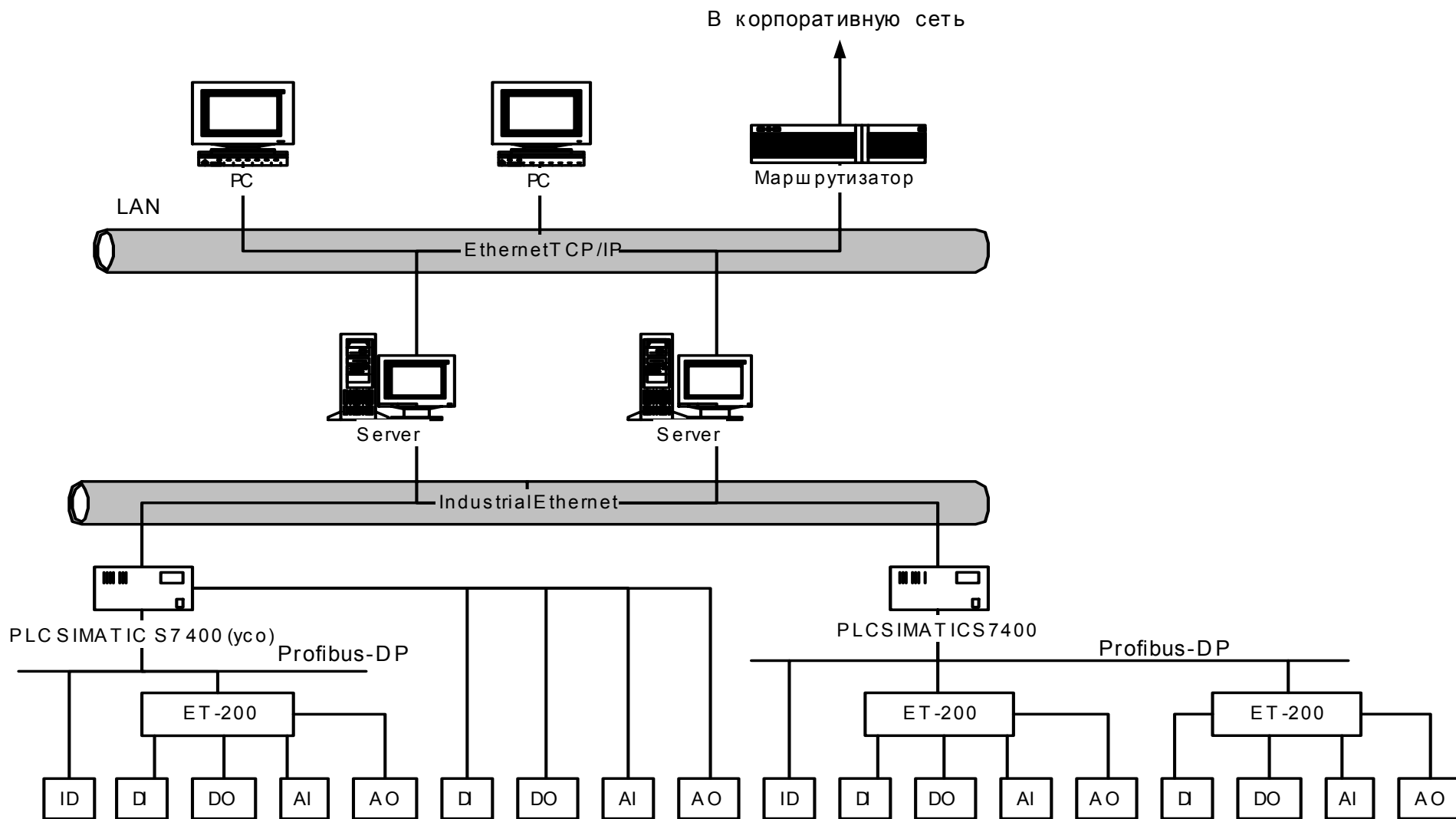
пакета STEP 7 входят все стандартные инструментальные средства программирования систем автоматизации SIMATIC 7, необходимые для выполнения всех фаз создания систем управления:

- конфигурирования и определения параметров настройки аппаратных средств;
- установки вида связи и ее параметров;
- программирования;
- тестирования, запуска и обслуживания;
- подготовки технической документации и архивации данных.

Для разработки программ STEP 7 позволяет использовать следующие стандартные языки программирования:

- список инструкций (Statement List – STL);
  - диаграммы лестничной логики (Ladder Diagram – LAD);
  - диаграммы функциональных блоков (Function Block Diagram – FBD).
- *Инструментальные средства* проектирования используются для расширения функциональных возможностей инструментальных средств. В состав инструментальных средств проектирования входят языки программирования высокого уровня (C/C++), графические языки программирования для решения технологических задач (SFC, диаграммы состояния, функциональные диаграммы), вспомогательное программное обеспечение для диагностики и т.д.
  - *Программное обеспечение runtime*. Позволяет использовать при разработке проектов заранее созданные программные блоки, выполняющие стандартные функции автоматического управления. Программное обеспечение runtime подразделяется на аппаратно зависимое и аппаратно независимое. Аппаратно зависимое обеспечение разрабатывается для конкретного оборудования, например для программирования функциональных модулей PLC. Аппаратно независимое программное обеспечение более универсально и позволяет выполнять стандартные законы управления, модульное управление, управление с нечеткой логикой; выполнять обмен данными между системами управления и стандартными программами Windows, использовать в системах управления многозадачную операционную систему реального времени.
  - *Человеко-машинный интерфейс*, включающий в свой состав программное обеспечение для оперативного управления технологическим процессом и визуализации его протекания. В его состав, в частности, входит универсальная SCADA-система контроля состояния технологического процесса SIMATIC WinCC, работающая под управлением Windows 95/98/2000/NT. Характеристика пакета SIMATIC WinCC приведена в главе 2.

На базе аппаратуры SIMATIC можно спроектировать систему любой сложности. В качестве примера на **рис. 5.15** показан пример комплектования системы на базе аппаратуры SIMATIC. На нижнем уровне в качестве ведомых устройств используются станции распределенного ввода-вывода ET200 и интеллектуальные датчики (ИД), объединенные сетью Profibus-DP, которые осуществляют сбор информации, а также вывод управляющих сигналов на удаленные объекты. В качестве ведущих устройств используются PLC SIMATIC S7. Широкий набор модулей УСО позволяет производить сбор и ввод дискретных и аналоговых сигналов непосредственно и в PLC, а также выдавать непосредственно



**Рис. 5.15.** Вариант построения информационной системы технологического процесса на аппаратуре SIMATIC:  
 ET-200 — удаленные модули связи с объектом; ID — интеллектуальные датчики; DI — дискретный ввод;  
 AI — аналоговый ввод; DO — дискретный вывод; AO — аналоговый вывод

с PLC команды управления на различные исполнительные устройства. Все программируемые контроллеры SIMATIC S7 объединены промышленной сетью Industrial Ethernet. Компоновка верхнего уровня информационной системы традиционна и включает через локальную компьютерную сеть PC Ethernet с протоколом TCP/IP и выход через маршрутизатор в корпоративную сеть предприятия.

Применение серверов баз данных для последовательной обработки и обмена данными между подсистемами PLC SIMATIC S7 (сеть Industrial Ethernet) и PC (сеть Ethernet TCP/IP) позволяет использовать единую базу данных для всех этих подсистем. В качестве серверов и рабочих станций можно рекомендовать промышленные компьютеры SIMATIC PC. В этих компьютерах применяются:

- материнская плата, разработанная и выпущенная фирмой SIEMENS;
- классическая компьютерная технология, использование микропроцессоров Intel;
- промышленное исполнение компьютера (фильтрация охлаждающего воздуха, герметизация корпуса, конструкция, повышающая удобства обслуживания, и т.п.);
- парольная защита от несанкционированного доступа и ряд других решений, что позволяет эксплуатировать компьютеры в тяжелых промышленных условиях.

Заметим, что современные информационные системы, созданные на базе аппаратных и программных средств SIMATIC, успешно функционируют на ряде металлургических предприятий России, в частности на конвейерных обжиговых машинах Лебединского ГОКа, в конвертерных цехах ОАО ММК и ЗАПСИБа, в доменном цехе ЗАПСИБ др.

#### 5.4. Контрольные вопросы

1. На каких принципах строится архитектура современных информационных систем технологических процессов?
2. Какие аппаратно-программные средства используют для обеспечения нижнего уровня автоматизации?
3. Чем отличаются промышленные компьютеры (PC) от промышленных программируемых контроллеров (PLC)? Чем вызвана необходимость использования PC в информационных системах технологических процессов?
4. Какие коммуникационные средства используются на верхнем уровне автоматизации? Для обеспечения решения каких задач их используют? Чем обусловлен выбор тех или иных средств?
5. Для каких целей используют промышленные сети? В чем заключаются их преимущества перед традиционными средствами передачи данных? На какие уровни они подразделяются?
6. Какие функции выполняют физические интерфейсы соединений? Приведите примеры стандартных физических интерфейсов и дайте их краткую характеристику.

---

ГЛАВА 6. ПРИНЦИПЫ ПОСТРОЕНИЯ И РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ (НА ПРИМЕРЕ ДОМЕННОГО ПРОИЗВОДСТВА)	6-2
6.1. Доменная печь как управляемая технологическая система	6-2
6.2. Принципы построения современной автоматизированной информационной системы доменной плавки	6-5
6.3. Пример реализации автоматизированной информационной системы доменной плавки (на примере ОАО ММК)	6-17
6.4. Распределенная система баз данных в аглодоменном производстве	6-25
6.4.1. Особенности разработки системы баз данных	6-25
6.4.2. Особенности функционирования системы баз данных	6-29
6.4.3. Характеристика аппаратно-программных средств вычислительного центра доменного цеха ОАО ММК	6-32
6.5. Контрольные вопросы	6-37

## Глава 6. ПРИНЦИПЫ ПОСТРОЕНИЯ И РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ (на примере доменного производства)

### 6.1. Доменная печь как управляемая технологическая система

В доменной печи в качестве шихтовых материалов используют агломерат, окатыши, металлодобавки, кокс, флюсы (обычно известняк), которые загружаются сверху, через колошник печи. Загруженные материалы опускаются сверху вниз, а газы, образующиеся в горне, поднимаются снизу вверх. Высокотемпературная, восстановительная атмосфера создается в печи в результате подачи высоко нагретого, обогащенного кислородом дутья и инжектируемого топлива, обычно природного газа или пылеугольного топлива, через фурмы в нижнюю часть (горн) печи. В горне печи происходит горение кокса и инжектируемого топлива до оксида углерода и водорода. В процессе плавки происходит восстановление различных элементов, в первую очередь оксидов железа, а кислород оксидов переходит в газ в виде  $\text{CO}$ ,  $\text{CO}_2$  и  $\text{H}_2\text{O}$ . В результате доменного процесса получают продукты плавки – чугуна, шлак, колошниковый (доменный) газ и колошниковую пыль.

Современная доменная печь объемом  $5000 \text{ м}^3$  в сутки выдает 12 тыс. тонн чугуна, 4 тыс. тонн шлака, 20 млн.  $\text{м}^3$  колошникового газа. Для производства этого количества чугуна в печь надо подать 23 тыс. тонн железорудного сырья, кокса, 14 млн.  $\text{м}^3$  горячего дутья и 1,5 млн.  $\text{м}^3$  природного газа. Лучшие отечественные доменные печи, оснащенные современными системами автоматизации, характеризуются расходом кокса примерно 400–410 кг/т чугуна при минимально теоретически допустимом около 300 кг/т чугуна и удельной производительностью по чугуну 2,5–2,7 т/( $\text{м}^3$  сут.). Крупные же доменные цехи заводов имеют в своем составе до 10 доменных печей. Естественно, что такая огромная концентрация мощности в одном технологическом агрегате и в доменном цехе в целом, не имеющая аналогов не только в металлургии, но и других областях техники, требует научно обоснованных, безошибочных методов управления доменным процессом.

Доменная плавка относится к числу непрерывных металлургических процессов, в основу которых положен принцип организованного противотока, что обуславливает высокую экономичность протекания тепловых и массообменных процессов. Так, тепловой коэффициент полезного действия доменной печи достигает 90–95 %, что значительно выше этого показателя на других металлургических печах. Но для этого необходимо организовать хороший контакт газов с шихтовыми материалами, т.е. необходимо обеспечить рациональное распределение газового потока, что является достаточно сложной задачей. В то же время доменное производство остается и самым энергоемким, на долю которого приходится около 50 % топлива, используемого черной металлургией, а энергетические затраты на выплавку чугуна остаются еще высокими и превосходят теоретические значения. При этом основная доля энергетических затрат приходится на дорогостоящий и дефицитный кокс, на экономию которого и направлены основные мероприятия по совершенствованию технологии плавки. Сегодня уро-

вень технологии доменной плавки, в первую очередь, определяется именно расходом кокса. Особенность доменного процесса заключается в том, что топливо в нем выполняет различные функции: как источник тепла; как химический реагент-восстановитель; кокс необходим и как механическая насадка в нижней части для фильтрации расплава. Таким образом, наличие кокса в шихте обуславливает возможность фильтрации газов и расплава, прямого восстановления, тепло- и массообмена. Основная часть кокса (70–80%) сгорает у фурм, остальная потребляется в процессах прямого восстановления и науглероживания чугуна. Эти разные, зачастую противоречивые, функции кокса приходится учитывать при управлении современной доменной плавкой.

Автоматизированный технологический комплекс доменной печи как объект управления характеризуется следующими основными признаками:

- большим объемом контролируемых переменных;
- отсутствием возможности непосредственного контроля отдельных параметров, т.е. низкой прозрачностью процесса выплавки чугуна;
- недостаточной точностью и представительностью результатов контроля параметров, определяющих ход доменной плавки;
- существенным запаздыванием информации о составе продуктов плавки;
- наличием случайных помех различной природы и характера, снижающих качество информации.

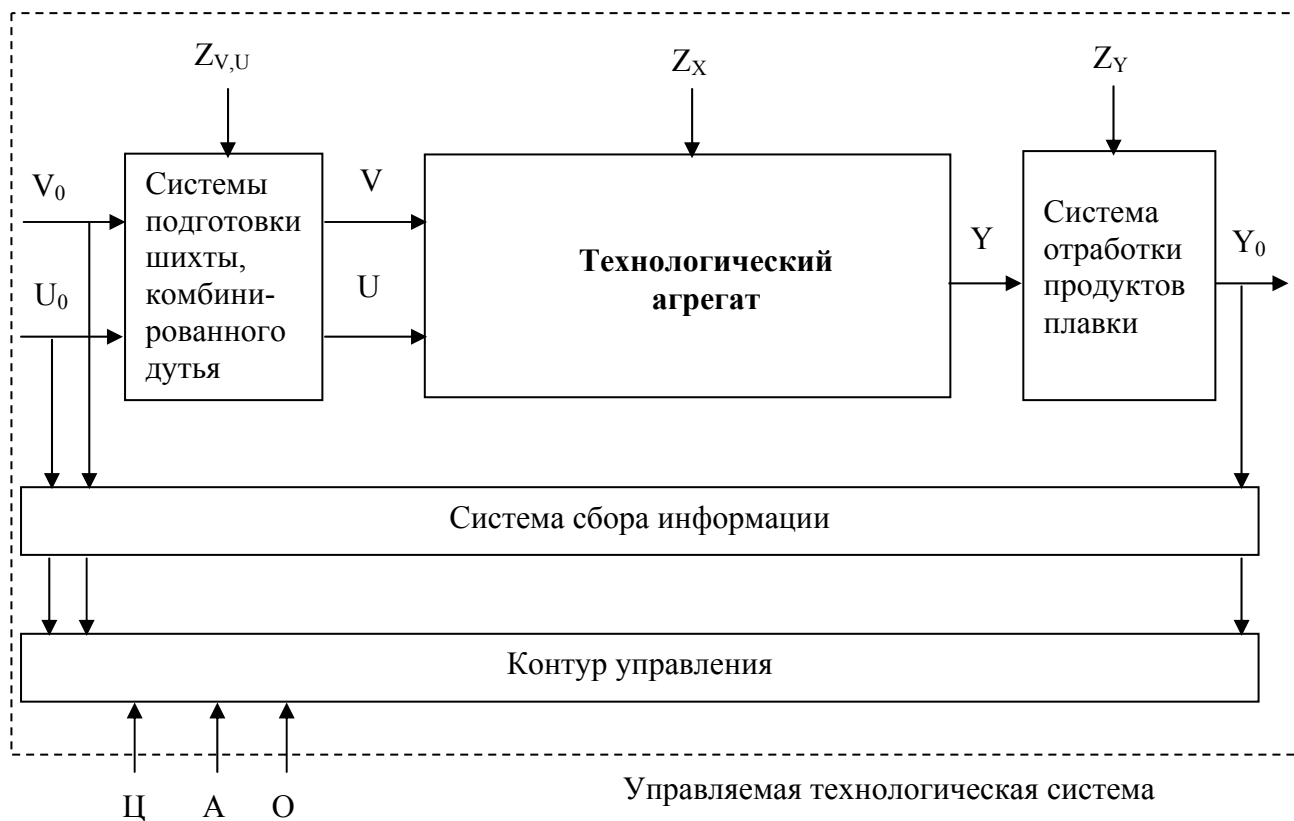
При анализе работы доменной печи ее следует рассматривать как управляемую технологическую систему. Упрощенно комплекс доменной печи может быть представлен схемой, приведенной на **рис. 6.1**. Заметим, что при анализе любого технологического процесса можно выделить три основные группы переменных:

- режимные параметры, отражающие средние значения контролируемых входных воздействий при установившемся состоянии технологического процесса ( $V, U$ );
- средние значения количественных и качественных показателей технологического процесса ( $Y$ );
- возмущения, которые условно можно разделить на внешние, приложенные к входам и выходам процесса ( $Z_{v,u}, Z_y$ ), и внутренние, приложенные к состоянию процесса ( $Z_x$ ).

Показателями (целевой функцией, критериями) ( $\Pi$ ) технологической эффективности работы печи, технологического персонала и информационной системы являются удельный расход кокса, производительность печи, выход кондиционного чугуна по содержанию серы в чугуне, температура жидких продуктов плавки в соответствии с требованиями последующего сталеплавильного производства. При этом должны быть выполнены ограничения ( $O$ ) на технологию доменной плавки, т.е. обеспечена минимальная вероятность попадания технологии состояния печи в так называемые критические области (критические, аварийные режимы работы).

Основной задачей при управлении доменным процессом является стабилизация теплового состояния печи. Главными причинами колебаний теплового состояния являются изменения качества шихты, отклонения температуры и со-

става дутья от заданных значений ( $Z_{V,U}$ ), нарушения в распределении материалов и газов по сечению печи ( $Z_X$ ). Для обеспечения стабилизации теплового режима требуется обеспечивать постоянный баланс основных составляющих режима плавки: нагрев материалов в печи, дренажные свойства шихтовых материалов; положение границ и формы зон вязкопластических материалов и плавления процессов в заплечиках и в фурменной зоне.



**Рис. 6.1.** Доменная печь как управляемая технологическая система:  $V_0, U_0, Y_0$  – отчетные данные о параметрах и показателях работы доменной печи;  $V$  – входные воздействия;  $U$  – управляющие воздействия;  $Y$  – выходные показатели процесса;  $Z_X$  – неконтролируемые изменения внутренних характеристик процесса;  $Z_{V,U}$  – неконтролируемые изменения входных и управляющих воздействий;  $Z_Y$  – потери чугуна со скрапом и шлаком, вынос колошниковой пыли;  $Ц$  – целевая функция управления;  $A$  – алгоритм управления;  $O$  – ограничения

Переходные процессы, связанные с действием возмущений, приводят к изменению теплового состояния процесса и, следовательно, к изменению состава продуктов плавки. При этом доменная печь как объект управления обладает большой инерционностью по отдельным каналам воздействий (постоянная времени достигает 2–4 часа) и запаздыванием (время запаздывания объекта 6–7 часов). Так при изменении состава шихты переходный процесс длится до 15–16 часов. Указанные обстоятельства, безусловно, значительно усложняют процесс управления. Компенсация колебаний химического состава чугуна осуществляется технологами в основном за счет изменения массы кокса в подачу или изменения дутьевых параметров, распределения материалов и газов на колошнике. Величина управляющих воздействий определяется статическими и динамическими

характеристиками процесса. Эти характеристики являются нелинейными и изменяются во времени при колебаниях условий плавки, требуя соответствующего изменения и величины управляющих воздействий.

Современная доменная печь является организованной управляемой интерактивной системой (см. **рис. 6.1**). Состояние этой системы определяется закономерностями комплекса физических процессов, протекающих в доменной печи, особенностями технологии, зависящей от конкретных топливно-сырьевых условий и оборудования доменной печи, включающей системы подготовки шихты, комбинированного дутья, систему отработки продуктов плавки, и принятым способом управления.

В связи с этим при оценке эффективности доменной плавки приходится анализировать не только процесс выплавки чугуна в доменной печи с его комплексом физических закономерностей, но и организованную функциональную систему с системами сбора информации, управления в контуре обратной связи. При этом под системой управления понимаются как используемые технические средства контроля и управления с присущими им метрологическими особенностями, так и участвующие в управлении технологи (лица, принимающие решения), целенаправленные действия которых обеспечивают управление доменным процессом.

Конечной целью разработок информационных систем и систем управления комплексом доменных печей (цехом) является создание системы управления доменным производством, которая обеспечивает нормальную работу доменного цеха в целом при минимальной себестоимости продукции с учетом ограничений, обусловленных требованиями технологии и возможностями оборудования.

Прежде чем познакомиться с принципами построения современной автоматизированной информационной системы доменного производства, рассмотрим обобщенную схему автоматизированной информационной системы.

## **6.2. Принципы построения современной автоматизированной информационной системы доменной плавки**

В настоящее время в промышленно развитых странах все доменные печи в той или иной мере оборудованы АСУ ТП. Функциональные возможности этих АСУ ТП варьируются от чисто информационных до прямого управления процессом. Архитектура автоматизированных систем, как правило, иерархическая с несколькими уровнями управления.

Одной из первых доменных печей, оборудованных многоуровневой распределенной АСУ ТП, является доменная печь № 6 завода фирмы "Кавасаки сэйтэцу" в Тибе, Япония. Первая очередь этой АСУ ТП вступила в строй еще в 1974 году.

При ее создании была использована концепция четырехуровневого построения автоматизированной системы, которая в самых общих чертах сохраняется и в настоящее время.

- *Первый (нижний) уровень управления* – датчики, преобразователи, аппаратура ручного управления.



- *Второй уровень (уровень локального управления)*, включающий ручное дистанционное управление, а также дублирующее цифровое.
- *Третий уровень (уровень цифрового локального управления)*, включающий совокупность микроЭВМ и программно-логических регуляторов, позволяющий действовать в полуавтоматическом режиме, где оператор только дополнительно изменяет задания.
- *Четвертый (верхний) уровень* включает сбор, контроль и хранение информации с помощью управляющей вычислительной машины (УВМ).

На верхнем уровне ЭВМ используются для решения планово-производительных и экономических задач коксохимического, агломерационного и доменного цехов. Средний – две мини-ЭВМ, выполняющие общие задачи управления технологическим процессом: обработка информации, протоколирование, статистический анализ, диагностика и оптимизация процесса. На нижнем уровне используются микроЭВМ (микропроцессоры) и программируемые контроллеры для сбора и первичной обработки информации, для прямого управления локальным участком или процессом.

Одна из первых в бывшем СССР АСУ ТП была разработана и внедрена в 1974 г. на доменной печи № 9 металлургического комбината "Криворожсталь". В дальнейшем АСУ ТП создавались на ряде отечественных доменных печей: №6 НЛМК (1978 г.), №2 КМК (1984 г.), №3 ЗСМК (1988 г.), №3 "Азовсталь", №5 ЧерМК (1986 г.) и др. В последующий период продолжались дальнейшая модернизация и совершенствование этих систем.

АСУТП крупнейшей в мире доменной печи № 5 ЧерМК, объемом 5500м<sup>3</sup> является трехуровневой иерархической распределенной системой.

*Нижний уровень* автоматизации включает средства и системы измерения, контроля, стабилизации и управления технологическими параметрами.

*Средний* – подсистемы автоматического контроля и регулирования отдельными технологическими процессами и агрегатами.

*Верхний* – ряд АСУ и подсистем комплексного контроля и управления технологическим процессом доменной плавки.

Все новые и реконструируемые мощные отечественные и зарубежные печи оснащаются современными многоуровневыми распределенными АСУ ТП. При автоматизации доменных печей используется понятие *рационального объема автоматизации*. Под рациональным объемом автоматизации понимается объем средств и систем автоматического контроля и управления процессом, который соответствует современному состоянию развития технических средств и математического обеспечения и удовлетворяет требованиям технологии производства основной продукции. Сегодня численное значение этого показателя приближается на лучших доменных печах России к 80–90 %, что свидетельствует о высоком уровне автоматизации доменной плавки.

Рациональный объем автоматизации доменных печей включает также три взаимосвязанных функциональных уровня:

1. *Нижний уровень* – средства автоматического контроля и диагностики основных входных, технологических и выходных параметров доменной плавки.

2. Средний уровень – локальные системы автоматического регулирования, стабилизации и диагностики основных технологических параметров.
3. Верхний уровень – распределенные автоматизированные подсистемы управления шихтовкой и дутьем, оценки и управления доменным процессом.

Сегодня АСУ ТП доменных печей представляют собой сложные системы, различающиеся по структуре, назначению, целям и техническим особенностям реализации. АСУ ТП имеют иерархическую, многоуровневую, открытую, распределенную структуру. Каждый уровень отличается своими функциями, характеристиками и используемыми программно-аппаратными средствами. Развитие и совершенствование информационных функций АСУ ТП привело к созданию по сути нового класса информационных систем – *автоматизированным информационным системам технического обслуживания и управления (АИСТОУ)*. По своей сути и решаемым задачам сегодня АСУ ТП доменной печи можно отнести именно к этому классу, т.к. помимо традиционных информационных функций АСУ ТП сегодня существенная роль в силу специфики доменной плавки отводится и следующим задачам:

- обнаружение аварийных ситуаций и выдача необходимой сигнализации;
- диагностика оборудования, электронная архивация данных о состоянии оборудования, ведение архивов технологической информации, журналов событий;
- обеспечение удобного доступа персонала к оперативной и архивной информации, создание и документирование суточных и других документов.

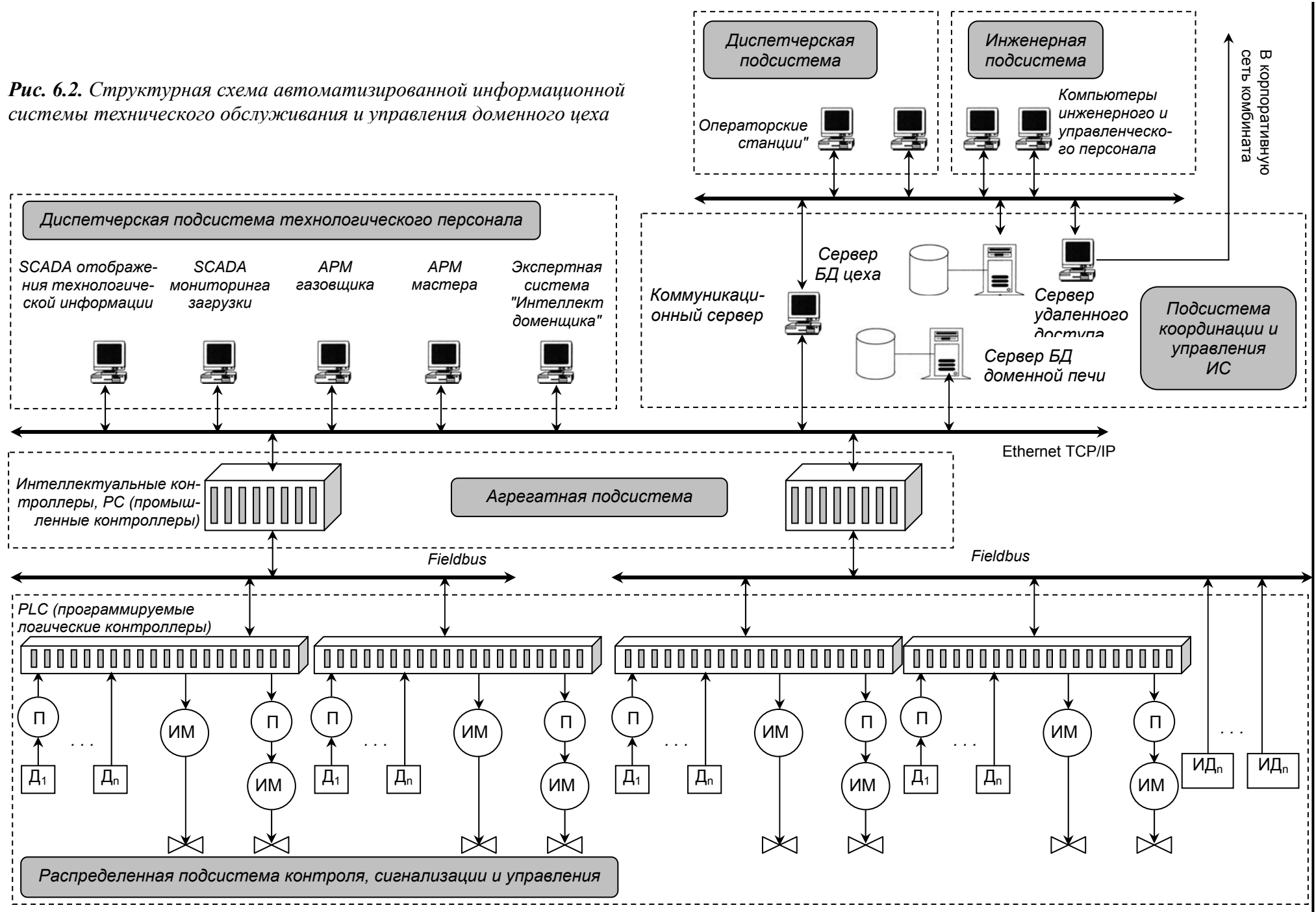
АИСТОУ обычно реализуется с использованием технологии открытых систем реального времени. В ней используют стандартные сетевые средства связи; желательна обеспечение программной совместимости с наиболее распространенными компьютерными платформами; предусмотрены значительные ресурсы в размерах буферов для хранения данных, имеются возможности расширения области применения пакетов, используемых для подключения новых подсистем и прикладных программ.

### *Архитектура АИСТОУ*

АИСТОУ включает следующие основные подсистемы (блоки) (**рис. 6.2**):

- распределенная подсистема контроля, сигнализации и локального управления (РСКУ);
- агрегатная подсистема;
- диспетчерская подсистема технологического персонала доменной печи;
- диспетчерская подсистема доменного цеха;
- инженерная подсистема;
- подсистема координации и управления информационной системой.

Рис. 6.2. Структурная схема автоматизированной информационной системы технического обслуживания и управления доменного цеха



Дадим краткую характеристику функций, выполняемых этими отдельными подсистемами.

### *Распределенная подсистема сигнализации, контроля и локального управления (РСКУ)*

РСКУ является самым нижним уровнем АИСТОУ. Основные решаемые функции:

- сбор данных от рассредоточенных по доменной печи датчиков;
- предварительная обработка собранных данных;
- выработка и реализация ряда управляющих сигналов на основании анализа данных, получаемых от датчиков (автоматическое регулирование отдельных параметров);
- осуществление ручного управления;
- предоставление собираемых данных на диспетчерский уровень АИСТОУ.

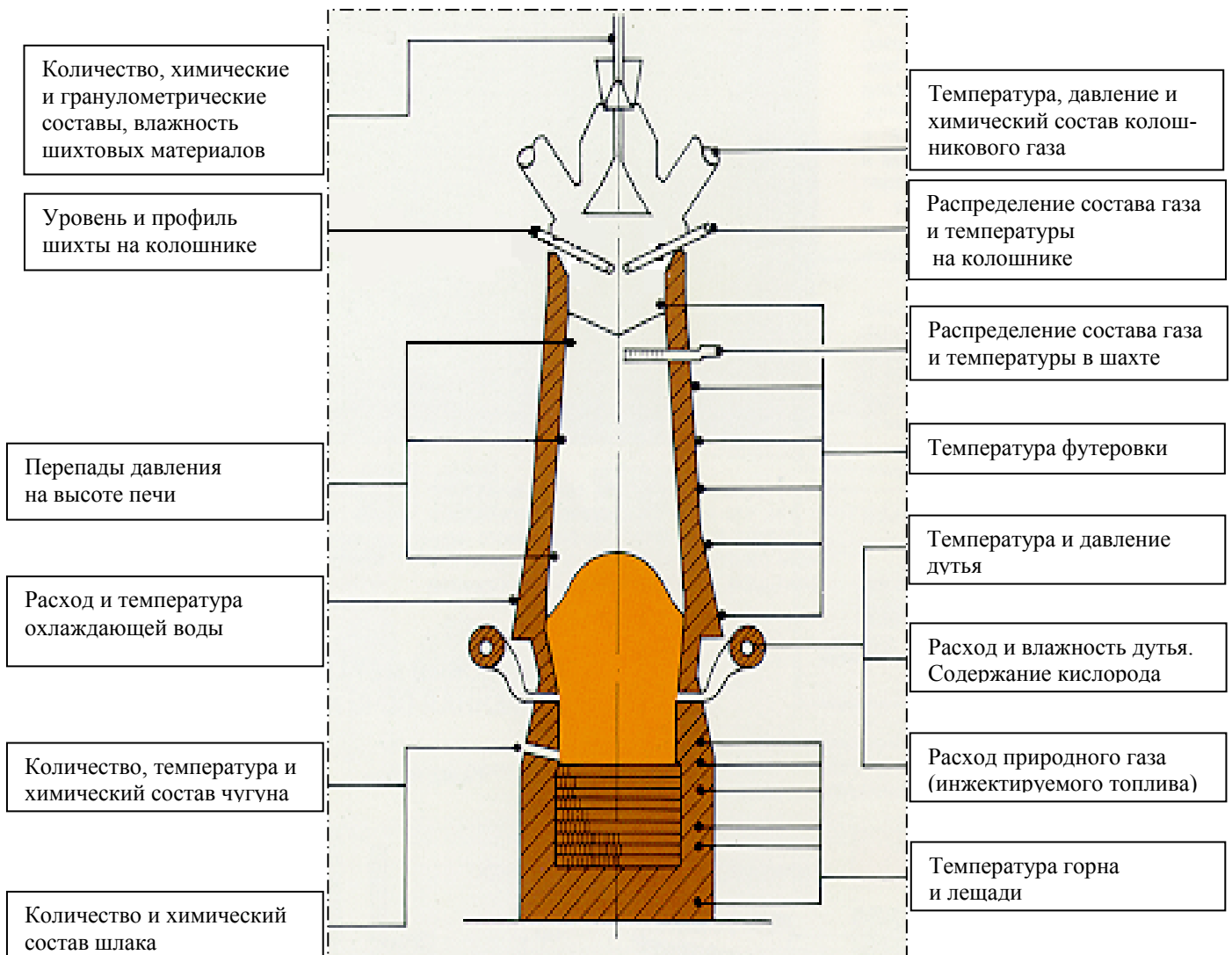
РСКУ формируется из датчиков, преобразователей, исполнительных механизмов, регулирующих органов и связанных с ними контроллеров. Поскольку перечисленные средства должны обеспечивать возможность построения локальных контуров управления, данный уровень должен работать в реальном времени. Время реакции систем, используемых в доменных печах, на этом уровне не должно превышать нескольких десятков миллисекунд. Огромные размеры доменной печи, повышенные значения температур, загазованность, резкие перепады температуры окружающей среды, высокий уровень вибрации и другие, предъявляют особые требования надежности средств этого уровня. При этом необходимо обеспечение возможности работы контроллеров в автономном режиме в критических местах при отказе сети. Надежность оборудования, способного работать в тяжелых условиях эксплуатации, оценивается временем наработки на отказ, превышающим 50000 часов (свыше 5 лет) в непрерывном круглосуточном режиме эксплуатации.

Обобщенная структурная РСКУ изображена на **рис. 6.2**. Основные функции подсистемы реализованы на основе контроллеров.

Опыт работы лучших отечественных и зарубежных мощных доменных печей свидетельствует о том, что для выплавки качественного чугуна с минимальными затратами необходимо контролировать более 1,5 тыс. характеристик входных и выходных параметров плавки, технологических параметров самого доменного процесса и работы оборудования печи, для чего используется более 3000 единиц аппаратуры. Основные подсистемы контроля технологических параметров представлены на **рис. 6.3**. Большое количество контролируемых параметров требует свертки информации в обобщенные показатели входа, выхода и состояния доменного процесса.

Для контроля ряда технологических параметров (температур, давления, расходов газовых сред и т.п.) широко используется типовая, стандартная аппаратура отечественного и импортного производства. За последнее десятилетие в отечественном и зарубежном доменном производстве опробовано и достаточно большое количество специальных средств измерения технологических параметров на базе классических и новых методов, не применявшихся ранее.

В частности, получили распространение бесконтактные уровнемеры, использующие радиометрический и радарный микроволновый принцип контроля уровня профиля засыпи на колошнике. Перспективной является автоматическая зондовая сканирующая система DDS-МИСиС, разработанная в Московском институте стали и сплавов, позволяющая контролировать распределение железорудных материалов и кокса под уровнем засыпи на работающей доменной печи. Промышленные образцы этой системы успешно эксплуатируются на доменных печах ОАО Северсталь. На ряде доменных печей применяют устройства регистрации электрических явлений на стальном кожухе печи, возникновение которых аргументируется ионной теорией жидких расплавов. Такие устройства позволяют косвенно контролировать уровень расплава в горне печи, ведутся испытания многоцелевого экспериментального устройства диагностики доменного процесса ТЕГРА с применением термодатчиков, расположенных непосредственно в горне печи. Устройство позволяет диагностировать нормальный нагрев, похоло-



**Рис. 6.3.** Основные подсистемы контроля технологических параметров доменной печи

дание и разогрев горна, ровный, тугой и быстрый ход печи. На ряде доменных печей России разработана и внедрена НПО «Черметавтоматика» система контроля диагностики разгара горна («Футеровка»), которая включает в себя специальные термодатчики, устанавливаемые по длине блоков при замене кладки печи, комплекс средств автоматизации, промышленный компьютер и специальное алгоритмическое программное обеспечение. Сегодня полностью автоматизирована на большинстве доменных печей загрузка шихтовых материалов и управление работой воздухонагревателей.

В то же время и сегодня недостаточно разработаны системы контроля распределения газов по радиусу печи, непрерывный контроль состава и физических свойств шихтовых материалов, загружаемых в печь, и жидких продуктов плавки на выпусках. Так, применяющиеся в основном на отечественных доменных печах ручные методы отбора и подготовки проб доменного сырья не обеспечивают надлежащую оперативность процесса, низка и надежность систем контроля перепадов статического давления на нескольких участках по высоте доменной печи. В настоящее время отсутствуют удовлетворительные методы контроля положения и формы зоны вязкопластичного состояния материалов. Недостаточный уровень контроля важнейших параметров доменной плавки, несомненно, снижает ее эффективность, а разработка соответствующих датчиков является сегодня одной из первоочередных задач.

Следует отметить, что сложность доменного процесса, необходимость определения оптимальных параметров доменной плавки при изменении условий ее ведения, возрастающая цена ошибок управления при управлении тепловым и газодинамическим режимами – эти и другие факторы определяют потребность в разработке новых информационных систем. Анализ современного уровня знаний основных закономерностей доменного процесса, имеющихся технических средств контроля, существенное расширение возможностей вычислительной техники и области применения методов математического моделирования показывают, что наиболее перспективными методами для диагностики состояния доменной печи являются, наряду с инструментальными, расчетно-инструментальные и компьютерные. Именно примеры таких систем мы рассмотрим позднее.

Приведем перечень основных подсистем РСКУ АИСТОУ.

*Подсистема управления шихтоподачей* осуществляет:

- автоматический контроль химического состава шихтовых материалов и их запасов в расходных бункерах бункерной эстакады;
- автоматическое управление процессом порционного дозирования (набор, хранение и выгрузка отдельных доз шихтовых материалов по каналам дозирования по заданной программе);
- адаптивную динамическую компенсацию контролируемых возмущений при дозировании шихтовых материалов;
- автоматическое управление формированием порций шихтовых материалов по заданной программе;
- управление загрузочным устройством;
- контроль уровня и профиля засыпи, скорости опускания шихты на колошнике.

**Подсистема управления дутьевым режимом** осуществляет:

- контроль и регулирование основных параметров блока воздухонагревателей;
- контроль давления и расхода холодного дутья;
- регулирование соотношения расходов "природный газ – холодное дутье";
- стабилизацию температуры горячего дутья;
- стабилизацию влажности горячего дутья;
- регулирование давления природного газа;
- контроль содержания кислорода в дутье;
- управление распределением природного газа по фурмам доменной печи.

**Подсистема управления доменным процессом** выполняет:

- регулирование давления газа под колошником;
- контроль температуры и состава колошникового газа;
- контроль перепадов статического давления по высоте печи;
- контроль температуры поверхности засыпи;
- контроль состава и температуры газов над (под) уровнем засыпи;
- контроль системы охлаждения и прогара фурм доменной печи;
- диагностику состояния футеровки печи;
- контроль температуры брони;
- контроль состава, температуры массы чугуна и шлака на выпуске.

### **Агрегатная подсистема**

Основные решаемые функции:

- обработка данных, включая масштабирование;
- синхронизация работы подсистем;
- организация архивов по выбранным параметрам;
- резервирование каналов передачи данных;
- визуализация накопленной информации;
- обмен информацией с диспетчерским уровнем и подсистемами нижнего уровня;
- хранение данных.

Аппаратная база должна быть более мощной, предусматривающей возможность обмена через промышленные и офисные сети с нижним и верхним уровнем. Обычно используют промышленные компьютеры PC совместимой архитектуры или интеллектуальные контроллеры.

### **Диспетчерская подсистема технологического персонала доменной печи**

Решает следующие задачи:

- сбор данных от аппаратуры нижнего уровня;
- оценка технологического состояния доменной печи;
- диагностика режима плавки;
- обнаружение аварийных ситуаций и необходимая сигнализация;

- ведение технологических архивов и журналов событий;
- визуализация технологического процесса, текущих и архивных данных, просмотр журналов;
- компьютерная поддержка принятия решений диспетчерского уровня;
- диспетчерское управление.

Диспетчерский уровень образуют несколько (в зависимости от полноты и сложности решаемых задач) персональных компьютеров, расположенных в помещении главного щита контроля и управления доменной печью и образующих автоматизированные рабочие места оперативного персонала доменной печи. При этом используются, чаще всего, промышленные компьютеры PC, ОС Windows 95/98/NT/2000, технологии Internet/Intranet.

Рассмотрим основные подсистемы диспетчерского уровня.

**Система централизованного контроля хода технологического процесса и состояния оборудования (АРМ газовщика)** включает ряд подсистем.

**Подсистема отображения технологической информации.** Предназначена для своевременного обеспечения сменного технологического персонала (по его вызову) оперативной информацией о доменном процессе в удобной для пользователя форме. В автоматизированной системе интерфейс связи человека (оператора) с ЭВМ приобретает особое значение. Двойное назначение этого интерфейса предъявляет к нему требования обеспечения легкого и быстрого восприятия информации человеком, за которым остается "последнее слово" в принятии решения, а также быстрый и безошибочный ввод человеком информации в ЭВМ. Задача построения графического интерфейса достаточно сложна. Необходимо учитывать множество факторов, в частности структуру имеющихся подсистем, объем и типы собираемой информации, профессиональные навыки и уровень компьютерной подготовки персонала, сложность решаемых задач и т.п.

Общение с ЭВМ специалиста-доменщика (мастера, газовщика и т.п.), не подготовленного специально для работы на ЭВМ, встречают серьезные трудности. Для этих целей предусматриваются специальные программные средства общения человека с ЭВМ, не требующие специальных знаний в области программирования. Поскольку АРМ работает под управлением ОС Windows, то в основу пользовательского интерфейса обычно заложены аналогичные диалоговые элементы: окна, строковые опускающиеся меню, панели инструментов и т.д. Кроме этого, а в АРМ имеется встроенная справочная система, для минимизации ручного ввода служат специальные значения, принимаемые по умолчанию. Достигнутая в результате эргономичность пользовательского интерфейса облегчает эксплуатацию АРМ и значительно сокращает этап обучения персонала. Свойства любого параметра можно узнать из соответствующего информационного окна, где отображаются название параметра, контролируемая подсистема, точность измерения и т.д.

**Подсистема отработки жидких продуктов плавки.** Предназначена для просмотра и редактирования информации о постановках ковшей и миксеров под печь, а также информации о выпуске жидких продуктов плавки.

**Подсистема шихтоподачи.** Предназначена для просмотра и редактирования информации о ходе загрузки, работе засыпного устройства, количестве



подач в смену, системе загрузки, а также информации о расходах железорудных материалов, кокса и флюсов в подаче и т.п.

**Подсистема представления отчетных данных (электронный рапорт).** Предназначена для формирования, хранения и получения твердой копии отчетно-технологической информации о работе комплекса доменной печи по заранее согласованным с руководством цеха форме; вычисления усредненных за заданные интервалы времени (час, смена, сутки) значений параметров плавки; формирования и выдачи на экран монитора по запросу сменного технологического персонала справочных форм сменно-суточной отчетности о работе печи от начала смены на текущий момент времени или за предыдущие смены.

**Подсистема формирования предупредительных сообщений.** Предназначена для обнаружения отклонений важнейших параметров процесса и наиболее ответственных элементов системы охлаждения печи от их нормального состояния и своевременного информирования об этом сменного технологического персонала в виде соответствующих предупредительных сообщений. Отклонения контролируемых параметров (например, верхнего и нижнего перепадов давления по высоте печи, температур периферийных термопар и в газоотводах, перепадов температуры охлаждающей воды на леточных горновых и верхних холодильниках лещади) от их нормального состояния определяются по выходу их текущих величин за заданные пределы.

**Система диагностики технологического состояния печи (АРМ мастера доменной печи)** включает ряд подсистем.

**Подсистема оценки и прогноза теплового состояния доменной печи.** Предназначена для оценки текущих и прогнозирования будущих показателей производительности печи, температуры чугуна и содержания кремния, удельного расхода и минимально возможного расхода (резерва) кокса, а также диагностики отклонений от нормального режима плавки по нагреву и развитию тепло- и массообменных процессов. Позволяет рассчитывать корректировку массы кокса и флюсующих материалов в подаче при изменении соотношения железорудных компонентов, состава кокса, параметров комбинированного дутья, химического состава чугуна.

**Подсистема оценки газодинамического режима и ровности хода доменной печи.** Система предназначена для стабилизации ровности хода доменной печи, предотвращения аномальных и критических состояний, оценки и планирования газодинамического резерва. К числу критериев устойчивости хода доменной печи относятся интенсивность хода по коксу и критерий газодинамической устойчивости слоя. Газодинамический резерв контролируется по разности предельно допустимого и фактического перепада давления газа в шахте. Оценивается также ход печи (ровность хода печи, развитие канального и периферийного хода и т.п.).

**Подсистема контроля и управления шлаковым режимом.** Предназначена для оценки шлакового режима и вязкопластического состояния шихты и обеспечивает выполнение следующих функций:

- расчет массы железорудных материалов и кокса в подаче для получения шлака с заданными свойствами (вязкости, текучести, десульфурирующей способности и т.д.);

- расчет шихты на заданные показатели;
- оценку газодинамической напряженности работы низа печи.

**Подсистема диагностики состояния фурменных очагов доменной печи.** Позволяет для конкретных конструктивных и режимных параметров устройств ввода дутья, параметров дутьевого режима определять геометрические размеры фурменного очага, поля температур, состава и скоростей газа, температур, траекторий и скоростей движения кусков кокса, частиц инжестируемого топлива.

**Подсистема диагностики газораспределением, распределения материалов на колошнике доменной печи и прогноза температурных полей в шахте.** Подсистема выполняет следующие функции:

- определение структуры столба шихтовых материалов;
- оценку расходов газовых потоков, процессов теплообмена по радиусу в высоте печи;
- оценку распределения шихтовых материалов по радиусу колошника;
- прогноз температурных полей газа и материалов в шахте с учетом неравномерности распределения материалов и газов.

**Подсистема контроля положения и формы зоны вязкопластичного состояния материалов.** Одним из возможных методов косвенного контроля является метод математического моделирования процессов плавления, основанный на совокупной оценке доступной информации, в частности, о перепадах статического давления в этой зоне, распределении материалов и газов на колошнике и т.п.

В АРМ мастера предусмотрено широкое использование математических моделей. Заметим в связи с этим, что анализ в целом состояния вопроса по реально используемым математическим моделям в практике технологии ведения доменной плавки показывает: в настоящее время разрыв между потенциальными возможностями средств автоматизации и реальными возможностями используемого программного обеспечения огромен. *Поэтому первостепенной задачей, по мнению ведущих отечественных и зарубежных специалистов, является создание эффективного математического обеспечения решения технологических задач доменной плавки.* Основное внимание при этом направлено на решение следующих задач:

- оценка, прогноз и регулирование теплового состояния доменной печи с целью уменьшения расхода кокса и получения стабильного качества чугуна по физическому нагреву и химическому составу;
- диагностика состояния фурменной зоны как основополагающего процесса газодинамики и газификации углерода кокса, а также обеспечение полноты горения инжестируемого топлива;
- диагностика состояния зоны плавления, размягчения как лимитирующей распределение газов и возможности интенсификации плавки;
- диагностика технологических ситуаций с целью предотвращения аварийных ситуаций.

Именно на этих вопросах и остановимся более подробно в дальнейшем.

**Подсистема "Эксперт".** Это экспертная система на базе искусственного интеллекта позволяет решать ряд технологических задач прогнозирования

расстройств хода печей и возможности принятия в нештатных ситуациях быстрых и достоверных решений в условиях неполноты и противоречивости информации (см. подробнее далее).

### *Диспетчерский уровень доменного цеха (АРМ диспетчера цеха)*

Решает следующие основные задачи:

- просмотр информации о потребителях чугуна;
- ввод информации о поставках и задержках ковшей;
- просмотр информации о производстве чугуна (наливе и сливе);
- просмотр информации о движении ковшей, миксеров;
- отображение текущего состояния печей;
- средний анализ шихты за смену;
- текущий анализ окатышей, агломерата, кокса;
- расходы шихтовых материалов;
- показатели работы печей и цеха;
- анализы чугуна и шлака;
- анализ колошникового газа.

Для аппаратного обеспечения обычно используются компьютеры PC-совместимой архитектуры, ОС Windows 95/98/NT/2000.

### *Инженерный уровень*

Вырабатывает стратегию управления технологией доменного цеха. Применяются компьютеры PC-совместимой архитектуры, ОС Windows 95/98/NT/2000, технологии Internet/Intranet. Инженерный уровень включает компьютеры инженерного и управленческого персонала доменного цеха и осуществляет:

- предоставление информации со всех доменных печей для просмотра инженерно-техническим и управленческим персоналом;
- визуализацию, архивирование и документирование технологической информации, используемой на инженерном уровне;
- решение комплекса инженерных задач в области стратегии управления доменной плавки, в частности анализа возможных режимов плавки при изменении свойств поступающего железорудного сырья и кокса, распределение имеющихся ресурсов между отдельными печами и т.п.;
- защиту от несанкционированного доступа к данным.

### *Подсистема координации и управления информационной системой*

Актуальность создания подобной системы обусловлена необходимостью хранения и управления большим количеством информации, интеграцией локальной сети доменного цеха в единое информационное пространство комбината, высокой степенью концентрации технических и программных средств информационных систем. Система обеспечивает, в частности, выдачу оперативных

сообщений о нарушениях и сбоях, возникающих в процессе работы информационной системы; ведение информационной базы и накопление данных по сбоям, отказам и перезапускам систем; формирование и выдачу по запросу персонала документов, характеризующих качество функционирования системы.

Таким образом, уровень предназначен:

- для создания и ведения базы данных. Включает в себя систему управления базой данных как отдельных доменных печей, так и всего цеха в целом;
- обеспечения функционирования локальной сети доменного цеха и интеграции ее в корпоративную сеть комбината;
- оценки состояния технических и программных средств информатизации. Позволяет получать данные о качестве функционирования программных и технических видов обеспечения АИСТОУ.

### **6.3. Пример реализации автоматизированной информационной системы доменной плавки (на примере ОАО ММК)**

Проиллюстрируем идеологию построения современной автоматизированной системы на примере печи № 1 доменного цеха ОАО ММК, схема которой представлена на **рис. 6.4**.

**Распределенная подсистема контроля и локального управления (РСКУ).** Входная информация о работе каждой доменной печи разделена на блоки.

1. Параметры, требующие периодичности опроса 2–5 секунд (критические параметры), такие как анализ схода шихтовых материалов, время срабатывания подач, перепады давления по высоте печи, давление колошниковога газа, горячего дутья, работа ВРШ, дискретные сигналы и т.д.
2. Блок остальных входных данных, автоматически контролируемых непосредственно на печи: анализ состава колошниковога газа, температуры периферийных термопар, колошниковога газа по газоотводам, влажность, температура, расход и состав горячего дутья, расходы природного газа и технологического кислорода, расходы кокса и железорудных материалов и т.п. Периодичность опроса состояния этих автоматически контролируемых параметров составляет 30–60 секунд.
3. Автоматический или полуавтоматический ввод контролируемых данных в ПК индустриального исполнения Pentium 200 рабочего места газовщика: химический состав железорудного сырья, его механическая прочность, содержание мелочи; технический анализ кокса, его прочность; состав жидких продуктов плавки, график выпусков жидких продуктов плавки и т.п. Данные в АСУ «Контроль за поступлением сырья в доменный цех» поступают из трех источников: по привозному сырью – по электронной почте сертификаты с Соколовско-Сарбаевского ГОКа, по местному сырью – с АСУ «Учет поступления и движения сырья на складах ЦПАШ» (цех подготовки аглошихты), по агломерату – с АСУ «Контроль работы аглофабрик». Поступающая из АСУ «Контроль за поступлением сырья в доменный цех»

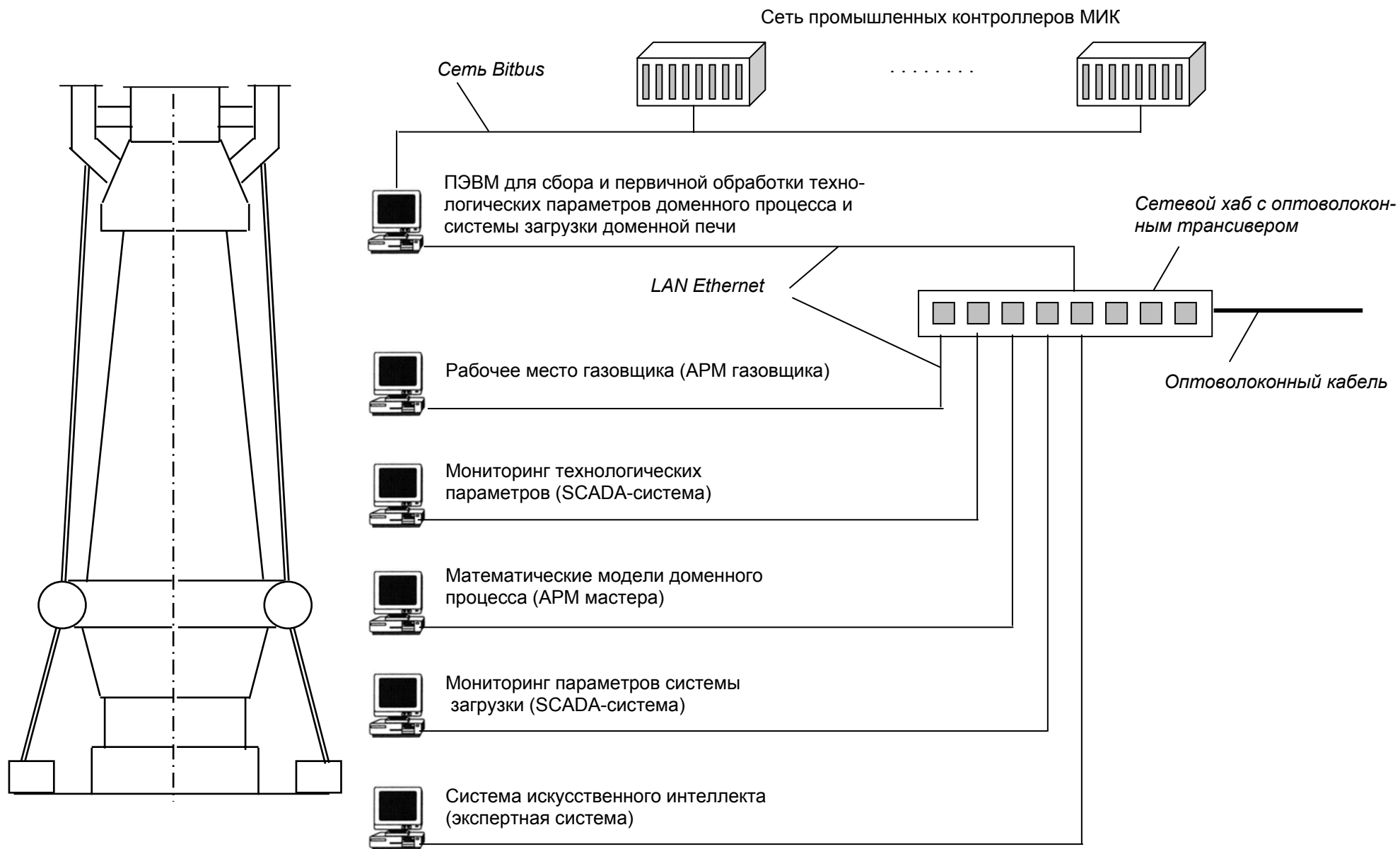


Рис. 6.4. Структура автоматизированной информационной системы доменной печи №1 ММК

информация о химическом составе и физических свойствах очередной партии сырья, а также информация о результатах химического анализа чугуна и шлака обновляется по мере ее поступления.

Уровень РСКУ АИСТОУ оснащен продукцией отечественных производителей. В частности, на этом уровне применяются отечественные промышленные контроллеры МИК (разработчики – Институт электронных управляющих машин, ТОО «Электронмашсистем» и кафедра автоматики МИФИ, г. Москва). Они предназначены для построения распределенных информационных систем, использующих промышленные сети типа BitBus (стандарт фирмы Intel). В состав технических средств МИК входят программируемый логический контроллер МИК СМ9107 и широкий набор модулей УСО. Программное обеспечение МИК построено на основе коммерческой версии пакета MIK\$\$Sys. На верхнем уровне иерархии используется классическая для промышленных систем схема «host» (хозяин)-компьютер, в качестве которого используется персональный компьютер индустриального исполнения Pentium 233, и множество подчиненных PLC, что обеспечивает на различных уровнях аппаратную и программную совместимость МИК с другими системами. Опыт эксплуатации контроллеров MIK\$\$Sys подтверждает их достаточную надежность, удобство их применения и использования имеющегося для них программного обеспечения. В качестве РС для сбора и первичной обработки параметров доменного процесса и системы загрузки используется Pentium 233 с оперативной памятью 32 Мбайт и емкостью жесткого диска 6.4 Гбайт.

### *Диспетчерский уровень доменной печи*

Для реализации диспетчерского уровня на доменных печах №1 впервые центром АСУ комбината были разработаны и внедрены:

- *система управления технологическим процессом с частичной заменой вторичных приборов на компьютерную систему мониторинга технологических параметров доменной печи;*
- *система мониторинга параметров систем загрузки доменных печей.*

Главными функциями этих по сути дела SCADA-систем являются отображение технологического процесса в виде мнемосхем на экране монитора компьютера, сигнализация об аварийных ситуациях, обеспечение компьютерного общего управления процессом со стороны оператора-технолога, возможность создания архивов аварий, событий и поведения переменных процесса во времени (тренды), а также полное или выборочное хранение параметров процесса через заданные промежутки времени постоянно или по условию и др.

В настоящее время на рынке имеется ряд программных продуктов отечественных и зарубежных фирм, специализирующихся на разработке универсальных SCADA-программ. Однако центру АСУ ОАО ММК удалось создать собственную SCADA-систему, в значительно большей степени учитывающую богатый опыт работы технологического персонала доменного цеха и особенности контроля и управления современной доменной плавкой (**рис. 6.5, 6.6**). Результатом данного подхода стала возможность предоставления информации не только о ходе технологического процесса, но и его истории (тренда), начиная от техно-

логического персонала доменного цеха, и кончая диспетчером комбината и руководителем любого ранга в любой точке корпоративной сети комбината. В качестве средств разработки прикладных программ применялись Microsoft Visual C++, Microsoft Visual Basic. Опыт промышленной эксплуатации SCADA-системы показал их высокую надежность и удобство разработанного интерфейса человек-машина MMI (Man Machine Interface) для решения технологических задач контроля и управления ходом доменной плавки.

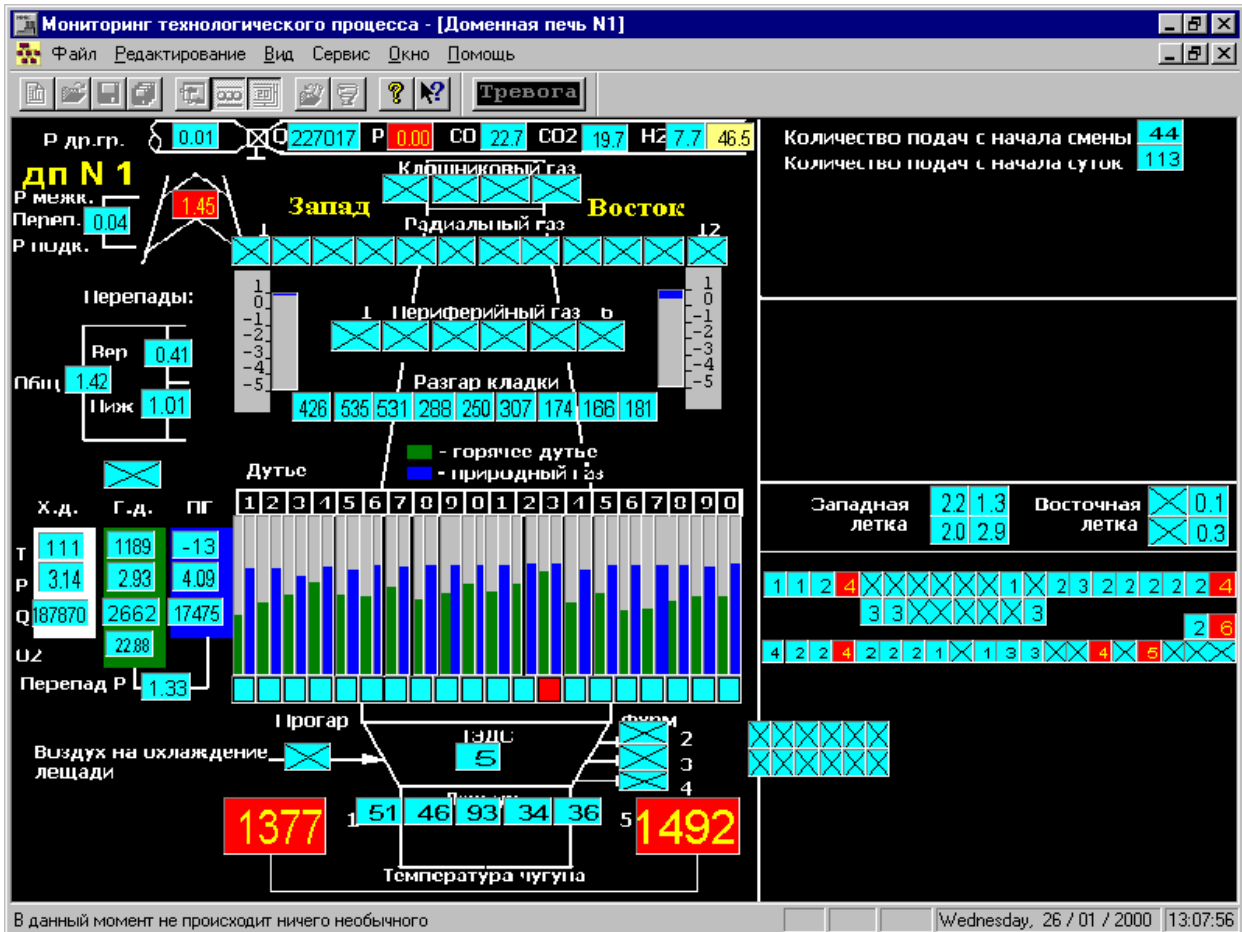


Рис. 6.5. Окно SCADA-системы технологического процесса доменной печи №1 ММК

Задачи, решаемые в области управления технологическим режимом доменной плавки, относятся к классу плохо структурированных задач, поэтому осуществление управления технологией даже при высоком уровне компьютеризации и математического моделирования без участия человека невозможно. В связи с этим перспективным является использование информационных технологий поддержки принятия решений. В системах поддержки принятия решений, основу которых составляет целый комплекс (база) математических моделей, решающих различные задачи, пользователь имеет возможность получить недостающую ему информацию для принятия решения путем установления диалога с математической моделью.

Комплекс математических моделей и пакетов прикладных программ позволяет решать ряд задач в области доменной плавки. Остановимся только на некоторых из них.



Рис. 6.6. Окно SCADA-системы для наблюдения за уровнем засыпи доменной печи №1 ММК

### *Интегрированный пакет «Автоматизированное рабочее место газовщика доменной печи»*

Пакет позволяет решать задачи ввода и редактирования информации:

- о выпуске чугуна и шлака – просмотреть и отредактировать информацию о постановках ковшей и миксеров под печь, а также информацию о выпуске жидких продуктов плавки;
- о шихтоподаче – просмотреть и отредактировать информацию о количестве подач в смену;
- о системах загрузки доменной печи – просмотреть и отредактировать информацию о системе загрузки, а также информацию о расходах железорудных материалов, кокса и флюсов в подаче.

Пакет обеспечивает расчет и просмотр следующих документов: химический состав загружаемых материалов и жидких продуктов плавки, технологических и технико-экономических показателей работы печи (рис. 6.7).

В качестве РС автоматизированного рабочего места газовщика используется персональный компьютер индустриального исполнения Pentium 200 с опе-



ративной памятью 32 Мбайта и емкостью жесткого диска 4 Гбайта, ОС Windows NT/2000. В качестве средств разработки прикладных программ применялся Microsoft Visual C++.

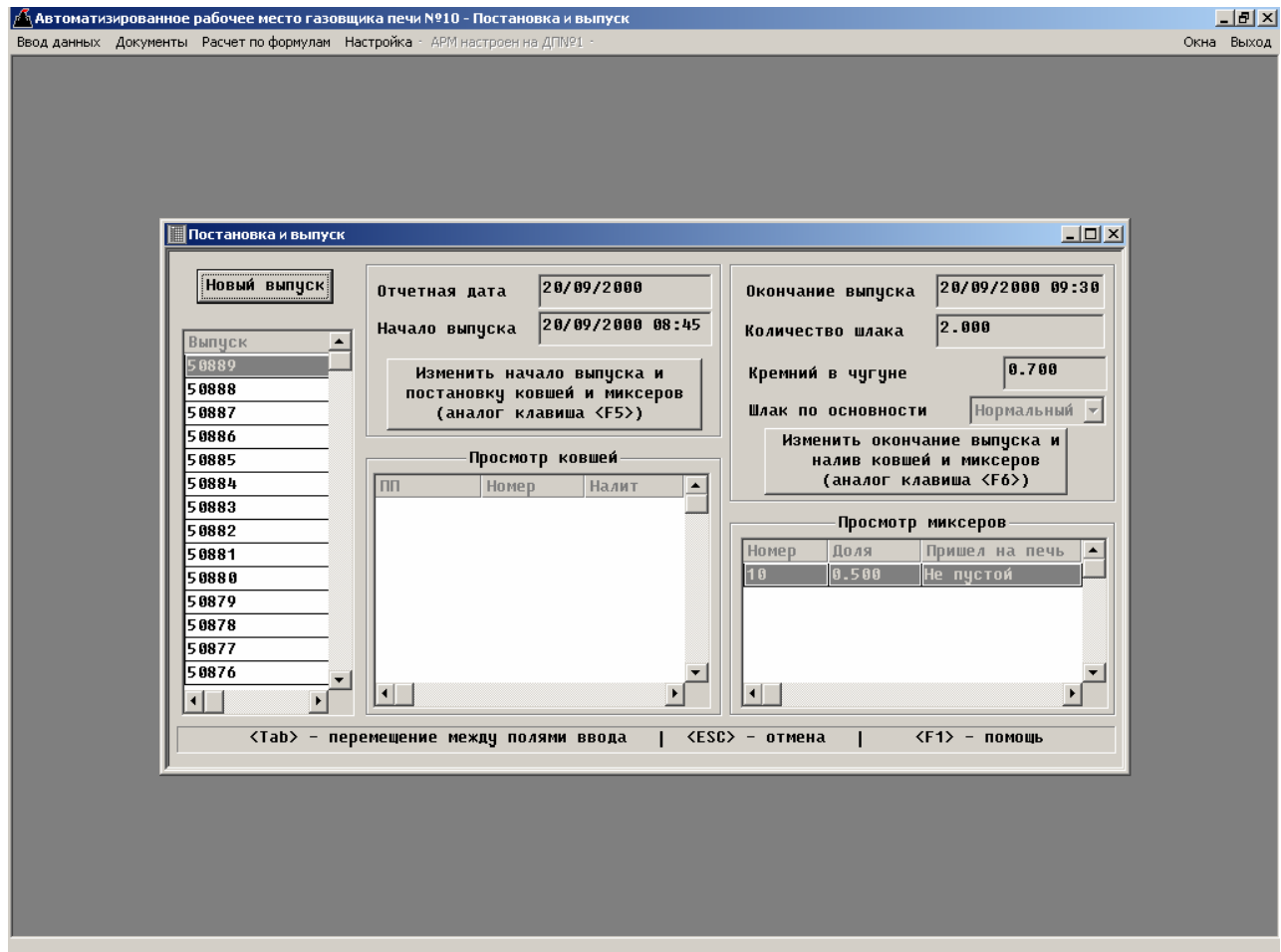


Рис. 6.7. Пример отображения информации АРМ-газовизика

**Интегрированный пакет «Автоматизированное рабочее место мастера доменной печи».** Пакет позволит решать задачи коррекции:

- массы кокса и флюсующих материалов в подаче при изменении соотношения железорудных компонентов, а также состава кокса;
- массы железорудных материалов и кокса в подаче для получения шлака заданных свойств;
- массы флюсующих материалов и кокса в подаче для получения заданного химического состава чугуна;
- массы кокса в подаче при изменении параметров комбинированного дутья (температуры и влажности дутья, расходов природного газа и технологического кислорода);
- прогноза изменения показателей доменной печи при изменении параметров плавки (параметров комбинированного дутья, характеристик загружаемого сырья и т.п.).

Так же, как и в АРМ газовщика, в автоматизированном рабочем месте мастера в качестве вычислительной базы используются два персональных компьютера индустриального исполнения Pentium 200 с оперативной памятью 32Мбайта и емкостью жесткого диска 4 Гбайта, ОС Windows NT/2000, а в качестве средств разработки прикладных программ применяется Microsoft Visual C++.

#### ***Экспертная система "Интеллект доменщика".***

Позволяет осуществлять:

- диагностику нарушений хода доменной печи;
- диагностику нарушений теплового режима.

Система содержит два модуля, которые используются для определения:

- причин (факторов), вызывающих соответствующие нарушения хода печи;
- причин (факторов), вызывающих соответствующие нарушения теплового режима доменной плавки.

Экспертная система "Интеллект доменщика" реализована на персональном компьютере индустриального исполнения Pentium 200 с оперативной памятью 32 Мбайта и емкостью жесткого диска 4 Гбайта, ОС Windows NT, а в качестве средств разработки прикладных программ используется Microsoft Visual C++.

#### ***Инженерный уровень***

В качестве аппаратного обеспечения инженерного уровня используются 3 персональных компьютера обычного офисного исполнения Pentium 200 с оперативной памятью 32 Мбайта и емкостью жесткого диска 2 Гбайта, ОС Windows 98/2000. В качестве средств разработки прикладных программ применялись Microsoft Visual C++, Microsoft Visual Basic. Пользователями этих компьютеров являются заместитель начальника цеха по технологии, инженеры технологического отдела.

***Пакет «Решение технологических задач доменной плавки».*** Пакет предназначен для технологического персонала доменного цеха и включает в себя три функционально связанных модуля, выполняющих:

- оценку базового (фактического) состояния доменного процесса;
- сопоставительный анализ работы доменных печей в различные периоды;
- прогноз изменения показателей доменной печи при изменении параметров плавки.

Эксплуатация этого пакета показала его эффективность для решения целого комплекса задач, встречающихся в практике управления технологией плавки в доменном цехе комбината.

***Пакет «Оптимизация работы доменного цеха».*** Разработанное программное обеспечение позволяет выбирать оптимальные значения параметров комбинированного дутья: расходы кислорода и природного газа на каждой из печей цеха при остановке отдельных печей цеха, изменении расходов природного газа, кислорода, состава и качества шихтовых материалов, поступающих в цех в различных технологических ситуациях работы цеха (максимум производства

чугуна, минимум расхода кокса по цеху и т.п.) В модели учитываются ограничивающие и лимитирующие факторы по газо-гидродинамическим, тепло-массообменным процессам на каждой из печей цеха, конъюнктура рынка.

**Пакет «Расчет состава агломерата для получения заданных свойств шлака».** Разработанный пакет прикладных вычислительных программ позволит производить расчеты, связанные как с определением объема производимого агломерата на аглофабриках, так и состава этого агломерата, исходя из характеристик имеющихся привозных железорудных материалов при условии получения надлежащих свойств шлака (вязкости, температуры плавления, обессеривающей способности) и обеспечения рациональных технико-экономических показателей доменной плавки.

### *Подсистема координации и управления информационной системой*

В настоящее время с использованием реляционной модели данных сформирована база данных доменного цеха, отражающая работу всех доменных печей по количеству и качеству выпускаемой продукции, количеству и качеству проплавленных шихтовых материалов, потреблению энергоносителей, основным технологическим параметрам доменной плавки. Для хранения данных о работе доменной печи в этой сети на сервере базы данных установлена система управления базами данных Microsoft SQL. В качестве сервера базы данных используется Pentium II с оперативной памятью 256 Мбайт и емкостью жесткого диска 6,4 Гбайта. Это позволило практически полностью решить в доменном цехе проблемы хранения, контроля, защиты, ввода, редактирования и извлечения информации, а также формирования необходимых отчетных данных.

Создание локальных компьютерных сетей для отдельных печей по архитектуре клиент-сервер, объединение их в распределенную компьютерную сеть всего доменного цеха с выходом последней в корпоративную компьютерную сеть ОАО ММК обеспечили единое информационное пространство и взаимосвязь с другими цехами и службами комбината, а также подсистемами цехов и производств. В качестве операционной системы для локальных сетей доменного цеха и отдельных печей хорошо себя зарекомендовала операционная система Microsoft Windows NT Server 4.0, а в качестве физической передающей среды широко используются экранированная витая пара и волоконно-оптический кабель. Выход сети доменного цеха в корпоративную сеть комбината осуществляется по двум дублирующим друг друга каналам. Основной – по беспроводной технологии, базируется на использовании радиоканала на частоте 2,4 ГГц и устройства организации радио сети ARLAN фирмы Aironet Wireless Communications, Inc, позволяющей обеспечивать скорость передачи данных 2 Мбит/с. Второй – проводной, использует выделенный двухжильный канал, основан на использовании высокоскоростных модемов Megabit Modem 768 фирмы PairGain Technologies, Inc и позволяет достигать скорости передачи данных 768 кбит/с. Следует особо отметить, что в настоящее время даже в локальных, распределенных сетях и корпоративной сети на комбинате широко используется Intranet (внутренняя сеть)-технология, что позволило широко использовать на-

дежные, удобные технологии и программное обеспечение глобальной сети Internet.

Проведенные мероприятия позволили организовать контроль и управление технологическим процессом на всех этапах – от складирования сырья в цехе подготовки агломерационной шихты до выплавки чугуна. Сегодня автоматизированная информационная система доменного цеха тесно связана с другими информационными системами, в первую очередь с теми, от которых зависит качество выплавляемого чугуна: цеха подготовки агломерационной шихты, агломерационного цеха, коксохимического производства, кислородно-конвертерного цеха, железнодорожного транспорта, кислородно-компрессорного производства и др.

#### **6.4. Распределенная система баз данных в аглодоменном производстве**

Структура информационного обеспечения сложных металлургических процессов, к которым относится аглодоменное производство, во многом определяет качество, себестоимость, а значит, и конкурентоспособность готовой металлопродукции на отечественном и мировом рынке. Наряду с развитием методов и средств сбора сигналов с технологических объектов совершенствуются методы накопления, хранения, обработки и представления технологической информации. Эти методы основаны на построении распределенной системы баз данных со сложной логической структурой. Такой подход к организации данных позволяет создавать сложные автоматизированные информационные системы, которые используются как для управления отдельными технологическими процессами, так и металлургическим производством в целом.

Проиллюстрируем идеологию разработки и функционирования современной распределенной системы баз данных, которая положена в основу автоматизированной информационной системы аглодоменного производства ОАО ММК.

##### **6.4.1. Особенности разработки системы баз данных**

Основными целями разработки системы баз данных являлись следующие:

- комплексное информационное обеспечение аглодоменного производства, т.е. накопление, хранение и представление технологической информации на всех этапах производственного цикла – от складирования сырья в цехе подготовки агломерационной шихты до выплавки чугуна в доменных печах;
- возможность реализации в доменных печах математических моделей для управления технологическими процессами, т.е. использование накопленных данных для получения и представления новой информации о ходе технологического процесса;
- использование текущей информации для экспертной оценки состояния технологического процесса.

Реализация поставленных целей позволяет обеспечить технологическому персоналу всестороннюю – информационную, модельную и экспертную – поддержку для принятия обоснованного решения об оказании управляющего воздействия на объект.

Современная методология разработки распределенных систем баз данных, имеющих сложную логическую структуру, базируется на использовании средств автоматизированного проектирования информационных систем, или так называемых CASE-средств (Computer Aided Software Engineering).

Для разработки информационной системы доменной печи использовались программные CASE-продукты VPwin и ERwin американской компании Logic Works.

Программа VPwin представляет собой компьютерную систему функционального моделирования технологических процессов, основанную на методологии проектирования информационных систем IDEF0. Методология IDEF0, принятая в качестве стандарта многими международными организациями, охватывает совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта предметной области. Функциональная модель отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями. С помощью набора графических инструментов для отображения действий и объектов была построена функциональная схема доменного производства, на которой обозначены исходные данные и результаты основных производственных операций, ресурсы, необходимые для их выполнения, управляющие воздействия, взаимосвязи между отдельными элементами моделируемой системы. Построенная функциональная модель, состоящая из набора иерархически связанных диаграмм, была использована для анализа, оптимизации производственного процесса, а также послужила основой для построения концептуальной модели информационной системы доменного производства.

На **рис. 6.8** представлен фрагмент разработанной функциональной модели доменного производства, которая состоит из следующих функциональных блоков:

- *Учет поступления шихтовых материалов в доменный цех.* Входной информацией являются данные о количестве и качестве сырья – железорудных материалов, кокса и флюсующих добавок, поступающих на бункерную эстакаду доменного цеха. На данном этапе выполняется логический контроль вводимых данных, запись их в базу данных, проверка полноты базы данных, заполнение технологической документации. Логический контроль данных осуществляется на основе нормативно-справочной информации (НСИ), а расчет и заполнение документации – по технологическим инструкциям. В результате формируется база данных, содержащая полную информацию по поступившему сырью, а также комплект документов о поступивших шихтовых материалах.
- *Учет проплавленных шихтовых материалов.* Исходными данными является информация о количестве и качестве загружаемых в доменную печь шихтовых материалов. На данном этапе производится полуавтоматический ввод этой информации в базу данных, а также проверка

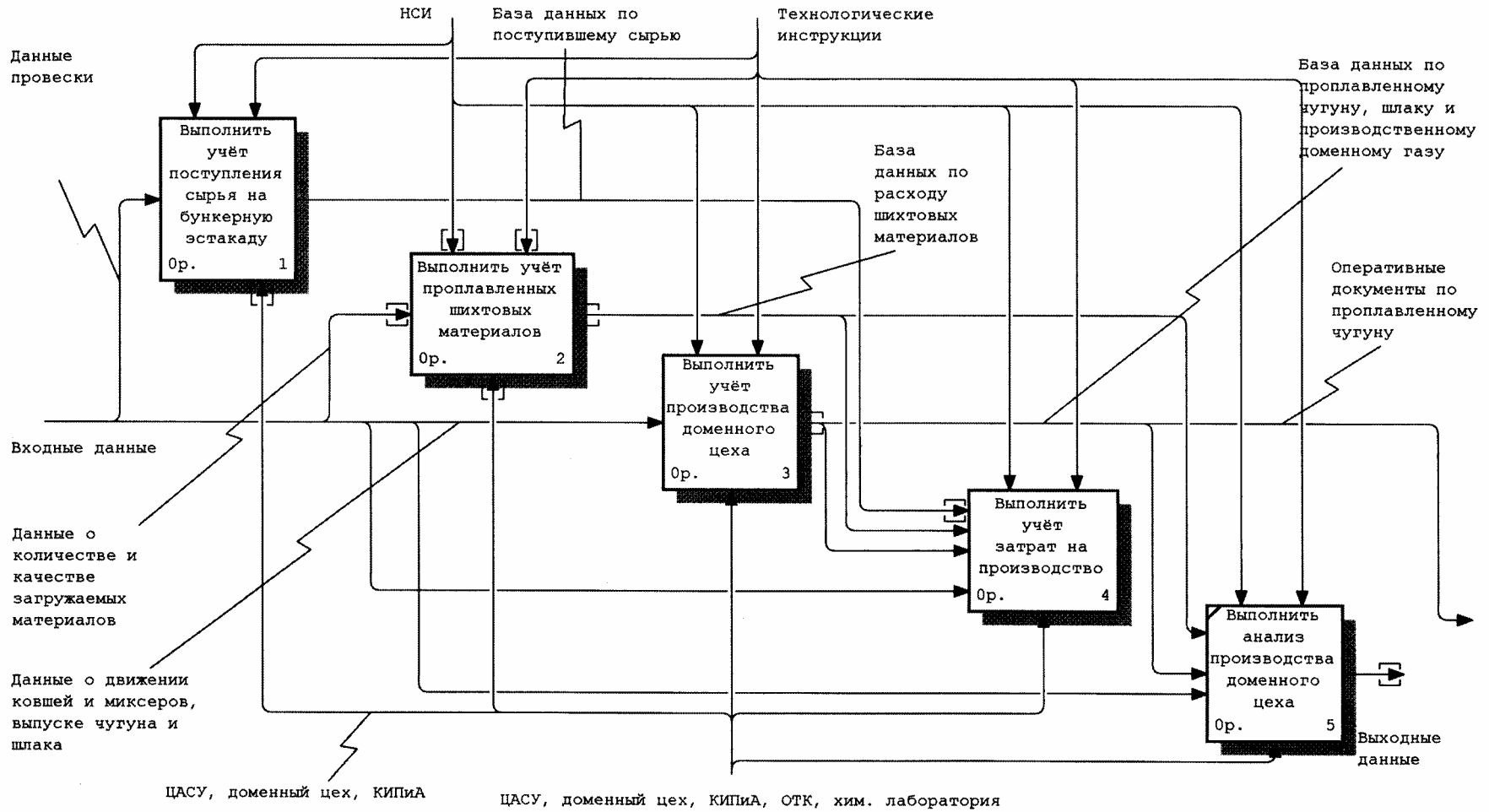


Рис. 6.8. Фрагмент функциональной модели доменного производства

ее на непротиворечивость. В результате формируется база данных по проплавленному чугуна, шлаку и произведенному доменному газу.

- *Учет производства доменного цеха.* Производится с целью накопления информации о движении ковшей и миксеров, количестве выплавленного чугуна, количестве и качестве произведенного шлака, полученного доменного газа. Сбор данных носит распределенный характер: в частности, информация о налитых ковшах и миксере поступает от газовщиков доменных печей, о весе чугуна, налитого в ковши, – с весового поста, о передвижении ковшей с жидким чугуном и миксеров – от диспетчера доменного цеха и т.д. Собранная информация в виде временных файлов подвергается логическому контролю и после корректировки записывается в базу данных. Одновременно производится расчет и формирование документов о количестве произведенного чугуна, произведенного шлака и полученного доменного газа.

*Учет затрат на производство.* В ходе данного этапа происходит накопление оперативной информации о количестве использованного кокса, флюсующих добавок, кислорода, природного газа. Источниками информации являются сформированные на предшествующих этапах базы данных по поступившему сырью, по расходу шихтовых материалов, по проплавленному чугуна, шлаку и полученному доменному газу. Логический контроль и расчет затрат производится на основании нормативно-справочной информации и технологических инструкций.

- *Анализ производства доменного цеха* производится по следующим показателям: качеству полученных чугуна и шлака; использованию кислорода; использованию природного газа; экономическим затратам на производство. Информация для данного блока автоматически поступает из баз данных по составу поступающего в доменный цех сырья, по расходу шихтовых материалов, по проплавленному чугуна, шлаку и полученному доменному газу. Данный этап заканчивается расчетом технико-экономических показателей, комплексным экономическим анализом доменного производства с выдачей оперативной документации.

Реализация информационной системы доменного производства осуществлена с использованием программы ERwin, которая представляет собой компьютерную систему информационного моделирования предметной области и предназначена для визуального проектирования схемы базы данных и ее автоматической генерации в файлы для выбранной реляционной СУБД. Методологическую основу ERwin составляет технология IDEF1X, в которой моделирование предметной области осуществляется с помощью графических ERD-диаграмм (Entity–Relationship Diagrams), или диаграмм «сущность–связь». Разработчик описывает структуру базы данных визуально в виде объектов (сущностей) с их атрибутами, служащих прообразами реляционных таблиц, и при помощи мыши указывает связи между этими сущностями, которые являются прототипами реляционных отношений.

Учитывая результаты функционального моделирования, специфику технологии доменной плавки, особенности доменной печи как объекта информационных систем, требования к качеству представляемой информации, были выде-

лены следующие основные объекты для проектирования базы данных доменного цеха:

- *Доменная печь*, включающая информацию о конструктивных характеристиках печей доменного цеха,
- *Шихтоподача*, которая отражает информацию о системах загрузки и типах и расходах загружаемых материалов.
- *Химический анализ шихтовых материалов*, отражающий информацию о физико-химических свойствах загружаемых материалов и времени проведения химического анализа и определения физических свойств.
- *Химический анализ кокса*, содержащий информацию о физико-химических свойствах используемого кокса и времени проведения химического анализа и определения физических свойств.
- *Технологические параметры* доменной плавки, отражающие основные контролируемые характеристики дутьевых параметров, отходящего колошникового газа, характеризующих газодинамический режим плавки.
- *Характеристики чугуна и шлака*, отражающие временные интервалы обработки продуктов, их химический состав и физический нагрев.

Пример процедуры построения логической и физической схем базы данных аглодоменного производства с помощью CASE-средства ERwin приведено ранее (см. главу 3).

Основным преимуществом использования CASE-продуктов, в частности программы Egwin, является то, что разработанная таким образом логическая схема не привязана к технологии какой-либо конкретной фирмы, поставляющей определенный тип СУБД или средства разработки прикладного программного обеспечения. Благодаря этому разработанная логическая структура данных может быть преобразована в физическую схему базы данных, ориентированную на работу с конкретной СУБД. В доменном цехе ОАО ММК в качестве основной программной платформы для функционирования информационной системы доменного производства используется сервер баз данных Microsoft SQL Server, который контролирует ввод, хранение, защиту, корректировку и извлечение информации для формирования необходимых отчетных данных.

#### **6.4.2. Особенности функционирования системы баз данных**

Особенностью эксплуатации системы баз данных в аглодоменном производстве является территориально-распределенный характер сбора и обработки технологической информации с использованием для передачи данных технологии локальных вычислительных сетей (ЛВС), серверов удаленного доступа и цифровых радиосредств. Информация накапливается и обрабатывается в базах данных, которые расположены на территориально-распределенных серверах с установленным специальным программным обеспечением – системой управления базами данных (СУБД), работающей по технологии клиент-сервер.

Логическая структура информационной системы отражает работу основных участков аглодоменного производства, поэтому система баз данных содержит следующие основные информационные компоненты (**рис. 6.9**):



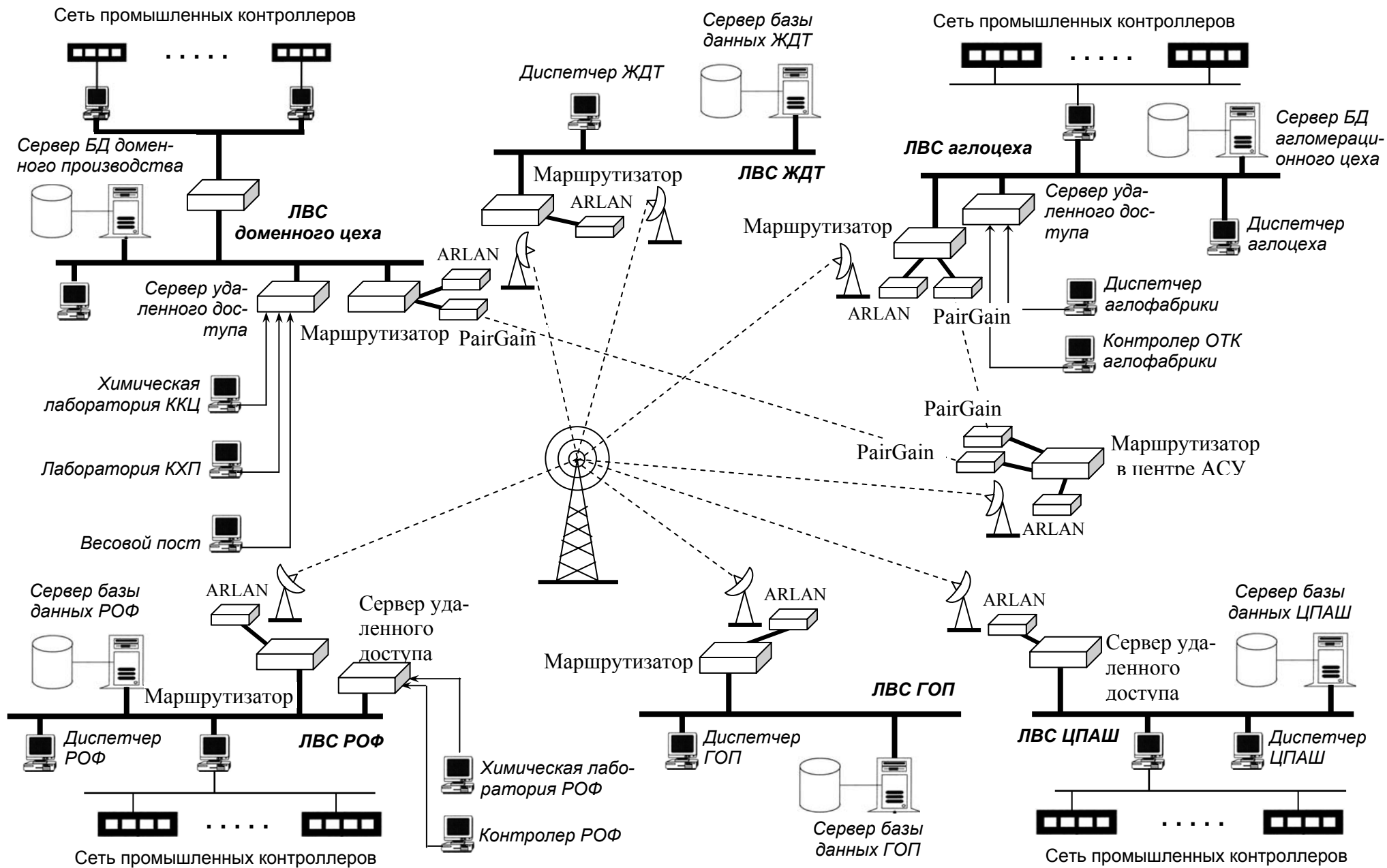


Рис. 6.9. Схема компьютерных сетей по обеспечению управления аглодоменным производством

- *база данных цеха подготовки агломерационной шихты (ЦПАШ)*, которая функционирует на сервере базы данных ЦПАШ. Основная часть данных в базу вводится вручную и корректируется диспетчером ЦПАШ, к которому поступает информация с весового поста, от машиниста экскаватора, расцепщика и др. Через сеть промышленных контроллеров автоматически поступают данные о технологическом процессе подготовки агломерационной шихты. В результате на сервере базы данных ЦПАШ формируется полная информация о поступлении и отгрузке сырья для производства агломерата со складов, качестве полученного сырья и подготовленной шихты, перечень сертификатов на привозное сырье и др.;
- *база данных рудно-обогажительной фабрики (РОФ)*, установленная на сервере базы данных РОФ, который автоматически через сеть промышленных контроллеров накапливает технологическую информацию о процессе подготовки железорудных материалов к доменной плавке. Вручную производится ввод данных о поступлении сырья и отгрузке продукции со складов. Через сервер удаленного доступа в базу данных поступает также информация о химическом анализе железорудных материалов, а также от контролера ОТК. Диспетчер РОФ осуществляет просмотр информации и при необходимости осуществляет корректировку ошибочно введенных данных. По накопленной информации рассчитываются суммарные материальные и энергетические затраты на подготовку и обогащение руды;
- *база данных агломерационного цеха*, установленная на сервере базы данных в агломерационном цехе. С помощью сети промышленных контроллеров обеспечивается автоматический ввод и контроль основных технологических параметров агломерационного производства и представление его в базе данных. Автоматизированный ввод информации через сервер удаленного доступа осуществляют контролер ОТК и диспетчер агломерационной фабрики. Диспетчер агломерационного цеха имеет возможность просматривать и корректировать введенные в базу данные. Накопленная информация позволяет производить оперативную оценку качества готового агломерата, учет поступления сырья и отгрузки готовой продукции со склада, анализ энергетических и материальных затрат на агломерационное производство;
- *база данных горно-обогажительного производства (ГОП)*, которая расположена на сервере базы данных ГОП. Здесь в электронном виде хранятся паспорта складов ЦПАШ, агломерационного цеха и РОФ, формируется база сертификатов на привозное сырье, осуществляется копирование (репликация) информации с серверов ЦПАШ, агломерационного цеха и РОФ по поступлению сырья и отгрузке продукции со складов, качеству подготовленной руды и полученного агломерата, производится комплексная оценка материальных и энергетических ресурсов, затраченных на подготовку шихтовых материалов;

- *база данных доменного производства*, расположенная на сервере базы данных в вычислительном центре доменного цеха. Здесь накапливается информация о работе каждой доменной печи.
- *база данных железнодорожного транспорта (ЖДТ)*, которая содержит оперативную информацию о передвижении по территории комбината железнодорожных вагонов с железорудным сырьем, коксом, чугуновозных ковшей, миксеров и др.

При разработке и сопровождении системы баз данных аглодоменного производства особое внимание уделяется надежности ее работы, достоверности вносимых данных и оперативности получения информации. Эти требования обеспечиваются соответствующим выбором и настройкой аппаратно-программных средств, а также повышением уровня квалификации пользователей системы. Рассмотрим подробнее характеристику аппаратно-программных средств для реализации системы баз данных на примере аглодоменного производства ОАО ММК.

### **6.4.3. Характеристика аппаратно-программных средств вычислительного центра доменного цеха ОАО ММК**

Аппаратно-технический уровень системы баз данных аглодоменного производства обеспечивается применением современных компьютеров для сбора и первичной обработки параметров технологических процессов на базе процессоров Intel Pentium 233 с оперативной памятью 32 Мбайт и емкостью жесткого диска 6,4 Гбайт, в качестве серверов – компьютеров Intel Pentium II с оперативной памятью 256 Мбайт и емкостью жесткого диска 6,4 Гбайт. В качестве физической передающей среды широко используются экранированная витая пара и волоконно-оптический кабель. Все компьютеры подключены к вычислительной сети аглодоменного цеха, которая для повышения ее общей производительности и надежности работы разбита на отдельные логические сегменты (подсети), т.е. является логически структуризованной. Взаимодействие между отдельными подсетями организуется с помощью коммутаторов и маршрутизаторов, установленных в сетевой стойке вычислительного центра (ВЦ) доменного цеха (**рис. 6.10**).

Объединение ЛВС отдельных производств и выход в единую корпоративную сеть комбината, как уже было отмечено, осуществляется по двум дублирующим друг друга каналам (**рис. 6.11**). В качестве базовой операционной системы для организации компьютерной сети доменного цеха ОАО ММК используется Windows NT. Как было отмечено раньше, управление пользователями в такой системе основано на концепции группового администрирования, когда права на вычислительные ресурсы назначаются не каждому пользователю в отдельности, а целой группе. Под вычислительными ресурсами в данном случае понимается вся совокупность аппаратных, программных и информационных ресурсов доменного цеха. Все компьютеры, подключенные к компьютерной сети цеха, образуют домен, управляются специальными компьютерами – серверами, расположенными на территории вычислительного центра (ВЦ) доменного цеха (**рис.6.10**). Отметим назначение и основные функции серверов ВЦ.

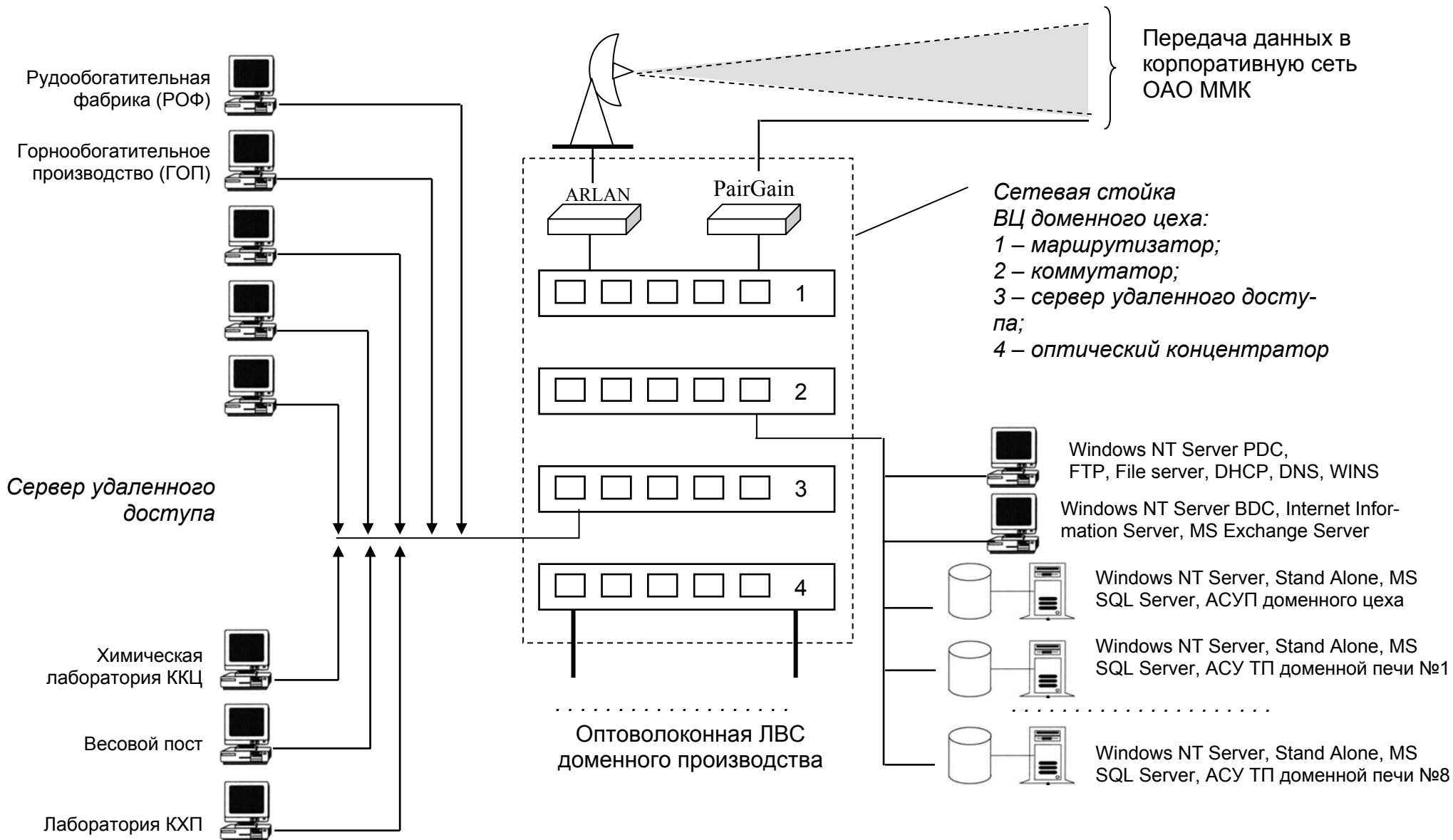


Рис. 6.10. Структура аппаратно-технического обеспечения вычислительного центра доменного цеха ОАО ММК

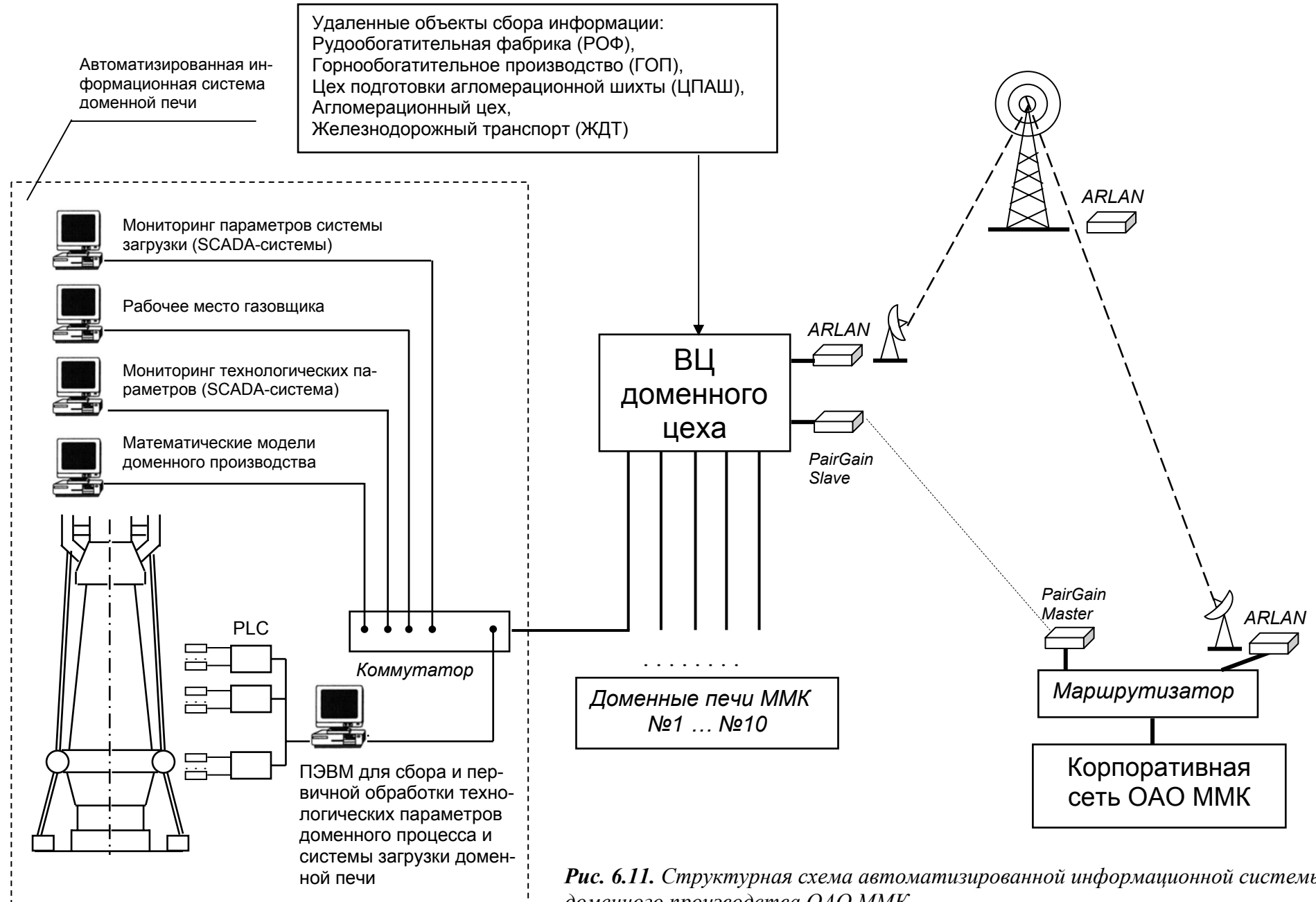


Рис. 6.11. Структурная схема автоматизированной информационной системы доменного производства ОАО ММК

*Главный контроллер домена (Primary Domain Controller, PDC)* работает под управлением операционной системы Windows NT Server v 4.0. Здесь хранится основная копия базы данных учетных записей (Security Accounts Management Database, SAM), включающей в себя все глобальные группы и учетные записи пользователей и компьютеров доменного цеха. Кроме того, данный сервер используется для аутентификации пользователей, когда они выполняют регистрацию в сети или осуществляют доступ к совместно используемым вычислительным ресурсам. На главном контроллере домена также располагаются программные инструментальные средства, используемые для централизованного управления работой сети и обеспечения взаимодействия пользователей сети доменного цеха с пользователями других цехов и служб комбината на основе Internet-технологии. В частности:

- *FTP-сервер* предоставляет услуги по доступу к файлам на удаленной машине в соответствии с протоколом передачи файлов (File Transfer Protocol, FTP). Этот протокол входит в семейство протоколов TCP/IP, предоставляет общий язык для упрощения переноса файлов между файловыми системами, имеющими различные форматы сохранения файлов. FTP-сервер функционирует как встроенная сетевая служба, что дает возможность использовать Windows NT Server для контроля доступа к каталогам FTP. Одним из главных преимуществ использования службы FTP является то, что пользователи имеют возможность перемещать и копировать файлы, не создавая управляющие Web-страницы и указатели на файлы;
- *файловый сервер (File server)* представляет собой многопользовательскую централизованную систему управления данными. В данном случае используется выделенный файловый сервер – специальный компьютер, который управляет внешним запоминающим устройством большой емкости и обеспечивает хранение общих файлов и доступ к ним из других узлов сети. Настройка прав доступа удаленных пользователей к файлам данных сервера осуществляется стандартными средствами Windows NT;
- *DHCP-сервер* служит для автоматизации процесса назначения и освобождения сетевых адресов (IP-адресов) рабочим станциям, благодаря чему администраторы сети освобождаются от работы по ручному назначению сетевых адресов и управлению ими. Механизм его функционирования основан на протоколе динамической настройки узла (Dynamic Host Configuration Protocol, DHCP). На сервере хранится таблица доступных и использованных IP-адресов, он обрабатывает запросы от рабочих станций, которым необходимо получить или обновить зарезервированный за ними IP-адрес;
- *WINS-сервер (Windows Internet Name Service, WINS)* – служба имен Internet Windows – используется для обеспечения динамического распознавания имен NetBIOS и преобразования их в IP-адреса, например, при использовании имени компьютера для поиска ресурсов;
- *DNS-сервер (Domain Name Service)* обеспечивает работу службы сервера доменных имен, который выполняет функции узла Internet и преобразует доменные имена в IP-адреса. Основным преимуществом службы DNS перед WINS является то, что DNS поддерживается практически всеми клиентами

протокола TCP/IP в отличие от службы WINS, которая была разработана специально для работы под операционными системами Microsoft.

*Резервный контроллер домена (Backup Domain Controller, BDC)* работает под управлением операционной системы Windows NT Server v 4.0 и выполняет те же функции, что и главный, но с одним существенным отличием: он не содержит основной копии базы данных учетных записей SAM. Вместо этого главный контроллер домена выполняет ее репликацию на все резервные контроллеры домена. Это означает, что при отсутствии в сети доменного цеха главного контроллера домена она вполне работоспособна, но нет возможности изменять учетные записи в домене (учетные записи пользователей и компьютеров). Основным назначением резервного контроллера домена является равномерное перераспределение нагрузки аутентификации пользователей сети. Кроме того, если главный контроллер домена будет недоступен по причине выхода из строя оборудования или вследствие запланированного отключения, например для замены каких-либо аппаратных компонентов, то выполнение его функций можно возложить на резервный контроллер домена. Это позволит продолжить выполнять аутентификацию пользователей и осуществлять управление сетью доменного цеха.

На резервном контроллере домена также располагаются:

- *информационный сервер Интернета (Internet Information Server, IIS)*, который предоставляет пользователям аглодоменной сети работу основных служб Интернета – WWW (World Wide Web), протокола передачи файлов FTP и Gopher;
- *почтовый сервер Microsoft Exchange* для организации и обслуживания работы электронной почты внутри сети аглодоменного производства;
- *выделенные серверы (Stand Alone)*, которые работают под управлением операционной системы Windows NT Server v 4.0. Назначением этих серверов является предоставление клиентам сети (конечным пользователям или прикладным программам) затребуемых информационных ресурсов, а именно информации о технологических параметрах работы доменных печей. С этой целью для каждой из восьми печей цеха установлен выделенный сервер, на котором функционирует система управления базами данных (СУБД) SQL Server v 6.5, осуществляющая накопление, хранение и выполняющая основную часть работы по обработке данных, поступающих с доменной печи. В SQL Server используется реляционная модель организации данных, построенная с учетом информационных потребностей доменного производства.

Такая информационная система, основу которой составляет система управления базами данных SQL Server, входит в состав АСУ ТП каждой доменной печи.

Отдельный выделенный сервер, работающий под управлением Windows NT Server v 4.0, осуществляет сбор и обработку первичной технологической информации с серверов баз данных каждой доменной печи с целью информационного обеспечения АСУП доменного цеха. В качестве программных средств системы баз данных аглодоменного производства используется программный продукт фирмы Microsoft SQL Server, который обеспечивает клиент/серверную обработку информации, поддерживает реляционную модель организации данных и организует доступ к базе данных средствами сетевой операционной системы

Windows NT. В качестве средств разработки клиентских частей прикладных программ применяются языки программирования Microsoft Visual C++, Microsoft Visual Basic. Поскольку весь обмен информацией между клиентом и сервером происходит посредством SQL-запросов, то значительным преимуществом такого подхода является снижение сетевого трафика, повышение скорости обработки запросов, а также интегрирование системы баз данных в корпоративную сеть комбината с использованием Intranet-технологии.

На каждой доменной печи установлены рабочие станции ПЭВМ, которые работают под управлением операционной системы Windows NT Workstation v 4.0 и осуществляют функции сбора и первичной обработки технологической информации доменного процесса (АСУ ТП) и системы загрузки доменной печи.

Таким образом, использование при разработке и реализации базы данных CASE-технологии проектирования информационных систем, современных аппаратно-программных средств функционирования и сопровождения системы баз данных позволило создать на ОАО ММК сквозную информационную систему доменного производства, обеспечивающую распределенную обработку и централизованный контроль и оперативное управление технологическим процессом на всех этапах – от складирования сырья в цехе подготовки агломерационной шихты до выплавки чугуна. Возможность вносить изменения непосредственно в логическую структуру базы данных и тем самым оперативно учитывать вновь возникающие информационные потребности пользователей придает построенной таким образом системе значительную гибкость и обеспечивает легкость при организации взаимодействия с другими информационными системами.

## 6.5. Контрольные вопросы

1. Почему при анализе работы доменной печи ее следует рассматривать как управляемую технологическую систему? Что в данном случае понимают под системой управления?
2. На какие группы подразделяются все входные и выходные параметры, характеризующие работу технологической системы? Приведите конкретные примеры из практики работы доменной печи.
3. Назовите основные принципы построения современной автоматизированной системы доменной плавки.
4. Перечислите основные подсистемы контроля технологических параметров доменной печи.
5. Какие задачи позволяет решать автоматизированная информационная система технического обслуживания и управления (АИСТОУ)? Назовите основные подсистемы АИСТОУ и перечислите функции, которые они выполняют.
6. Какие функции выполняет диспетчерская подсистема технологического персонала доменной печи?
7. Каковы особенности функционирования системы баз данных в аглодоменном производстве? Какие аппаратно-программные средства при этом используют?
8. Поясните структуру распределенной системы баз данных аглодоменного производства.



---

Глава 7. ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ	7-2
7.1. Понятие интеллектуальной системы	7-2
7.2. Архитектура современных экспертных систем	7-3
7.3. Классификация экспертных систем	7-7
7.4. Модели представления знаний	7-10
7.4.1. <i>Продукционная модель</i>	7-10
7.4.2. <i>Логические модели представления знаний</i>	7-14
7.4.3. <i>Представление знаний с использованием семантических сетей</i>	7-19
7.4.4. <i>Методы неточных рассуждений с ненадежными данными</i>	7-23
7.5. Инструментальные средства построения экспертных систем	7-26
7.5.1. <i>Традиционные языки программирования</i>	7-26
7.5.2. <i>Языки искусственного интеллекта</i>	7-26
7.5.3. <i>«Оболочки» и программные обстановки</i>	7-27
7.6. Контрольные вопросы	7-28

## Глава 7. ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

### 7.1. Понятие интеллектуальной системы

Интеллектуальные системы и технологии применяются для тиражирования профессионального опыта и решения сложных научных, производственных и экономических задач.

Под **интеллектуальной системой** понимается искусственно созданная система, обладающая способностью накопления и корректировки знаний на основе активного восприятия информации о мире и обобщенного опыта, а также целенаправленного поведения.

*Искусственный интеллект (ИИ)* – это программная система, имитирующая на компьютере мышление человека. Для создания такой системы необходимо изучить процесс мышления человека, решающего определенные задачи или принимающего решения в конкретной области.

Когда человеческий мозг приступает к решению даже самой простой задачи, для выбора нужных действий в его распоряжении имеется огромный объем информации. Например, если бы человек, прежде чем шагнуть на проезжую часть, анализировал все факты, имеющие прямое, косвенное или вообще не имеющие никакого отношения к его цели перейти дорогу, он простоял бы на обочине несколько лет.

Однако в мозгу существует сложная система, руководящая выбором правильной реакции на конкретную ситуацию. Такой выбор называется *упрощением*. Механизм упрощения блокирует мысли, не имеющие отношения к решаемой в данный момент задаче. Точно так же, как удаление у дерева ненужных веток помогает ему расти, механизм упрощения способствует достижению цели, игнорируя все бесполезные для этого факты. Одним словом, без упрощения человеческий мозг будет парализован, механизм же упрощения направляет мысль человека в нужное русло. Аналогичный подход используется и в искусственном интеллекте. Как мы с вами увидим в дальнейшем, большинство человеческих рассуждений может быть представлено с помощью очень простого правила вида

*Если (условие), то (выполнение действий).*

Однако даже современному компьютеру тяжело обработать за один раз несколько сот правил, человек же – эксперт обрабатывает за один раз около пятидесяти тысяч таких правил в своей предметной области. А если рассматривать экспертизу обо всем мире в целом, то потребуются миллионы и миллиарды правил. За счет чего же можно уместить в компьютере информацию о мире? Ответ очевиден – только за счет повышения уровня абстракции представления знаний о мире.

Для того чтобы заставить компьютер понимать знания, представленные в виде человеческой речи, необходимо построить анализатор естественного языка, с помощью которого, в основном, передаются знания. Основными функциями анализа языка являются:

- лексический анализ (анализ слов);

- синтаксический анализ (анализ порядка слов в предложении с учетом правил грамматики);
- семантический анализ (анализ значения предложения самого по себе и в его связи с другими предложениями).

С самого начала исследования в области ИИ пошли по двум направлениям.

*Первое (бионическое)* – попытка смоделировать с помощью искусственной системы психофизиологическую деятельность человеческого мозга с целью создания искусственного разума.

*Второе (прагматическое)* – создание программ, позволяющих с использованием компьютера воспроизводить не саму мыслительную деятельность, а являющиеся ее результатами процессы. Здесь достигнуты важные практические результаты, поэтому в дальнейшем речь будет идти именно об этом направлении.

В настоящее время ИИ – одна из бурно развивающихся областей науки, которая разрабатывает методы и специальные программно-технические средства поиска решений интеллектуальных творческих, или математически неформализованных задач (НФЗ). Неформализованные задачи носят невычислительный целенаправленно-смысловой характер.

Разнообразные НФЗ в различных сферах научно-технической деятельности обладают рядом особенностей.

Прежде всего, постановка (формулировка) НФЗ не может быть формально определена в числовом виде, а определяется лишь в смысловом представлении на ограниченном естественном языке. Для НФЗ характерна неполнота, ошибочность, неоднозначность, неопределенность или противоречивость как исходной информации (знаний и данных), так и используемых правил ее преобразования.

Каждая НФЗ имеет семантическое (смысловое) решение, представляющее собой словесное описание (или графическое изображение) некоторого явления, технологического процесса или технологической операции, принципиальной схемы системы или конструкции объекта.

## 7.2. Архитектура современных экспертных систем

Наибольшие практические результаты в области искусственного интеллекта связаны с созданием экспертных систем (ЭС), которые начинают использоваться в металлургии при решении практических задач.

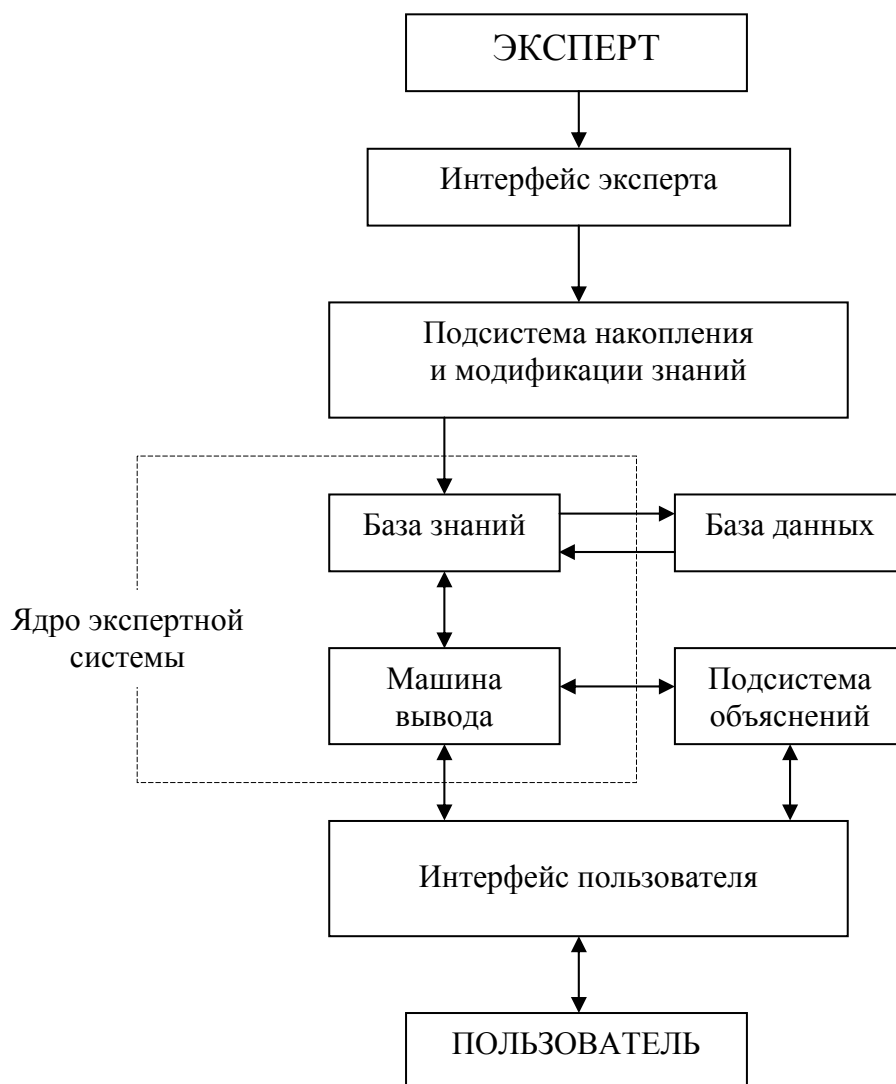
В настоящее время существует целый ряд определений ЭС различной степени общности и детализации. Дадим некоторые определения понятия ЭС.

**Экспертная система** – это сложные программные комплексы, аккумулирующие знания специалистов в конкретных предметных областях и тиражирующие этот эмпирический опыт для консультации менее квалифицированных пользователей. Иными словами, экспертная система – это компьютерная система, использующая знания специалистов в некоторой прикладной области и при-

нимающая решения (делающая заключения) в рамках этой области на уровне эксперта.

Большинство современных экспертных систем включает следующие базовые компоненты (**рис. 7.1**):

- интерфейс пользователя;
- база знаний;
- редактор базы знаний (подсистема накопления и модификации знаний);
- машина вывода (решатель, подсистема логического вывода);
- подсистема объяснений;
- интерфейс эксперта;
- база данных.



**Рис. 7.1.** Блок-схема типичной экспертной системы

В коллектив разработчиков ЭС входит как минимум четыре человека (групп людей):

- эксперт;

- инженер по знаниям (инженер базы знаний);
- программист;
- пользователь.

Определим эти термины.

*Эксперт* – это известный специалист в данной предметной области, который владеет разнообразными знаниями предметной области, знает огромное число источников знаний (книг, справочников и т.п.), а также за годы обучения, теоретической и экспериментальной работы в конкретной предметной области приобрел навыки и опыт высокоэффективного решения определенного класса задач, относящихся к предметной области.

*Инженер по знаниям* – это специалист по искусственному интеллекту, выступающий в роли промежуточного буфера между экспертом и базой знаний. Он возглавляет коллектив, это ключевая фигура при разработке систем, основанных на знаниях (на рис. 7.1 не показан).

*Программист* – специалист по разработке и отладке программного обеспечения, профессионально владеющий программными средствами реализации алгоритмов ЭС.

*Пользователь* – специалист предметной области, для которого предназначена система. Обычно его квалификация недостаточно высока, и поэтому он нуждается в помощи и поддержке своей деятельности со стороны ЭС.

Рассмотрим несколько детальнее каждую из указанных компонент и ее место в общей архитектуре экспертной системы, а также общий принцип работы пользователя с ЭС.

*Интерфейс эксперта* обеспечивает пользователю возможность вести дружественный диалог с системой, т.е. осуществлять проблемно-ориентированное общение между пользователем и компьютером. Хотя работа с ЭС не является простым делом, пользовательский интерфейс этих систем является дружественным и обычно не вызывает трудностей при ведении диалога.

*База знаний.* База знаний содержит два основных элемента: факты (данные) из предметной области и специальные эвристические правила, которые управляют использованием фактов при решении проблемы, т.е. содержимое БЗ формируется в соответствии с принятым способом (моделью) представления знаний.

*Редактор базы знаний* (подсистема накопления и модификации знаний) – это программа, предназначенная для ввода в базу знаний новых знаний о предметной области для представления их в базе знаний. При этом в режиме приобретения знаний (обучения системы путем ввода знаний в БЗ) инженер базы знаний работает в тесном контакте с экспертом из прикладной области, на которую ориентирована данная БЗ, а вместе с ней и ЭС. В процессе приобретения знаний (обучения) БЗ переводит знания эксперта с его профессионального языка на язык правил и стратегий. В отличие от БД, содержащих статические связи между полями записей, записями и файлами, БЗ находится в непрерывном динамическом обновлении, отражая рекомендации соответствующих экспертов. Действительно, с увеличением объема БЗ как основа принятия решений, так и сами решения могут изменяться.

*Машина вывода* – это «мозг» ЭС, т.е. программа, поддерживающая методологию обработки информации из базы знаний, получение и представление заключений и рекомендаций. Она моделирует ход рассуждений эксперта на основании знаний, имеющихся в БЗ. Синонимы машины вывода: решатель, дедуктивная машина, машина логического вывода.

База знаний и решатель вместе составляют основную часть ЭС – ее *ядро*. Иногда к ядру ЭС относят только базу знаний.

*Подсистема объяснений*. Особенностью ЭС является возможность получить не только решение, но и необходимые объяснения. Подсистема объяснений – программа, позволяющая пользователю получать ответ на вопросы: «Как была получена та или иная рекомендация?», «Почему система приняла такое решение?» и т.п. Подсистема объяснений предназначена для отображения в удобном для пользователя виде промежуточных и окончательных выводов и объяснения производимых системой действий. Различают два вида объяснений:

- *объяснения, выдаваемые по запросам*. Пользователь в любой момент может потребовать от ЭС объяснения своих действий;
- *объяснения решения проблемы*. После получения результата решения пользователь может потребовать объяснений того, как оно было получено. Система должна пояснить каждый шаг своих рассуждений, ведущих к решению задачи.

*Интерфейс эксперта*. Как правило, для общения эксперта используется диалоговый режим и графический интерфейс, позволяющий наглядно отображать ход принятия решений системой. Наличие такой компоненты предполагает, что система использует методы рассуждений, доступные объяснению и пониманию эксперту.

*База данных* – хранит всю информацию, поступающую в режиме реального времени, генерируемую техническими средствами контроля технологического процесса, а также вводимую вручную технологическим персоналом.

Общая схема взаимодействия пользователя с ЭС сводится к следующему. После создания конкретной ЭС в результате совместной работы инженера по знаниям, эксперта, программиста при главенствующей роли первого, взаимодействие пользователя с системой производится через интерфейс на некотором не-процедурном языке, близком к естественному или профессиональному языку предметной области, на которую ориентирована данная ЭС. В интерфейсной компоненте ЭС производится трансляция предложений этого языка во внутренний язык представления знаний системы.

Описание запроса поступает в подсистему логического вывода. Основу БЗ составляют формально представленные в ней факты и правила модели предметной области, которая на основе информации из БЗ генерирует рекомендации по решению поставленного вопроса.

В подсистеме логического вывода реализуется некоторая стратегия выбора соответствующего правила из БЗ. Посредством подсистемы объяснений производится отображение промежуточных и окончательных выводов и объяснение производимых системой процедур.

В процессе работы с ЭС пользователь в рамках интеллектуального интерфейса должен получить следующие основные возможности:

- решение поставленных перед системой задач в приемлемом для его понимания виде;
- поддерживать в актуальном состоянии БЗ системы путем ее модификации, в первую очередь, приобретением новых знаний;
- обеспечение различного рода справочной информации по системе, включая обучающие функции и демонстрационные примеры.

Процесс приобретения знаний состоит в передаче знаний и опыта от источника (эксперт, опытные факты, публикации, результаты расчетов с использованием математических моделей и т.п.) системе. Как правило, в качестве источника знаний выступают эксперты не в целом по предметной области, на которую ориентирована ЭС, а специалисты по ее узким направлениям.

Полученные от эксперта знания представляют собой набор фактов, правил, процедур и оценок (включая вероятностные) в его узкой профессиональной области. С целью упрощения процесса погружения знаний в БЗ организуется совместная работа инженера базы знаний, отвечающего за проектирование и разработку ЭС.

Эксперт, как правило, не владеет основами знаний по искусственному интеллекту, способам представления знаний в ЭС, тогда как инженер базы знаний имеет в этой области целенаправленную подготовку.

Заметим попутно, что процесс передачи знаний от эксперта посредством инженера базы знаний является на сегодня одним из наиболее узких мест проектирования экспертных систем.

Приведенная структура ЭС является весьма обобщенной и, конечно, не может отразить всех деталей ее структуры. Заметим, что в содержании и возможностях каждой из компонент существует большое разнообразие.

### 7.3. Классификация экспертных систем

Основные типы могут быть сгруппированы по способу использования данных и знаний, хранящихся в экспертной системе:

- ЭС, осуществляющие обработку данных на основе знаний;
- ЭС, осуществляющие формирование данных на основе знаний.

#### ***Обработка данных на основе знаний***

Этот тип экспертных систем решает задачи интерпретации, диагностики и мониторинга.

*Интерпретация данных.* Это одна из традиционных задач для экспертных систем. Экспертные системы, выполняющие интерпретацию, как правило, используют информацию от датчиков для описания ситуации. Интерпретирующие системы имеют дело непосредственно с реальными данными. Им приходится обрабатывать информацию зашумленную, недостаточную, неполную, ненадежную или даже ошибочную. Интерпретирующие системы объясняют наблюдаемые данные, приписывая им символичный тип, описывающий ситуацию или состояние системы.

Примером экспертных систем этого класса является система PROSPEKTOR, которая выступает в качестве опытного консультанта по обнаружению полезных ископаемых на основе геологических данных.

**Диагностика.** Под диагностикой понимается обнаружение неисправностей в некоторой системе. Системы диагностики судят о нарушениях в работе системы по наблюдениям. Обычно системы диагностики соотносят наблюдаемые нарушения поведения систем с обуславливающими их причинами. Диагностирующие системы часто являются консультантами, поскольку не только ставят диагноз, но и помогают в отладке. Они могут взаимодействовать с пользователем, чтобы оказать помощь при поиске неисправностей, а затем предложить порядок действий по их устранению.

Экспертные системы диагностики представляют собой самый многочисленный и реализованных класс экспертных систем. Наиболее известной среди них является система MYCIN – диагностика инфекционных заболеваний крови, ANGY – система диагностики и терапии сужения коронарных сосудов.

**Мониторинг.** Основная задача мониторинга – непрерывная интерпретация данных в реальном масштабе времени и сигнализация о выходе тех или иных параметров за допустимые пределы. Главная проблема при их создании – «пропуск» тревожной ситуации и инверсная задача «ложного» срабатывания. Примером такой системы является контроль аварийных ситуаций на химическом заводе – ЭС FALCON.

#### **Формирование данных на основе знаний**

Этот тип экспертных систем решает задачи *обучения, планирования, проектирования, прогнозирования и управления.*

**Обучение.** Системы обучения диагностируют и отлаживают поведение обучаемого с помощью компьютера и подсказывают правильные решения. Как правило, работа таких систем начинается с создания гипотетического описания знаний, например студента, позволяющих интерпретировать его поведение. Они диагностируют проблемы в знаниях обучаемого и подбирают соответствующие средства для их устранения. Примером такого типа систем является система SOPHI, обучающая студентов знаниям электроники, система PROUST – обучение языку Паскаль и др.

**Планирование.** Под планированием понимается нахождение планов действий, относящихся к объектам, способным выполнять некоторые функции. В таких ЭС используются модели поведения реальных объектов с тем, чтобы логически вывести последствия планируемой деятельности. Экспертные системы, занятые планированием, проектируют действия. Они определяют полную последовательность действий прежде, чем начинается их выполнение. Планирующие экспертные системы зачастую должны иметь способность к возврату, т.е. отвергать некоторую последовательность рассуждений или часть плана из-за нарушения ограничений задачи. Примером этого класса систем может служить система ISIS, строящая план выполнения промышленного заказа. Система выбирает последовательность операций, необходимых для выполнения заказа, определяет сроки его выполнения и выделяет ресурсы для каждой операции.



Проектирование. Основная задача систем проектирования состоит в разработке подробной документации, предназначенной для создания объектов, которые удовлетворяют определенным требованиям. Требования к задачам проектирования во многом совпадают с требованиями для задач планирования. Основные проблемы здесь – это получение четкого структурного описания знаний об объекте. Одной из известных систем проектирования является система XCON, проектирующая конфигурацию вычислительной системы по заявкам пользователей, проектирование БИС – система CADHELP.

Прогнозирование. Экспертные системы, осуществляющие прогноз, определяют вероятные последствия заданных ситуаций. Системы прогнозирования используют имитационное моделирование, т.е. программы, которые отражают причинно-следственные взаимосвязи в реальном мире, чтобы сгенерировать ситуации, которые могут возникнуть при тех или иных подходах. Примером этого класса систем может служить система PLANT, предсказывающая потери зерновых культур, предсказания погоды – система WILLARD.

Управление. Экспертная система, решающая задачу управления, обеспечивает адаптивное управление всем поведением некоторой системы. Управляющая система должна постоянно интерпретировать текущую ситуацию, прогнозировать будущее, диагностировать причины возникающих проблем, формулировать план их ликвидации и контролировать его выполнение. Так, система ECESIS обеспечивает автономное управление системы жизнеобеспечения и применяется на борту обитаемой космической станции.

В последние годы выделяют два больших класса ЭС, существенно отличающихся по методологии и средствам их проектирования, условно называемые простыми и сложными.

Создание ЭС – это трудоемкая операция, которая может занимать до 20 человеко-лет.

Простая ЭС может быть охарактеризована как традиционная, выполненная на персональном компьютере. Время разработки от 3 месяцев до одного года, БЗ содержит от 200 до 1000 правил.

Сложная ЭС может быть охарактеризована как гибридная, выполненная на специальной символьной ЭВМ (типа Пролог-машины) либо мощной универсальной ЭВМ. Время разработки от 1 до 5 лет; БЗ содержит от 1500 до 10 тыс. правил.

*Выделяют следующие стадии существования ЭС, характеризующие степень ее проработки и отлаженности.*

Демонстрационный прототип – это ЭС, которая решает часть требуемых неформализованных задач, демонстрируя жизнеспособность методов теории искусственного интеллекта. Демонстрационный прототип имеет в БЗ порядка 50–100 правил. Развитие демонстрационного прототипа приводит к исследовательскому прототипу.

Исследовательский прототип – то ЭС, которая решает все требуемые задачи, но не устойчива в работе и не полностью проверена. На доведение ЭС до стадии исследовательского прототипа уходит примерно 1–2 года. Исследовательский прототип обычно имеет в БЗ 200–500 правил.

*Действующий прототип* – надежно решает все задачи, но для решения сложных задач может требоваться чрезмерно много времени и больше памяти компьютера. Доведение ЭС до стадии действующего прототипа осуществляется примерно за 2–3 года, при этом число правил возрастает до 500–1000.

*Промышленная ЭС* – обеспечивает высокое качество решений всех задач при минимальных затратах времени и памяти компьютера. Обычно процесс преобразования действующего прототипа в промышленную систему состоит в расширении знаний до 1500 правил. Доведение ЭС до стадии промышленной системы требует примерно 2–4 года.

Обобщение класса задач, решаемых ЭС на стадии промышленной ЭС, позволяет перейти к стадии *коммерческой системы*, т.е. к системе, пригодной не только для собственного использования, но и для продажи различным потребителям. Доведение ЭС до стадии коммерческой системы требует примерно 3–6 лет, при этом база знаний системы увеличивается до 1000–3000 правил.

## 7.4. Модели представления знаний

Основным отличительным признаком систем искусственного интеллекта является работа со знаниями. Представление знаний выливается в существенную проблему, связанную со многими вопросами – как получать, формализовать, моделировать знания, как их хранить и пополнять в системе, как их использовать?

Для ответов на эти вопросы рассмотрим наиболее распространенные модели представления знаний.

### 7.4.1. Продукционная модель

Продукционная модель чаще всего применяется в экспертных системах металлургии, поэтому остановимся на ней более подробно. Она привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода.

Продукционная модель (модель представления знаний с использованием правил) позволяет представить знания в виде приложений типа

*Если (условие), то (действие).*

Под *условием* понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под *действием* – действия, выполняемые при успешном исходе поиска. Иными словами, в ЭС, основанных на правилах, предметные знания представляются набором правил, которые проверяются на группе фактов или знаний о текущей ситуации. *Когда часть правила Если удовлетворяется фактом, то действие, указанное в части то, выполняется. Когда это происходит, то говорят, что правило выполнено.*

Например

*Если* верхний перепад давления в доменной печи растет, *то* возрастает газодинамическое сопротивление слоя в верхней части шахты.

*Если* температура плавления вещества меньше текущей температуры этого вещества, *то* фазовое состояние вещества – жидкость.

Действия правила могут состоять в модификации *набора* фактов в базе знаний.

Приведем небольшой пример из технологии доменной плавки.

*Если* диаграмма, фиксирующая расход дутья, подаваемого в доменную печь, представляет собой ровную линию с небольшими «пиками» в сторону увеличения расхода в момент перевода воздухонагревателей в различные режимы работы, и *если* диаграммы, фиксирующие давление горячего и холодного дутья, имеют вид ровных маломеняющихся линий (с пульсациями около среднего значения в пределах 5 кПа) с периодическими «пиками» в сторону снижения давлений в момент перевода воздухонагревателей, и *если* давление колошниковога газа имеет также вид маломеняющейся линии с небольшими и ровными по величине «пиками» в сторону снижения давления, согласующимися с опусканием очередной подачи в печь, и *если* диаграмма температуры колошниковога газа по отдельным газоотводам имеет вид ленты с шириной 50–60 °С, а абсолютное среднее значение при выплавке передельного чугуна при загрузке в печь холодных железорудных материалов составляет 250–310 °С, и *если* на диаграмме температур периферийных газов ширина ленты разброса показаний отдельных термодпар составляет 100–150 °С при среднем значении температуры газов 700–850 °С, и *если* содержание диоксида углерода по диаметру колошника отвечает оптимальному, и *если* содержание кремния и серы отвечает нормам ГОСТа и техническим условиям, и *если* различие в массе чугуна в соседних выпусках составляет 20–30 тонн (для печей среднего и большого объема), и *если* на зондовых (шомпольных) диаграммах отсутствуют горизонтальные площадки и обрывы шихты, *то* наблюдается нормальная работа доменной печи.

Программа, управляющая перебором правил, как раз и находится в *подсистеме логического вывода (машине вывода)*.

Существуют два важных способа использования правил в системе, основанной на правилах:

- прямая цепочка рассуждений;
- обратная цепочка рассуждений.

### *Прямая цепочка рассуждений (от данных к поиску цели)*

Данные – это исходные факты, на основании которых запускается машина вывода, т.е. программа, перебирающая правила из базы.

Рассмотрим пример применения прямого и обратного способов поиска для решения, постановка которого определяется четырьмя фактами: **M**, **N**, **C**, **D**, а также тремя правилами:

1-е правило –  $F \vee B \rightarrow Z$ ,

где  $\vee$  – дизъюнкция, операция, соответствующая употреблению союза «или».

Эта запись означает, что если существует ситуация **F** или ситуация **B**, то существует также ситуация **Z**.

2-е правило –  $C \vee D \rightarrow F$ .

3-е правило –  $M \vee N \rightarrow B$ .

Известный набор фактов ( $M, N, C, D$ ) мы будем называть базой данных. В результате решения необходимо доказать, что факт  $Z$  существует (является истинным). Все исходные данные находятся в базе данных. На **рис. 7.2** приведена блок-схема операций прямого способа вывода, а на **рис. 7.3** – дерево вывода для прямой стратегии поиска.

Давайте посмотрим теперь, как эти правила работают. Мы допустим, что каждый раз, когда набор правил проверяется относительно данных, только первое (самое верхнее) правило, согласующееся с данными, выполняется, поэтому на **рис. 7.2** правило  $M \vee N \rightarrow B$  выполняется лишь один раз.

Первое правило, которое выполняется, – это  $M \vee N \rightarrow B$ , поскольку  $M$  и  $N$  уже находятся в базе знаний. И как следствие этого правила выводится факт существования  $B$ , и  $B$  помещается в базу знаний. Это приводит к выполнению второго правила  $C \vee D \rightarrow F$ , и как следствие этого  $F$  выводится и помещается в базу знаний. Это, в свою очередь, вызывает выполнение третьего правила  $F \vee B \rightarrow Z$ .

Этот метод называется прямой цепочкой рассуждений, так как поиск новой информации происходит в направлении стрелок, разделяющих левые и правые части правил. Система использует информацию из левых частей, чтобы вывести информацию, содержащуюся в правых частях.

Описанная последовательность рассуждений называется прямой цепочкой потому, что констатирующая часть правила (часть *то*) выполняется только в том случае, если удовлетворяется условная часть правила (часть *Если*). Отправной точкой рассуждений, таким образом, служит уже возникшая ситуация, а затем делаются выводы. Что же должна делать программа, реализующая прямую цепочку рассуждений? Программа должна отвечать на вопросы пользователя, используя принцип прямой цепочки рассуждений и базу знаний.

Однако у реально существующих экспертных систем было бы не три правила, а сотни или даже тысячи. Если использовать столь большую систему, чтобы найти информацию, связанную с  $Z$ , то нужно будет исполнять много правил, не имеющих никакого отношения к  $Z$ . Было бы получено большое число цепочек вывода и ситуаций, справедливых, но с  $Z$  никак не связанных. Поэтому если цель состоит в том, чтобы установить один частный факт, а именно ситуации  $Z$ , то прямая цепочка рассуждений может оказаться напрасной тратой времени и денег.

В таких ситуациях обратная цепочка рассуждений (от цели к данным) окажется более эффективной.

### *Обратная цепочка рассуждений*

При этом методе вывода система начинает с того, что ситуация  $Z$  существует, и требуется определить те правила, которые привели к такой ситуации.

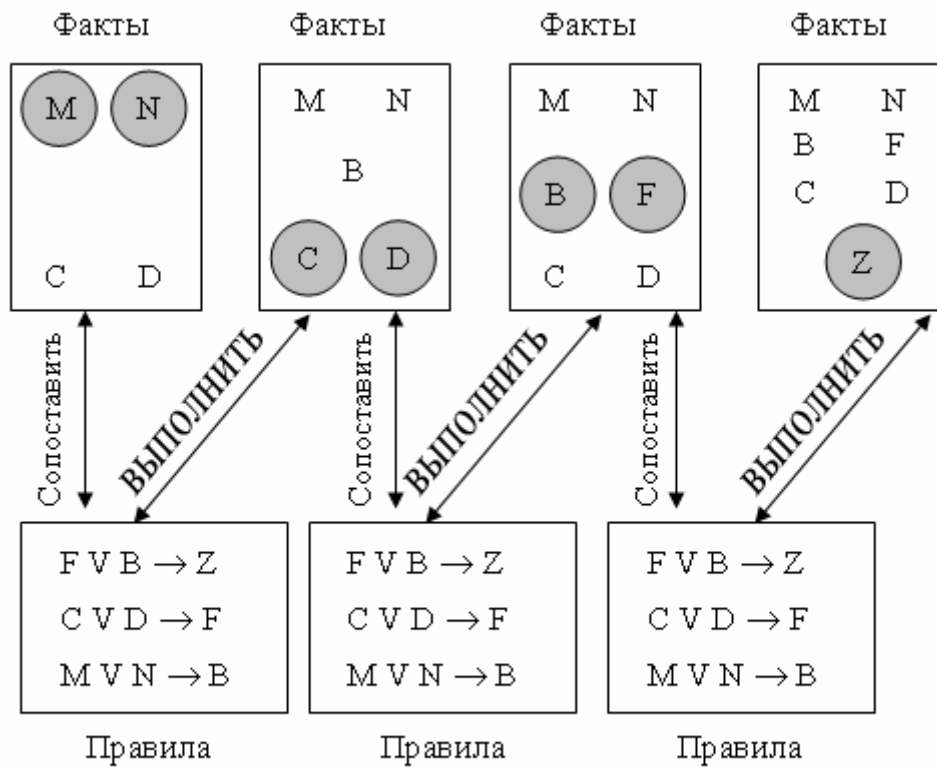


Рис. 7.2. Блок-схема операции прямого способа вывода

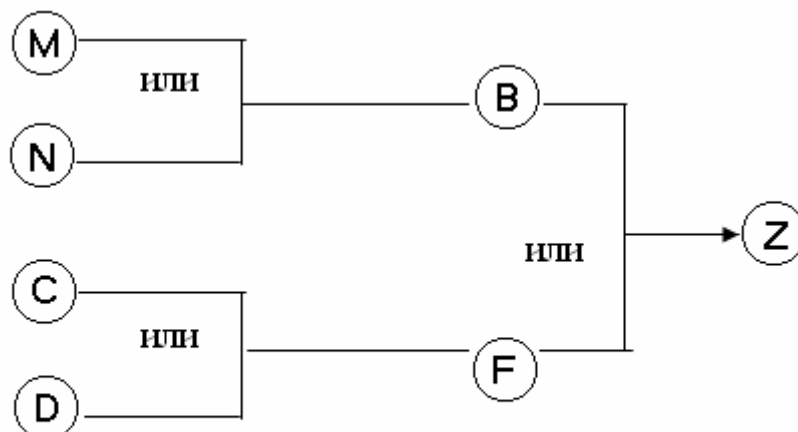


Рис. 7.3. Дерево вывода по прямой цепочке рассуждений

Предположим, что не удалось получить заданную марку стали в кислородном конвертере. В чем дело – неправильно был определен расход кислорода или неправильно выбран расход флюсов? Рассмотрим ту же задачу, что и при прямой цепочке рассуждений. По известному результату необходимо определить условия, которые к нему привели, т.е. по симптомам найти причины. Она отличается от предыдущей задачи тем, что там уже были известны условия ( $M, N, C, D$ ), но последствия  $Z$ , к которым они приведут, известны не были.

На **рис. 7.4** показано, как обратная цепочка рассуждений должна работать, используя правила из примера прямой цепочки рассуждений.

На первом шаге системе говорится, чтобы она установила (если может), что ситуация  $Z$  существует. Сначала она проверит базу в поисках  $Z$ , и в случае неудачи, будет искать среди правил то, которое приводит к установлению факта  $Z$ , т.е. правило, у которого  $Z$  стоит справа от стрелки. Она находит правило  $F \vee B \rightarrow Z$  и решает, что должна установить факты  $F$  и  $B$ , чтобы вывести  $Z$ .

На втором шаге система пытается установить факт  $F$ , сначала проверяя базу данных, а затем ищет правило, в правой части которого стоит  $F$ . Из этого правила  $C \vee D \rightarrow F$  система решает, что должна установить существование фактов  $C$  и  $D$  для получения заключения о  $F$ .

На шагах с третьего по пятый система находит  $C$  и  $D$  в базе и устанавливает факт существования  $F$ , но решает, что она должна установить факт  $B$ , прежде чем она получит заключение о  $Z$ .

На шагах с шестого по девятый система находит  $M$  и  $N$  в базе и устанавливает факт существования  $B$  и, наконец, на десятом шаге выполняет первое правило, чтобы установить основную цель – факт существования  $Z$ .

Цепочка выводов, созданная здесь, идентична той, что была создана в результате прямой цепочки рассуждений. Отличие этих подходов заключается в способе поиска правила и данных, здесь *идет поиск и среди правил, и среди данных*.

#### 7.4.2. Логические модели представления знаний

Логические модели представления знаний отображают знания данной предметной области в виде совокупности простых фактов, утверждений и суждений.

Факты отображаются в виде базовых элементов некоторой формальной системы. Утверждения и суждения отображаются в виде формул, которые состояются из базовых элементов с использованием синтаксических и семантических правил.

Любая формальная логическая система, отображающая знания, задается совокупностью множеств:

$$L = \langle T, S, A, P \rangle,$$

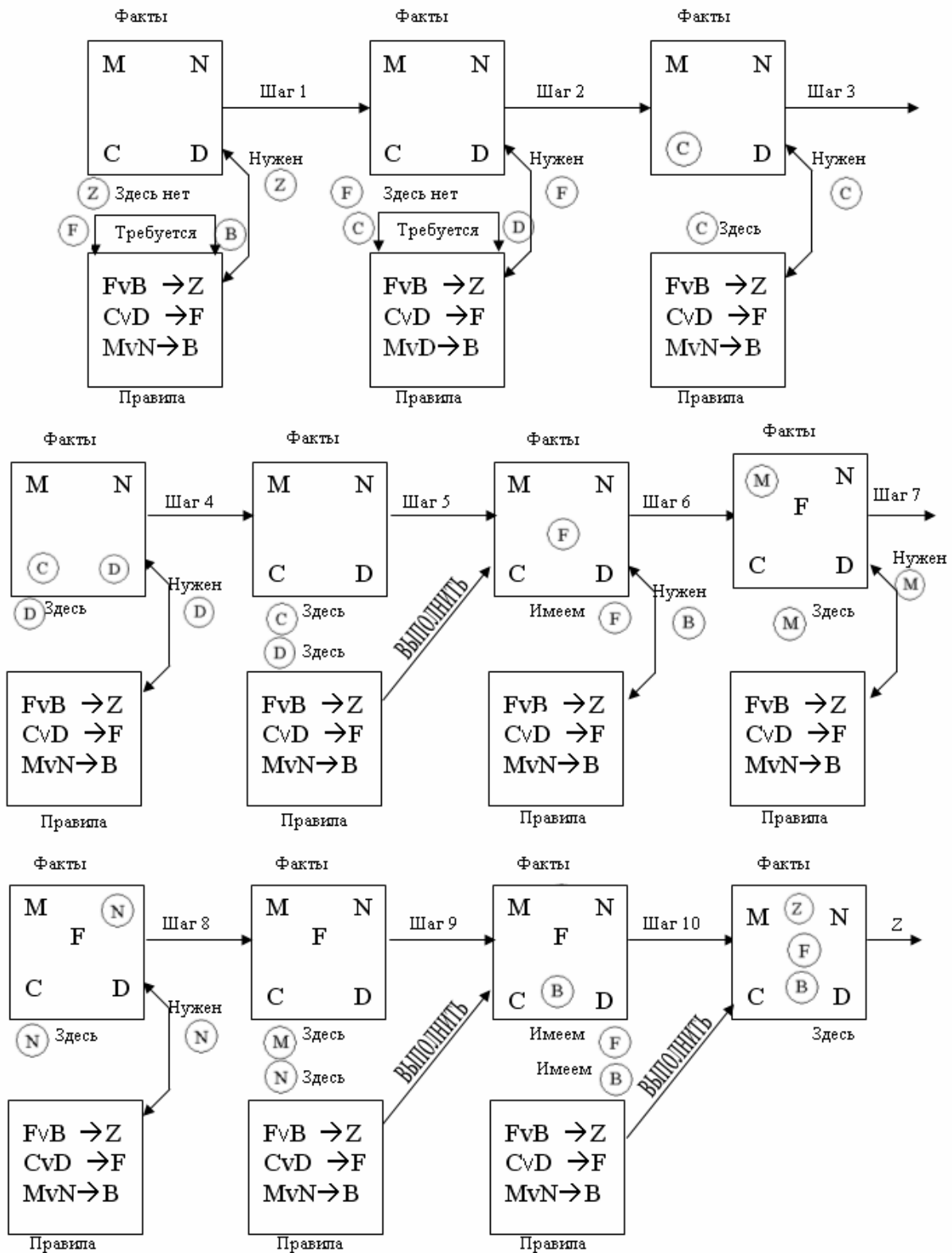


Рис 7.4. Блок-схема операции обратного способа вывода

где  $T$  – множество базовых элементов (символов, букв, слов);

$S$  – множество синтаксических правил, позволяющих строить из  $T$  синтаксически правильные выражения или формулы;

$A$  – множество априорно истинных выражений, называемых аксиомами;

$P$  – семантические правила, или правила вывода, позволяющие расширять множество аксиом  $A$  за счет новых формул, которые также являются истинными в данной формальной системе.

Логические модели представления знаний предполагают, что область или задача описывается в виде набора аксиом.

Для разработки логических моделей представления знаний используют исчисление высказываний и исчисление предикатов.

### *Исчисление высказываний*

Исчисление высказываний рассматривает высказывания, или утверждения, которые могут быть *либо истинными, либо ложными*. Например, высказывание «Увеличение температуры греющей среды повышает плотность теплового потока на поверхность нагреваемого тела – научный факт, который всегда является истинным». Каждое *простое высказывание* можно записать в следующей символической форме, или нотации:

$A$  = «насос увеличивает давление жидкости»;  $B$  = «теплообменник применяют для передачи теплоты» (где  $A$  и  $B$  – истинные простые высказывания, или атомы).

Для представления *сложных высказываний, или формул*, необходимо простые высказывания объединить с использованием логических связей.

С этой целью по аналогии с образованием составных высказываний в речи в логике высказываний вводятся следующие операции над высказываниями.

**Отрицание** – унарная операция, соответствует употреблению частицы «не», для обозначения которой используют знак

$\neg$ , not, (не).

**Конъюнкция** – бинарная операция, соответствующая употреблению союза «и».

$\wedge$ , &, and (и).

Условие истинности для конъюнкции  $A \wedge B$  заключается в том, что  $A$  истинно и  $B$  истинно.

**Дизъюнкция** – бинарная операция, соответствующая употреблению союза «или».

$\vee$ , or (или).

Условие истинности для дизъюнкции  $A \vee B$  заключается в том, что  $A$  истинно или  $B$  истинно.

**Импликация** – бинарная операция, близкая по смыслу к союзу «если...то ...».

$\rightarrow$  (если...то...).

Условие истинности для импликации  $A \rightarrow B$  заключается в том, что если  $A$  истинно, то  $B$  истинно. Высказывание  $A$  в импликации  $A \rightarrow B$  читается « $A$  им-



плицирует В» или «из А следует В». Высказывание  $A \rightarrow B$  ложно только в одном случае, когда А истинно, а В ложно.

**Эквивалентность** – бинарная операция, обозначается знаком  $\leftrightarrow$  (тогда, когда).

Высказывание  $A \leftrightarrow B$  истинно только тогда, когда А и В оба истинны либо оба ложны.

**Скобки ()**.

Рассмотрим запись в символической форме сложного высказывания:

$E =$  « В технологическую схему входят нагревательные печи и прокатный стан». Имеются два простых истинных высказывания:  $C =$  «В технологическую схему входят нагревательные печи»;  $D =$  «В технологическую схему входит прокатный стан».

Сложное высказывание  $E$  представляет собой конъюнкцию, т.е.

$E \leftrightarrow (C \wedge D)$ .

Сложное высказывание  $H =$  «Жидкую сталь можно разлить как в изложницы, так и на машине непрерывного литья заготовок» представляет собой дизъюнкцию двух простых высказываний:  $M =$  «Жидкую сталь можно разлить в изложницы»,  $P =$  « Жидкую сталь можно разлить на машине непрерывного литья заготовок», т.е.

$H \leftrightarrow M \vee P$ .

В сложном высказывании  $M =$  «Если в полупродукте содержатся железосодержащие частицы, то их можно отделить от пустой породы методом магнитного обогащения» выделим три простых высказывания:

$P =$  «полупродукт содержит железосодержащие элементы»;

$Q =$  «полупродукт поступает на магнитное обогащение»;

$N =$  «железосодержащие элементы отделяются от пустой породы».

Тогда сложное высказывание  $M$  можно представить как импликацию: «Если (полупродукт содержит железосодержащие частицы) и (пустую породу), то (железосодержащие частицы отделяются от пустой породы)»:

$M \leftrightarrow [(P \wedge Q) \rightarrow N]$ .

### Исчисление предикатов

Исчисление предикатов является расширением классического исчисления высказываний, в котором каждое высказывание рассматривается как единое целое, не обладающее внутренней структурой, а истинность и ложность формул и атомов фиксированы.

В исчислении предикатов основным элементом является *предикат* – *переменное высказывание*, то есть логическая функция

$P(x_1, x_2, \dots, x_n)$ ,

истинность или ложность которого зависит от значения входящих в предикат *предметных переменных*, или *аргументов*  $x_1, x_2, \dots, x_n$ .

Обратим ваше внимание на то, что предикат  $P$  принимает значение “истина” или “ложь в зависимости от того, какие значения принимают входящие в него переменные  $x_1, x_2, \dots, x_n$ ».

Порядок следования аргументов должен всегда задаваться в соответствии с интерпретацией предиката, принятой для моделирования предметной области при разработке информационной системы, т.е. разработчик должен принять решение о фиксированном, приемлемом для интерпретации порядке и неукоснительно соблюдать его в системе при работе с такими данными.

Например, факт “В доменную печь вдувается горячее дутье” можно представить следующим образом:

*Вдувание (доменная печь, дутье).*

Причем теперь в системе все предикаты такого типа должны в списке аргументов на первом месте содержать тип печи, а на втором месте – вид вдуваемого вещества.

*Вдувание (мартеновская печь, кислород).*

*Вдувание (методическая печь, природный газ) и т.п.*

Высказывания, состоящие из предиката и связанных с ним аргументов, называются **элементарными**. Элементарные высказывания могут объединяться в **сложные высказывания** с помощью логических связей.

Действительно, вследствие того, что предикаты являются двоичными (логическими) переменными, над ними можно производить логические операции. Рассмотрим выражение

$$(P_1(x_1, x_2) \wedge P_2(x_1, x_2)) \vee (P_2(x_2, x_3) \wedge \neg P_3(x_2, x_3)).$$

Его можно рассматривать как составную формулу алгебры логики, которая является функцией четырех логических переменных  $P_1(x_1, x_2)$ ,  $P_2(x_1, x_2)$ ,  $P_2(x_2, x_3)$ ,  $P_3(x_2, x_3)$ . Заметим, что  $P_2(x_1, x_2)$ ,  $P_2(x_2, x_3)$  зависят от разных переменных и, следовательно, определяют разные логические функции.

Это же выражение можно рассматривать и как составной трехместный предикат  $P_4(x_1, x_2, x_3)$ . Ведь в зависимости от значений предметных переменных  $x_1, x_2, x_3$  предикаты, входящие в выражение, будут принимать значения «истина» или «ложь», а следовательно, и значение этого выражения будет зависеть от значений этих трех переменных.

Для того чтобы в исчислении предикатов можно было манипулировать переменными, введены дополнительные структуры – **кванторы**.

*Кванторы служат для указания меры, в какой экземпляры переменных должны быть истинны, чтобы высказывание в целом тоже было истинным.*

Кванторы употребляют только с предметными переменными.

$\forall$  – квантор общности («для всех»);

$\exists$  – квантор единственности («существует такой»).

Например, пользуясь кванторами, предложение “все студенты-дипломники знают теорию горения” можно представить следующим образом:

$$(\forall X) (\text{студенты-дипломники}(X) \rightarrow \text{теория горения}(X)),$$

а предложение “некоторые студенты-дипломники будут работать по специальности” можно представить следующим образом:

$$(\exists X) (\text{студенты-дипломники}(X) \rightarrow \text{работать по специальности}(X)).$$

Пользуясь кванторами, предложение “все пламенные печи имеют топливосжигающие устройства” можно представить следующим образом:

$$(\forall X) (\text{пламенные печи}(X) \rightarrow \text{топливосжигающие устройства}(X)),$$

а предложение “в некоторые доменные печи вдувается пылеугольное топливо” в виде

$(\exists X) (\text{Доменные печи } (X) \rightarrow \text{пылеугольное топливо } (X)).$

Кванторы общности и кванторы существования могут перемежаться в пределах одного и того же выражения. В этом случае порядок, в соответствии с которыми вводятся квантифицируемые переменные, может влиять на смысл высказывания. Например, утверждение

$(\in X) (\exists Y) (\text{Служащий } (X) \rightarrow \text{Руководит } (Y, X))$

может быть интерпретировано как “каждым служащим  $X$  руководит некоторое лицо  $Y$ ”.

Логическое выражение

$(\exists Y) (\in X) (\text{Служащий } (X) \rightarrow \text{Руководит } (Y, X))$

соответствует утверждению “есть такое лицо  $Y$ , которое руководит всеми  $X$ ”.

Для закрепления материала приведем примеры с использованием исчисления высказываний и исчисления предикатов. Применяя исчисления высказываний, запишем некоторые факты, представленные фразами предметной области.

«Реакция экзотермическая»: *(Реакция – экзотермическая).*

«Реактор адиабатический»: *(Реактор – адиабатический).*

«Реактор изотермический»: *(Реактор – изотермический).*

«Реакция не экзотермическая»: *(Реакция –  $\neg$  экзотермическая).*

«Реакция экзотермическая и реактор адиабатический»:

*(Реакция экзотермическая)  $\wedge$  (Реактор адиабатический).*

Однако исчисление высказываний обладает рядом ограничений для выражений сложных утверждений. Например, используя только исчисление высказываний, необходимо написать такие формулы и фразы естественного языка:

Реакция – А -экзотермическая: *«Реакция – А –экзотермическая»;*

Реакция – В – экзотермическая: *«Реакция – В – экзотермическая»* и т.д. для каждой экзотермической реакции.

Вместо этого, используя исчисление предикатов, можно написать следующие выражения:

*Экзотермическая* ( $x$ ), где  $x$  – предметная переменная, имеющая значения: *Реакция – А, Реакция – В* и т.д.

Таким образом, можно представить все экзотермические реакции наиболее простой моделью.

Простой факт, что “Для каждой экзотермической реакции имеет место любой адиабатический реактор, реактор не может быть изотермическим” можно представить в виде формальной записи исчисления предикатов:

$(\in X) (\text{Экзотермическая } (X) \wedge (\exists Y) (\text{Адиабатический } (Y)) \rightarrow \neg (\text{Изотермический } (Y)).$

### 7.4.3. Представление знаний с использованием семантических сетей

Термин *семантическая* означает смысловая, а сама семантика – это наука, устанавливающая отношения между символами и объектами, которые они обозначают, т.е. наука, определяющая смысл знаков. Термин *семантическая сеть* применяется для описания метода представления знаний, основанного на сетевой структуре.

Семантические сети были первоначально разработаны для использования их в качестве психологической модели человеческой памяти. Теперь это стандартный метод представления знаний в экспертных системах.

*Основная идея способов представления знаний на основе аппарата семантических сетей состоит в рассмотрении проблемной области (а точнее, ее модели) как совокупности объектов и отношений между ними.*

Объекты при таком подходе представляются именованными вершинами, а связи – направленными именованными дугами некоторой семантической сети. Тогда система знаний представляется некоторой семантической сетью (ориентированным графом, образованным именованными вершинами и дугами или их совокупностью).

В качестве вершин семантической сети выступают только те объекты проблемной области, которые необходимы для решения поставленных задач. В качестве таких объектов могут выступать понятия, события, процессы и т.д.

Знания можно описывать отношениями между абстрагированными понятиями и сущностями, являющимися конкретными объектами реального мира. Понятия и отношения между ними можно описать сетью, состоящей из узлов и дуг. Узлы в такой сети выражают сущности и понятия, а дуги могут быть снабжены метками, которые показывают, что именно они описывают.

Таким образом, семантическая сеть – это ориентированный граф, вершины которого – понятия, а дуги – отношения между ними.

Проблема поиска решения на базе знаний типа семантической сети сводится к задаче поиска фрагмента сети в виде некоторой подсети, соответствующей поставленному вопросу.

Проиллюстрируем использование семантической сети в качестве процедуры поиска решений неформализованных задач.

Общая методика решения неформализованных задач основана на использовании принципа многоуровневой декомпозиции исходной задачи.

Сущность принципа многоуровневой декомпозиции при поиске решения состоит в том, что решение исходной задачи –  $P_0$  сводится к поиску решений образующих ее задач меньшей размерности –  $P_1, P_2, \dots, P_n$ , в целом представляющих всю задачу. Процедуру декомпозиции повторяют для каждой подзадачи до тех пор, пока каждая из полученного набора подзадач, образующих решение исходной задачи, не будет иметь очевидное, или тривиальное решение.

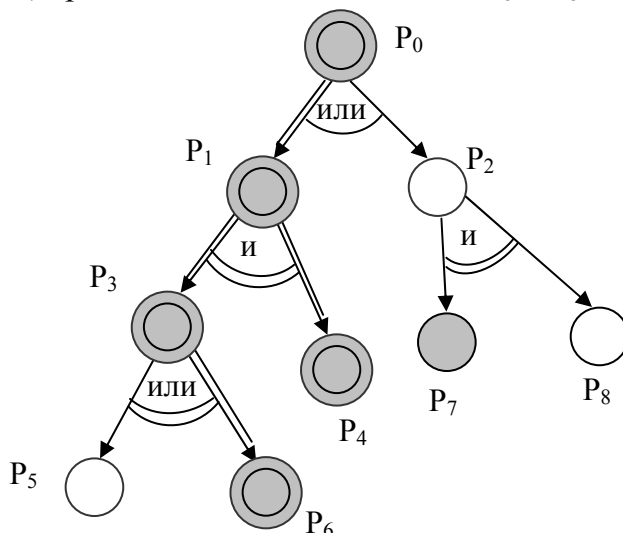
Процедуру поиска исходной неформализованной задачи декомпозицией ее на подзадачи можно представить в виде семантического графа – дерева декомпозиции исходной задачи, называемого «И/ИЛИ» («AND/OR») графом. Каждой вершине этого графа ставится в соответствие описание некоторой задачи (подзадачи).

В графе выделяют два типа вершин:

- вершины типа «И (AND)»;
- вершины типа «ИЛИ (OR)».

Для «И»-вершины решение задачи сводится к решению *всех подзадач*, соответствующих ее вершинам-потомкам. Для «ИЛИ»-вершины решение задачи сводится к решению *любой из подзадач*, соответствующих ее вершинам-потомкам. Во множестве вершин «И/ИЛИ» графа выделяют одну начальную вершину (вершину-корень), т.е. задачу, которую следует решить, и подмножество висячих конечных (целевых) вершин, т.е. заведомо разрешимых задач.

Пример дерева декомпозиции исходной НФЗ на подзадачи приведен на **рис. 7.5**. Здесь  $P_0$  – исходная задача, для решения которой требуется решить или подзадачу  $P_1$ , или подзадачу  $P_2$ . Решение подзадачи  $P_1$  сводится к решению подзадач  $P_3$  и  $P_4$ , в свою очередь решение подзадачи  $P_3$  возможно при наличии  $P_5$  или  $P_6$ . Решение подзадачи  $P_2$  сводится к решению подзадач  $P_7$  и  $P_8$ . Решение задач  $P_4, P_6, P_7$  предполагается известным, а  $P_5$  и  $P_8$  неизвестным.



**Рис. 7.5.** Дерево декомпозиции и решающий граф «и/или»

Цель процедуры поиска решения в «И/ИЛИ» графе – показать, что начальная вершина разрешима, т.е. для этой вершины *существует решающий подграф*.

Определение разрешимой вершины в «И/ИЛИ» графе можно сформулировать следующим образом:

- 1) конечные (целевые) вершины разрешимы, так как решение известно по исходному предположению;
- 2) «ИЛИ» – вершина разрешима тогда и только тогда, когда разрешима хотя бы одна из ее вершин-потомков;
- 3) «И» – вершина разрешима тогда и только тогда, когда разрешимы все ее вершины – потомки.

Таким образом, *решающий подграф «И/ИЛИ» графа* – это подграф из разрешимых вершин, который показывает, что начальная вершина разрешима в соответствии с приведенным выше определением. На **рис. 7.5** разрешимые вер-

шины заштрихованы, а неразрешимые не заштрихованы, двойными линиями выделен решающий подграф задачи  $P_0$ .

Для некоторой подзадачи могут быть неизвестны ни ее решения, ни способ сведения ее к более простым подзадачам. Такую подзадачу называют *неразрешимой*. Определение неразрешимой вершины в «И/ИЛИ» подграфе можно сформулировать следующим образом:

- 1) вершины, не являющиеся конечными и не имеющие вершин-потомков, неразрешимы;
- 2) «ИЛИ» – вершина неразрешима тогда и только тогда, когда неразрешима каждая из ее вершин-потомков;
- 3) «И» – вершина неразрешима тогда и только тогда, когда неразрешима хотя бы одна из ее вершин-потомков.

В общем случае процедуру решения можно представить в виде графа  $G=(X, Y)$ , где  $X=\{x_0, x_1, \dots\}$  – множество (в общем случае бесконечное) вершин графа, каждая из которых отображает одно из состояний;  $Y$  – множество дуг, инцидентных паре вершин  $(x_i, x_j)$ ;  $x_i, x_j \in X$ .

Множество дуг, исходящих из вершины  $x_i$ , отображает множество операторов, которые могут быть применимы к состоянию, отображаемому вершиной  $x_j$ . В множестве вершин  $X$  выделяют подмножество вершин  $X_0 \subseteq X$ , отображающее множество начальных состояний ( $S_0$ ), и подмножество вершин  $X_t \subseteq X$ , отображающее множество конечных (целевых) состояний ( $S_t$ ). Множество  $X_t$  может быть задано как явно, так и неявно, т.е. через свойства, которыми должны обладать целевые состояния. Очевидно, что решение неформализованных задач методом поиска в пространстве состояний сводится к процедуре поиска пути  $L$  в графе  $G$ . Путь из  $x_0 \in X_0$  в  $x_t \in X_t$  называют решающим (целевым). Построение пространства осуществляется с помощью следующей операции. К некоторой вершине из  $x_0 \in X_0$  применяют все возможные операции, порождающие все ее вершины-потомки. Порождение всех вершин-потомков для некоторой вершины  $x_j$  называют операцией раскрытия вершин. Если получена целевая вершина, то она не раскрывается. Операция построения пространства состояний заканчивается, когда все нераскрытые вершины являются целевыми.

Для поиска решений на «И/ИЛИ» графе необходимо обеспечить полноту поиска, т.е. организовать поиск так, чтобы все целевые вершины были найдены, если они существуют. Надежным способом обеспечения полноты является полный упорядоченный перебор всех вершин графа. Для каждой операции упорядоченного перебора необходимо определить порядок, в котором будут перебираться вершины графа. Обычно выделяют два основных способа перебора вершин: поиск «в глубину» и поиск «в ширину» (**рис. 7.6**). При поиске «в глубину» раскрывается та вершина, которая была построена самой последней, а при поиске «в ширину» вершины раскрываются в том же порядке, в котором порождаются. Поиск «в глубину» и «в ширину» называют рутинным, или слепым поиском, поскольку при этом порядок раскрытия вершин предопределен и не зависит от расположения цели. При увеличении пространства поиска способы слепого поиска требуют чрезмерных затрат времени и/или памяти компьютера. Для сокращения времени поиска применяют *эвристические методы* поиска, т.е. методы,

использующие некоторую информацию о предметной области для рассмотрения не всего пространства поиска, а таких рациональных путей в нем, которые с наибольшей вероятностью приводят к цели.

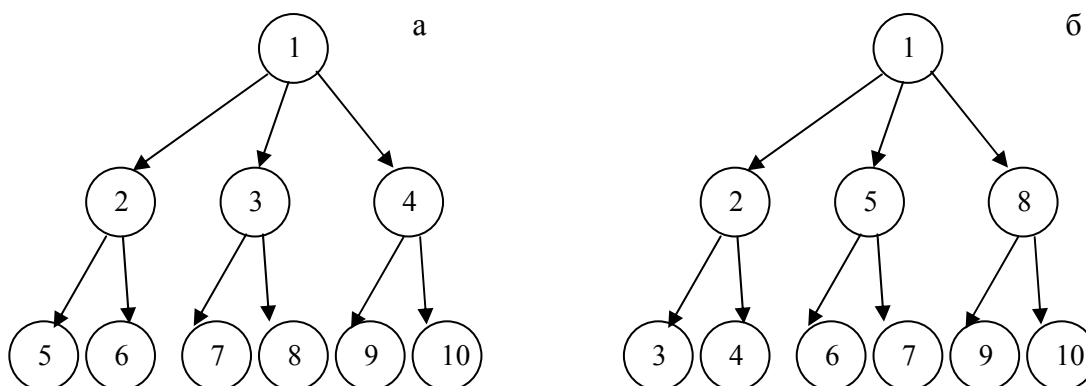


Рис. 7.6. Процедура решений способом «в ширину» (а) и «в глубину» (б)

#### 7.4.4. Методы неточных рассуждений с ненадежными данными

В неформализованных задачах, которые решают ЭС в металлургии, особенно при управлении сложными агрегатами и процессами, очень часто приходится использовать ненадежные или неточные знания и факты, которые невозможно представить только двумя значениями характеристических чисел, или идентификаторами (квантификаторами) – 1 (истина) и 0 (ложь). Тем не менее на основании неточных или ненадежных данных часто можно делать вполне определенные выводы и получать практические результаты. Для этого необходимо рассматривать комбинацию некоторых фрагментов знаний, а также знания по определенности или достоверности и в результате выводить новые знания, давая оценку их достоверности. Традиционно такую недостоверность, неопределенность или ненадежность знаний в современной физике и технике представляют вероятностью, подчиняющейся законам Байеса (байесовская вероятность), но в инженерии знаний нелогично иметь дело со степенью надежности в виде байесовской вероятности, приписанной знаниям изначально.

Среди методов неточных рассуждений с ненадежными данными одним из первых разработан метод с использованием так называемых *коэффициентов уверенности CF*, ставший эффективным средством обработки ненадежных знаний. Метод с использованием коэффициентов уверенности отображает ненадежные данные с помощью коэффициентов CF, которые принимают значения на отрезке  $[-1;1]$ , где «1» – заведомо истина, «-1» – заведомо ложь. Положительные CF называют мерой доверия, а отрицательные CF – мерой недоверия. В настоящее время разработан *метод нечетких выводов, названный субъективным байесовским методом*.

Для решения сложных НФЗ используется метод иерархической декомпозиции задачи на несколько подзадач. Связь между подзадачами, на которые декомпозирована неформализуемая задача, определяемая двумя понятиями – ис-

тина и ложь, может быть представлена через операции «И» и «ИЛИ», которые мы уже рассмотрели ранее. В задачах же с ненадежными исходными данными кроме связей «И» и «ИЛИ» важную роль играет *комбинированная связь «КОМБ» (СОМВ)*, которая подкрепляет или опровергает цель на основании двух и более доказательств.

Необходимость использования функции КОМБ (СОМВ) можно проиллюстрировать на примере диагностики развития тугого хода доменной печи. Пусть мы каким-либо образом определили, что фактор один – «наличие верхних пиков давления колошникового газа» – надежен со степенью 0,4, а фактор два – «общий перепад давления увеличивается» – надежен со степенью 0,6. Таким образом, развитие тугого хода печи при наблюдении только одного из факторов можно подтвердить с надежностью только 0,6 или 0,4 соответственно. Но если рассматривать оба фактора вместе, то логично считать состояние тугого хода более достоверным, чем по показаниям каждого из факторов в отдельности, т.е. больше 0,6. Как мы увидим в дальнейшем, при использовании КОМБ-правила степень надежности «развитие тугого хода» по двум этим факторам в совокупности составит 0,76.

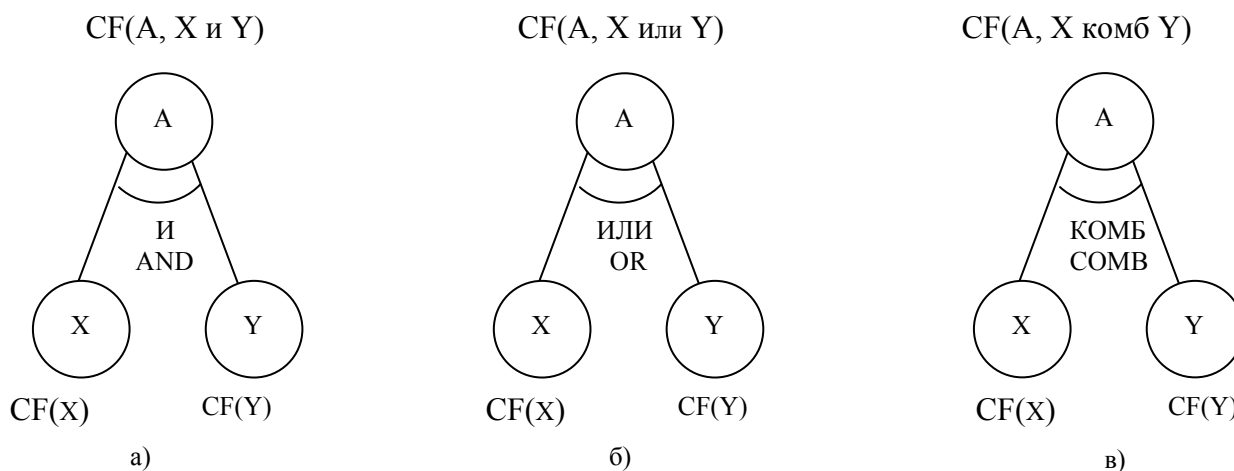
Суммарное действие факторов может не только увеличивать надежность диагноза, но и уменьшать ее. Пусть, например, фактор один имеет надежность 0,4, а фактор два – надежность -0,25. Тогда при комбинированной связи надежность диагноза, выводимого на основе этих двух факторов по КОМБ-правилу, уменьшится до 0,15.

На **рис. 7.7** представлено описание продукционных правил с помощью связей типа «И», «ИЛИ», «КОМБ», где  $X$ ,  $Y$  – результаты симптомов,  $A$  – цель или гипотеза.  $C_1$ ,  $C_2$ ,  $C_3$  – это степени надежности (коэффициенты уверенности) каждому продукционному правилу, численные значения которых необходимо определить.

Допустим, что каким-либо образом уже определены степени надежности  $X$  и  $Y$  как результат предыдущих выводов или наблюдений; необходимо сделать вывод или вычислить степень надежности  $A$ , используя продукционное правило и базы знаний.

Если выбрать метод выводов как для каждой из связи «И», «ИЛИ», так и для связи «КОМБ», то степень надежности можно распространить и на всю иерархическую сеть выводов. В итоге можно получить степень надежности конечной цели, а также указать ее при окончательном решении.





**Рис. 7.7.** Описание продукционных правил (ПП) с помощью связей «И», «ИЛИ», «КОМБ»

$C_1, C_2, C_3, C_{31}, C_{32}$  – степени надежности

а – если X и Y, то A со степенью надежности  $C_1$

б – если X или Y, то A со степенью надежности  $C_2$

в – 1- если X, то A со степенью надежности  $C_{31}$   
 2 - если Y, то A со степенью надежности  $C_{32}$   
 3 - если X КОМБ Y, то A со степенью надежности  $C_3$

Обозначим коэффициент уверенности исходных доказательств X и Y соответственно через  $CF(X)$  и  $CF(Y)$ , а степени надежности  $C_1, C_2, C_{31}, C_{32}, C_3$  вывода A в продукционном правиле обозначим соответственно через  $CF[A, X \text{ и } Y]$ ;  $CF[A, X \text{ или } Y]$ ;  $CF[A, X]$ ;  $CF[A, Y]$ ;  $CF[A, X \text{ комб } Y]$ .

При определении итогового коэффициента уверенности исходят из следующих правил:

- если в условии (предпосылке) продукционного правила только один член (например, X), то итоговый CF и есть  $CF(X)$ ,  
 $CF = CF(X)$ ; (7.1)
- при связи типа «И»(AND) из нескольких выводов обычно выбирают минимальное значение степени надежности, т.е.  
 $CF = \min \{CF(X), CF(Y)\}$ ; (7.2)
- при связи типа «ИЛИ»(OR)  
 $CF = \max \{CF(X), CF(Y)\}$ . (7.3)

Таким образом, выбирают минимальное значение степени надежности или нескольких выводов при связи «И» и максимальное при связи «ИЛИ», других подходящих методов пока не существует.

При связи же «КОМБ» (COMB) отдельно получают  $CF[A, X]$  и  $CF[A, Y]$ , а для определения CF можно использовать следующую комбинированную функцию:

$$CF = \begin{cases} +1, \text{ если } CF[A,X]=1 \text{ или } CF[A,Y]=1; & (7.4) \\ CF[A,X]+CF[A,Y] - CF[A,X]CF[A,Y], & \\ \text{если } CF[A,X]>0 \text{ и } CF[A,Y]>0; & (7.5) \\ CF[A,X]+CF[A,Y], & \\ \text{если } CF[A,X]CF[A,Y]\leq 0; & (7.6) \\ CF[A,X] \neq \pm 1 \text{ и } CF[A,Y] = \pm 1, & \\ CF[A,X]+CF[A,Y] + CF[A,X]CF[A,Y], & \\ \text{если } CF[A,X]<0 \text{ и } CF[A,Y] < 0; & (7.7) \\ -1, \text{ если } CF[A,X]= -1 \text{ или } CF[A,Y]= -1. & (7.8) \end{cases}$$

Значение коэффициента уверенности из трех и более независимых доказательств можно вывести, последовательно используя указанные выше формулы.

Хотя коэффициенты уверенности не имеют строгого обоснования, но благодаря простоте восприятия они нашли широкое применение во многих инструментальных программных средствах обработки знаний. Так, для условий примера диагностики тугого хода доменной печи, учитывая, что  $CF[A,X]=0,6$ ,  $CF[A,Y]=0,4$ , воспользовавшись соотношением (7.5), получим коэффициент уверенности, равный 0,76. Примеры использования этих коэффициентов в ЭС металлургических агрегатов мы рассмотрим в дальнейшем.

## 7.5. Инструментальные средства построения экспертных систем

### 7.5.1. Традиционные языки программирования

В эту группу инструментальных средств входят и традиционные языки программирования (С, С++, Object Pascal и т.д.), ориентированные в основном на численные алгоритмы и слабо подходящие для работы с символьными и логическими данными. Поэтому создание систем искусственного интеллекта на основе этих языков требует большой работы программистов. Использование традиционных языков программирования позволяет включать интеллектуальные подсистемы (например, интегрированные экспертные системы) в крупные программные комплексы общего назначения. Среди традиционных языков удобными являются объектно-ориентированные языки.

### 7.5.2. Языки искусственного интеллекта

Это прежде всего языки Пролог и Лисп. Универсальность этих языков меньшая, по сравнению с традиционными языками, но потерю универсальности языки искусственного интеллекта компенсируют богатыми возможностями по работе с символьными и логическими данными, что крайне важно для задач искусственного интеллекта.

Пролог и Лисп относятся к числу языков, в основу которых положены идеи логического программирования. Основное отличие логического программирования от традиционного состоит в том, что программа является не четко определенной последовательностью операторов, а представляет собой формальное описание предметной области, пользователь не предписывает компьютеру, каким образом она должна решать задачу, а описывает ее.

### **7.5.3. «Оболочки» и программные обстановки**

Под «Оболочками» понимают «пустые» версии существующих экспертных систем, т.е. готовые экспертные системы с пустой базой знаний. Достоинство оболочек в том, что они вообще не требуют работы программистов для создания готовой экспертной системы. Требуются только специалисты в предметной области и инженеры по знаниям для заполнения базы знаний. Однако если некоторая предметная область плохо укладывается в модель, используемую в некоторой оболочке, заполнить базу знаний в этом случае весьма сложно. Для получения прикладной ЭС пользователь должен создать свою собственную базу знаний, используя представленный оболочкой язык представления знаний. Манипуляцию знаниями, генерацию объяснений, а также сервис разработки и отладки базы знаний обеспечивают встроенные средства оболочки.

В настоящее время разработан широкий спектр оболочек экспертных систем. Перечислим наиболее известные и распространенные сейчас «оболочки» ЭС.

Примером такой оболочки может служить EMYCIN (Empty MYCIN – пустой MYCIN), которая представляет собой незаполненную экспертную систему MYCIN. Оболочка EXSYS – сравнительно несложная система продукционного типа с обратным методом вывода. Оболочка VP Expert – компактная система, основанная на системах продукций. Недостатком применения оболочек является возможное несоответствие конкретной оболочки разрабатываемой с ее помощью прикладной ЭС.

*Программные обстановки (Средства автоматизированного создания ЭС)* для разработки экспертных систем представляют собой высокоуровневые системы поддержки автоматизированной разработки и создания проблемно-ориентированных ЭС. Они, как правило, содержат богатый спектр языков и моделей представления знаний. Построение ЭС с помощью рассматриваемых средств заключается в формализации исходных знаний, записи их на входном языке представления знаний и описании правил логического вывода решений. Далее экспертная система наполняется знаниями. Наиболее известной программной обстановкой является система KEE (среда инженерии знаний). Система имеет большие возможности для работы с фреймами, обладает средствами тестирования и отладки знаний, прекрасным пользовательским интерфейсом и графикой.

Система Gold Work сочетает в себе различные формы представления знаний, такие как фреймы и продукции. Имеются средства имитации параллельной работы нескольких групп продукций, поддержки немонотонной логики, за-

дания факторов уверенности. Система позволяет использовать окна, меню, графику для организации пользовательского интерфейса, обеспечивает гибкий способ доступа к базам.

Система Nexpert Object является мощным средством разработки ЭС, способным конкурировать с такими системами, как Gold Work и KEE, и имеющим подобную организацию базы знаний. Система работает в среде MS WINDOWS, имеет интерфейс с языками программирования, хорошие возможности работы с базами данных.

Система Personal Consultant Plus разработана крупной американской фирмой Texas Instrument Inc., которая обеспечивает ее сопровождение. В системе используется правило «Если ... то», к которым добавлены некоторые объектно-ориентированные среды. Эти средства представлены контекстными деревьями с небольшими модификациями, позволяющими помещать факты в контексты или объединять средства, производящие вывод по прямой и обратной цепочке.

Некоторые инструментальные системы написаны с использованием данного инструментального средства и весьма эффективно эксплуатируются в промышленности. Например, «Weld Select» помогает инженерам по сварочным швам выбрать лучшие электроды для сварки металлов.

## 7.6. Контрольные вопросы

1. Какие системы называются интеллектуальными? В чем заключаются особенности их работы?
2. Чем отличается бионический подход при исследовании в области искусственного интеллекта от прагматического?
3. Что такое экспертная система? Какие основные компоненты она в себя включает? Для решения каких задач используют экспертные системы?
4. Чем пользователь-эксперт отличается от обычного пользователя при работе с экспертной системой?
5. С какой целью используют модели представления знаний?
6. Каким образом происходит представление знаний с использованием продукционной модели? В чем заключается отличие способов использования правил при прямой и обратной цепочке рассуждений?
7. Какие принципы положены в основу логической модели представления знаний?
8. Какова основная идея способов представления знаний на основе аппарата семантических сетей?
9. Каким образом решается проблема обработки и представления ненадежных или неточных знаний и фактов в экспертных системах?
10. С помощью каких инструментальных средств создаются экспертные системы?

ГЛАВА 8. ЭКСПЕРТНЫЕ СИСТЕМЫ В ДОМЕННОМ ПРОИЗВОДСТВЕ	8-2
8.1. Состояние вопроса	8-2
8.2. Технологические основы экспертных систем доменной плавки	8-4
8.3. Общая характеристика ЭС «Интеллект доменщика»	8-25
8.4. Обработка входной информации	8-30
8.5. Характеристика базы знаний	8-34
8.6. Контрольные вопросы	8-41

## **Глава 8. ЭКСПЕРТНЫЕ СИСТЕМЫ В ДОМЕННОМ ПРОИЗВОДСТВЕ**

### **8.1. Состояние вопроса**

Несмотря на многовековую историю, процесс доменной плавки изучен еще не во всех деталях. Как мы уже отмечали, проблемой всех существующих математических моделей является недостаточно высокая точность значений входящих в них параметров. Это относится, в частности, к факторам, определяющим порозность столба шихты, поля скоростей движения твердых и жидких фаз, распределения газового потока, к параметрам скорости восстановления железорудных материалов и т.п. Процесс выплавки чугуна в доменной печи подвержен влиянию многих переменных контролируемых и неконтролируемых факторов, которые вызывают нарушения хода печи и ее теплового состояния и требуют высокой квалификации технологического персонала при управлении доменной плавкой.

Вместе с тем ведущими специалистами-металлургами накоплен громадный запас знаний и опыта ведения процесса, диагностики агрегатов, что позволяет им вести технологию при отсутствии части объективной информации и неполной оценке некоторых аспектов состояния агрегатов и технологии.

Для решения этих проблем с целью диагностики и управления ходом доменной печи создаются экспертные системы, построенные в концепции искусственного интеллекта и включающие в себя опыт специалистов-доменщиков. Системы с применением искусственного интеллекта (ИИ) создаются с целью стабилизации хода печи и повышения качества чугуна. Экспертные системы способны использовать детерминированные знания о процессе и обобщать практический опыт специалистов. Процесс накопления знаний и опыта в ЭС продолжается после ввода ее в действие, учитывая последующие изменения технологии. Экспертная система повышает уровень знаний среднего специалиста до уровня квалифицированного, обеспечивает единообразие решений и оперативную адаптацию в изменяющихся условиях массового, интенсивного производства.

Наибольшее распространение экспертные системы для управления ходом доменных печей получили в Японии. Первая ЭС, разработанная компанией Nippon Still, была установлена для опытной эксплуатации на доменной печи №4 в Кимицу (1984 г.) и введена в промышленную эксплуатацию в 1986 г. ЭС выполняет две функции: диагностику аномальных состояний технологического процесса и управление тепловым режимом печи.

Система ИИ с развитой логикой – ALIS, примененная на доменных печах № 3 и 4 в Кимицу, осуществляет контроль хода печи, состояния горна, движения шихты, контроль состояния оборудования. Система ALIS содержит два комплекта баз знаний, что позволяет проводить ее корректировку в реальном масштабе времени. Коэффициент использования рекомендаций регулирующих воздействий, назначенных системой, достигает 95 %.

Система управления BASYS разработана и применена на доменной печи № 5 завода Фукуяма фирмы NKK в 1986 г. Система построена по иерархическо-

му принципу с использованием баз знаний и порождающих правил и выполняет те же функции, что и ALIS. В системе реализованы законы нечеткой логики с использованием коэффициентов достоверности информации, что облегчает формирование баз знаний. Математическое обеспечение обработки информации построено на применении трехмерных функций принадлежности для нечетких множеств, что позволяет сократить количество порождающих правил. Система характеризуется высокой надежностью в эксплуатации, коэффициент использования, т.е. отношение числа реализованных команд к их общему числу, составляет 97 %. В настоящее время ЭС BASYS стала применяться и для автоматизированного управления распределением шихтовых материалов по сечению колошника.

Опыт использования ЭС для управления доменной печью имеет также фирма «Кавасаки Сэйтецу» на доменной печи № 4 в Мидзусима. Она ввела в эксплуатацию усовершенствованную систему управления ходом доменной печи, построенную по концепции ЭС. Технические средства системы включают ЭВМ управления доменной печью (сбор, предварительная обработка и передача данных) и ЭВМ искусственного интеллекта (обработка полученных данных, обработка знаний по интеллектуальным правилам для диагностики состояния печи, выработка регулирующих воздействий).

ЭС BASYS также применяется и на доменной печи № 1 завода Кейхин фирмы «Кобе Стил» для управления тепловым состоянием печи. При использовании ЭС среднеквадратичное отклонение температуры чугуна на выпусках составило 9 °С, без нее – 13 °С при колебании температуры на выпусках 1480 – 1520°С.

С 1990 г. ходом доменной плавки на доменной печи № 2 завода Кокура управляет ЭС, состоящая из двух подсистем: собственно управления плавкой и принятия решений. Внедрение системы позволило уменьшить дисперсию температуры выпускаемого чугуна.

Для управления работой на доменной печи № 2 на заводе Сита внедрена и успешно эксплуатируется система SAFAlA, реализующая системный подход к анализу состояния печи с применением искусственного интеллекта. Для оценки формы и положения зоны когезии в системе используют данные о температуре холодильников, индекс газопроницаемости, гранулометрический состав шихты, сход шихты и др. При эксплуатации системы в 1989 г. общее число удачных рекомендаций и оценок состояния печи, выполненных системой, составляло, спустя два месяца после начала работы системы, 95–98 %.

Гибридная ЭС как перспективная компьютерная система управления доменной печью с использованием математических моделей и эмпирических правил была разработана и применена в 1988 г. на доменной печи № 1 в Касима. Система сочетает достоинства математических моделей (высокая эффективность управления) и достоинства эмпирических правил (устойчивость контролируемых процессов). При нормальном состоянии печи управление осуществляется по моделям, при аномальном состоянии – по эмпирическим правилам. Выбор метода управления выполняется по эмпирическим правилам оценки.

В Южной Корее с 1990 г. работает ЭС реального времени. Система выполняет следующие функции: диагностику образования неактивной зоны в ниж-

ней части шахты, нарушения равномерности газового потока в печи; выработку управляющих воздействий для предотвращения нестабильной работы печи.

На доменных печах Европы и Северной Америки ЭС с искусственным интеллектом еще не получили такого широкого распространения, как в Японии, и находятся на стадии опытных разработок. Например, в Бельгии в Центре металлургических исследований (С.Р.М.) разработана ЭС – ACCES. Система позволяет выявить на ранней стадии нарушения в работе печи, например сползание гарнисажа, определить их причины и предложить управляющие воздействия для их ликвидации. При прогнозировании похолоданий горна система обеспечивает на 100 % прогноз тяжелых похолоданий и на 75 % – легких похолоданий.

Таким образом, мировой опыт свидетельствует об эффективности использования экспертных систем для диагностики и управления тепловым и газодинамическим режимами, наиболее сложным и самым энергоемким переделом современной металлургии – доменной плавкой.

В России разработкой научных основ создания экспертных систем доменной плавки активно занимаются ученые ряда организаций: НПО «Черметавтоматика», Московского института стали и сплавов (технический университет), Российского института искусственного интеллекта, Института системного анализа РАН, Института проблем управления, Уральского государственного технического университета (УГТУ–УПИ), Уральского отделения РАН и др.

Однако наилучшие практические результаты достигнуты в этой области в России сотрудниками Научно-инженерного центра ассоциации содействия Всемирной лаборатории (г. Москва). В дальнейшем мы рассмотрим опыт разработки и внедрения в доменном цехе ОАО «Магнитогорский металлургический комбинат» первой отечественной промышленной экспертной системы реального времени «Интеллект доменщика», разработанной сотрудниками этого центра.

## 8.2. Технологические основы экспертных систем доменной плавки

Хорошие технико-экономические показатели доменной плавки – высокую производительность и низкий удельный расход кокса – можно получить только при "нормальной работе" печи. Понятие о нормальной работе связывается с так называемым ровным ходом печи. Такой ход характеризуется следующими признаками.

1. Плавным и устойчивым сходом шихтовых материалов с постоянной скоростью опускания кусков шихты в поперечном сечении печи. Здесь важно отметить, что скорость опускания материалов в отдельных точках сечения печи может быть различной, но при ровном ходе плавки эта скорость должна быть постоянной.

2. Устойчивым характером распределения шихтовых материалов и газового потока по сечению доменной печи, при котором достигается максимальная степень использования химической и тепловой энергии газа и минимальный для данных условий плавки удельный расход кокса.

3. Постоянными параметрами дутьевого режима, т.е. условиями стабильности расхода дутья, его температуры и влажности, а также постоянными расхо-



дами природного газа (пылеугольного топлива или мазута) и технологического кислорода.

4. Устойчивым тепловым состоянием плавки, когда имеет место получение чугуна и шлака заданного состава с соответствующими физико-химическими свойствами.

5. Наличием устойчивого гарнисажа, что обеспечивает постоянство рабочего профиля печи и его сохранность в процессе эксплуатации печи.

Ровный ход достигается строгим соблюдением всех технологических инструкций, начиная от приемки и складирования шихтовых материалов, загрузки их в печь, выполнения технологии плавки и заканчивая выпуском продуктов доменной плавки.

При ведении доменной плавки, несмотря на сравнительно большую тепловую инерцию современных доменных печей, возможно значительное отклонение режимных параметров плавки от заданных. Причины таких отклонений от ровного хода плавки могут быть разнообразными: изменение химического и гранулометрического составов шихтовых материалов и несвоевременная реакция технологического персонала на произошедшие изменения, выход из строя основного и вспомогательного технологического оборудования, а также ошибки персонала, ведущего доменную плавку, и др.

Наблюдение и корректировка режима доменной плавки осуществляется как визуальным наблюдением, так и с помощью системы отображения информации.

При распознавании того или иного вида отклонения печи от нормального режима плавки недопустимо делать вывод по 1-2 признакам. Полученная информация должна оцениваться по совокупности признаков и сопоставляться. В противном случае не исключается возможность принятия ошибочных решений, которые могут еще в большей степени усугубить наступившее нарушение плавки.

Визуальное наблюдение за ходом доменной плавки технологическим персоналом предусматривает, прежде всего, оценку светимости воздушных фурм. Светлые яркие фурмы с раскаленными до белого каления кусками кокса указывают на высокий нагрев печи. Переход от белого накала кусков кокса к розовому характеризует заметное уменьшение нагрева. Появление на фурмах красных кусков кокса и темных кусков железорудного материала свидетельствует о значительном снижении теплового состояния доменной плавки.

Следующий визуальный параметр, который можно использовать для суждения о ходе плавки, – это характер циркуляции кусков кокса в фурменном очаге. При нормальном технологическом режиме в глазке фурмы можно наблюдать интенсивную циркуляцию кусков кокса. При замедлении доменного процесса наблюдается вялая циркуляция кусков твердого топлива. Это может быть одним из признаков горячего хода плавки. При нижнем подвисании шихты, а также при работе печи в режиме тихого хода возможно полное отсутствие циркуляции кокса в зоне горения.

### *Параметры, характеризующие тепловое состояние печи*

Химический состав чугуна. Чувствительным к нагреву горна печи является содержание в чугуне кремния, марганца и серы. С увеличением нагрева печи наблюдается возрастание содержания кремния и марганца в чугуне и снижение в нем серы.

Излом пробы чугуна. Здесь отметим, что кремний в чугуне способствует выделению углерода в нем в виде графита, дающего в изломе пробы серый цвет. В том случае, когда углерод в чугуне связан в виде карбидов железа ( $\text{Fe}_3\text{C}$ ) или марганца ( $\text{Mn}_3\text{C}$ ), излом чугуна будет белым. В связи с этим чугун с содержанием кремния более 0,6-0,7 % имеет в изломе кристаллический излом серого цвета (серый чугун), а при содержании кремния менее 0,5 % излом пробы чугуна становится аморфным и имеет белый цвет (белый чугун). В последнем случае большая часть углерода связана с железом в виде прочного соединения карбида железа ( $\text{Fe}_3\text{C}$ ). При суждении о тепловом уровне горна печи по излому пробы чугуна следует иметь в виду, что на вид излома пробы большое влияние оказывает скорость ее охлаждения. При быстром охлаждении образуются карбиды железа, поэтому при охлаждении пробы чугуна водой или при заливке чугуна в металлическую пробницу (кокиль) периферийная часть пробы бывает чаще всего "отбеленной". Напротив, проба, отлитая в песке и постепенно охлаждаемая в нем, при тех же условиях дает серый излом.

Искрение чугуна при выпуске. При достаточном производственном опыте о составе чугуна и его нагреве можно сравнительно точно судить визуально по характеру искрения чугуна и выделению графита на выпуске. Здесь основная закономерность проявляется в том, что чем больше кремния в чугуне (т.е. чем выше его нагрев), тем меньше его искрение при выпуске и тем больше присутствует выделение графита в окружающее пространство и на поверхности чугуна в виде скоплений (так называемая "спель"). Объясняется это явление тем, что при выходе чугуна из печи на литейный двор и снижении его температуры уменьшается растворимость углерода в чугуне, что вызывает усиленное выделение графита. Горячий литейный чугун с содержанием кремния до 3 % и выше практически не дает искр. Передельные чугуны с содержанием кремния менее 1 % искрят обильно по мере снижения его нагрева. Что касается природы искрения чугунов при их выпуске, то это объясняют окислением поверхностных слоев чугуна, появлением газообразных оксидов железа, разрывающих пленку чугуна. При выплавке чугунов с содержанием марганца в пределах 0,5-1,0 % по искрению чугуна можно судить и о содержании этого элемента в металле. В этом случае на поверхности зеркала чугуна наблюдается появление мелких и частых искр - язычков желтого цвета от окисления марганца на поверхности чугуна. По внешнему виду чугуна за перевалом, где зеркало металла освобождено от шлака, можно делать заключение о нагреве печи и содержании в нем серы. Сера в соединениях с железом в виде ( $\text{FeS}$ ) при холодном ходе печи легко заметна на поверхности чугуна в виде отдельных темных пленок. Однако появление таких пленок возможно и при нормальном тепловом режиме плавки, но при плохом развитии процессов десульфурации.

Состав шлака. О тепловом уровне доменной плавки возможно судить по цвету шлака. При похолодании печи наблюдается снижение степени перехода в чугун таких элементов, как железо и марганец. Вследствие этого в шлаке возрастает содержание монооксида железа (FeO) и монооксида марганца (MnO). Известно, что FeO окрашивает шлак в бутылочно-зеленый цвет, а затем - в черный, а монооксид марганца окрашивает шлак в горохово-зеленый цвет. Нагрев доменной печи изменяет основность шлака. Повышение основности шлака (при неизменности состава и расходов загружаемых шихтовых материалов) свидетельствует о повышении нагрева печи. Это связано с уменьшением содержания кремнезема в шлаке вследствие восстановления кремния и перехода его в чугун. Основность шлака сравнительно легко определяется визуально по виду излома остывшего шлака. Шлак высокой основности имеет серый камневидный излом, а кислый – полностью стекловидный или имеющий стекловидную корочку. В расплавленном состоянии основность шлака определяется при помощи пики, на которую он берется из желоба. Кислый шлак при поднятии пики над уровнем шлака дает длинные стекловидные нити ("длинный" шлак), основной шлак нитей не дает. Остывший основной шлак склонен к рассыпанию в силу того, что двухкальциевый силикат ( $2\cdot\text{CaO}\cdot\text{SiO}_2$ ), входящий в состав основного шлака, при охлаждении при температуре  $675\text{ }^\circ\text{C}$  переходит из  $\alpha$ -фазы в  $\gamma$ -фазу, что сопровождается увеличением объема примерно на 10 %. Это и приводит шлак к разрушению с образованием мелкого порошка. Признаком, определяющим недостаточный нагрев печи, является наличие так называемого "греналя" холодного железистого шлака, образующего помимо расплавов чугуна и шлака третью фазу, плывущего на поверхности чугуна тонким слоем. Греналь трудно отделить от чугуна и задержать на перевале. Именно появление греналя является часто причиной "закозления" шлаковозов.

### *Показатели, характеризующие сход шихты в печи*

Как известно, для фиксации уровня шихты в печи и для записи опускания материалов устанавливаются два зонда, движение которых фиксируется системой отображения технологической информации. Зонды поднимаются в верхнее положение перед загрузкой материалов в печь для предотвращения засыпания их шихтой, а затем после опускания очередной подачи в печь опускаются на уровень шихтовых материалов и движутся вместе с ней до заданного уровня насыпи. Затем описанный цикл повторяется.

Второй параметр, который характеризует сход шихтовых материалов в доменной печи, – это число сработанных подач в единицу времени. Конечно, ровному ходу доменной плавки отвечает постоянство числа сработанных подач в час (в смену, сутки ...).

Давление горячего дутья является также показателем ровности хода печи. Нормальному сходу шихты отвечает постоянство давления дутья. Допускаются изменения давления дутья в пределах 10 кПа (примерно 0,1 атм) за 30 минут.

Масса чугуна за выпуск при ровном ходе плавки должна колебаться для доменных печей среднего и большого объема в пределах 20-30 тонн. Вынос ко-

лошниковой пыли при ровном ходе доменного процесса постоянный. Это также является показателем ровного хода печи.

### *Показатели, характеризующие устойчивость распределения газового потока по сечению печи*

Прежде всего, таким показателем является кривая изменения содержания диоксида углерода ( $\text{CO}_2$ ) по диаметру колошника доменной печи. Наилучшие показатели доменной плавки – высокая производительность и низкий расход кокса – достигаются при оптимальном распределении материалов и газов по сечению печи. Этому соответствует оптимальное изменение  $\text{CO}_2$  по диаметру колошника. При устойчивом распределении газового потока текущие значения  $\text{CO}_2$  в кольцевых сечениях печи имеют небольшие колебания относительно установленных оптимальных значений.

Следующий параметр, характеризующий устойчивость газораспределения, – это температура колошниковога газа. Температура отходящего из печи газа фиксируется термопарами, установленными в каждом газоотводе в непосредственной близости от уровня шихты (примерно 2-3 метра от уровня материалов). Температура, фиксируемая каждой термопарой, имеет синусоидальный вид, что связано с периодичностью загрузки в печь холодных шихтовых материалов с последующим их прогревом газом до момента последующего опускания большого конуса, а также с периодичностью попадания гребня шихтовых материалов, образующегося во вращающейся воронке распределителя шихты в различные секторы колошника. В связи с этим фиксируемые температуры на диаграммной ленте укладываются в область определенной ширины, которая для условий ровного хода плавки равна 50–60 °С.

Критерием устойчивости распределения газового потока по окружности печи может являться также температура периферийных газов. Эта температура фиксируется обычно шестью - восемью термопарами, установленными по окружности печи. Термопары устанавливаются заподлицо с огнеупорной кладкой под броневой защитой колошника и заключаются в массивные чехлы, поэтому они имеют гораздо большую инерционность по сравнению с термопарами, регистрирующими температуру колошниковога газа. При нормальном ходе печи температуры периферийных газов укладываются в полосу шириной 100–150 °С.

### *Информационное отображение нормального хода доменной плавки*

Расход дутья представляет собой ровную линию с небольшими колебаниями, связанными с накоплением в горне и выпуском жидких продуктов плавки. Периодически на диаграмме появляются "пики" в сторону увеличения расхода в момент перевода воздухонагревателей с "газа" на "дутье", когда идет заполнение нагретого воздухонагревателя дутьем и уменьшается сопротивление тракта подачи горячего дутья в печь (рис. 8.1).

Давление горячего и холодного дутья имеет вид ровной маломеняющейся линии. Имеются пульсации давления горячего дутья в пределах 0,05 атм (около 5 кПа). В соответствии с расходом дутья в момент перевода воздухонагрева-

телей из одного режима в другой, на диаграмме давления горячего дутья появляются пики в сторону снижения давления. Следует отметить, что изменения, происходящие в доменной печи, в меньшей степени отражаются на значении давления холодного дутья по сравнению с давлением горячего дутья.

Общий и частные перепады давления газа представляют собой ровные линии с небольшими периодическими пиками в сторону повышения давления в момент опускания очередной подачи в печь. При этом пикообразное возрастание давления более отчетливо проявляется на показаниях верхнего перепада давле-

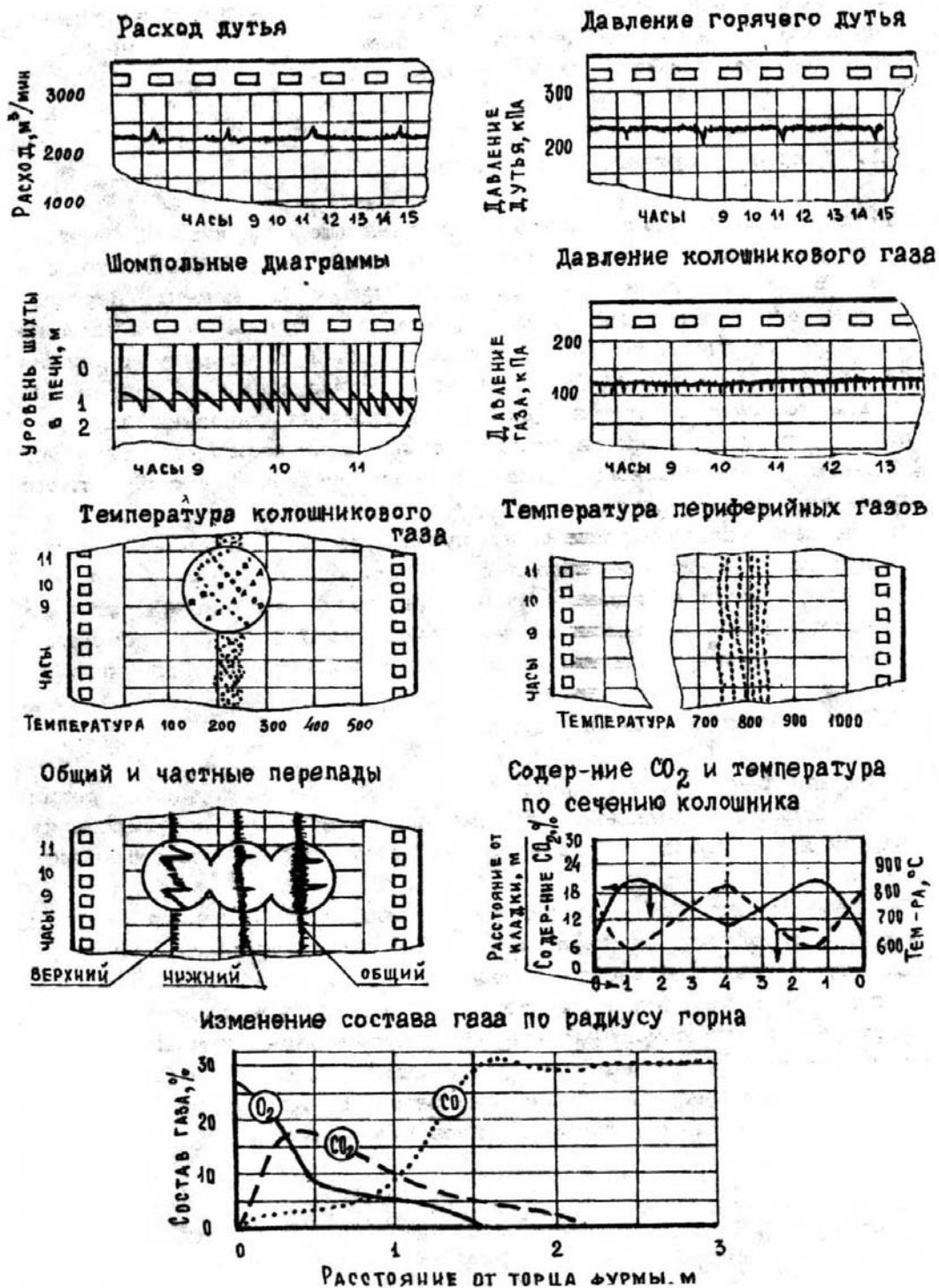


Рис. 8.1. Тренд параметров при нормальном ходе доменной плавки

ния газа и в меньшей степени – на нижнем.

Сущность замера уровня засыпи и отображения движения материалов в печи описана ранее. Правильность показаний устройств для регистрации уровня шихты в печи периодически тщательно проверяется. Допустимая разница в показаниях уровнемеров обычно не должна превышать 0,25 метра. Частота опускания зондов свидетельствует о числе сработанных подач. Меняющаяся частота опускания зондов не всегда является признаком неровного движения материалов в печи. В процессе накопления расплавов чугуна и шлака и подхода их к уровню воздушных фурм скорость опускания материалов в печи уменьшается, а после выпуска возрастает. При нормальном ходе доменной плавки на показаниях зондовых диаграмм отсутствуют горизонтальные площадки и вертикальные провалы, свидетельствующие о подвисаниях и осадках шихты.

При ровном ходе доменной плавки показания термопар, установленных в газоотводах печи, регистрируют синусообразное изменение температур, связанное с периодичностью загрузки в печь холодных порций шихтовых материалов и последующим их прогревом, а также с работой вращающегося распределителя шихты (ВРШ). Ровный ход доменной плавки характеризуется тем, что разброс показаний термопар не превышает 50–60 °С.

Температура периферийных термопар представляет несколько линий с шириной разброса не более 100–150 °С, при этом средняя температура периферийных газов при нормальном режиме работы составляет 700–850 °С.

Давление колошникового газа при работе системы локального автоматического регулирования представляет собой относительно ровную линию с регулярными и равномерными понижениями, связанными с опусканием шихты в печь и носящими название "нижних пик". При нормальном ходе печи на диаграмме отсутствуют пики в сторону повышения давления, а длина нижних пик практически одинакова.

Распределение содержания диоксида углерода ( $\text{CO}_2$ ) по диаметру колошника при нормальном режиме доменной плавки имеет малые отличия от оптимального. При этом изменение температуры газа является зеркальным отражением кривой изменения  $\text{CO}_2$  по диаметру (радиусу) колошника доменной печи.

### *Нарушение устойчивости газового потока в печи*

#### *Периферийный газовый поток*

Этот вид расстройтва газораспределения в печи характеризуется преимущественным движением газового потока у стен печи, когда газ движется широким потоком в периферийных участках столба шихты. Центральная зона столба шихты слабо омывается газом и становится малопроницаемой.

При таком виде расстройтва наблюдается ухудшение в использовании тепловой и восстановительной энергии газа, что в конечном итоге приводит к возрастанию удельного расхода кокса, уменьшению производительности доменной печи, к устойчивому загромождению горна, к ускоренному разгару огнеупорной кладки печи и частому прогару холодильников шахты.

Причины, вызывающие периферийный газовый поток, могут быть различными, в частности:

- 1) длительная работа доменной печи на разрыхляющих периферию системах загрузки. Например, увеличенная доля подач "коксом вперед" (ККРР↓ или КРРК↓);
- 2) длительная работа печи на излишне больших подачах. Известно, что большие подачи способствуют загрузке центральной части столба шихты железорудными материалами, что может привести к закрытию центра столба шихты этими материалами и спровоцировать движение газа в периферийных участках шихты;
- 3) загромождение горна доменной печи, затрудняющее движение газового потока в центральной зоне печи;
- 4) работа печи с малым расходом дутья и связанная с этим малая кинетическая энергия истечения дутья, что сопровождается слабым развитием фурменных очагов;
- 5) перегрузка осевой зоны столба шихты мелкими материалами за счет их перемещения из периферийных участков слоя шихты.

На **рис. 8.2** условно показан тренд параметров при переходе печи из нормального режима плавки в режим периферийного газового потока.

В начальный период появления периферийного движения газа давление горячего дутья первоначально уменьшается на 0,1–0,15 атм (10–15 кПа). Однако затем по мере загромождения горна давление дутья начинает возрастать. Расход дутья в начальный период расстройств самопроизвольно возрастает, а затем по мере потери активности работы центра печи начинает уменьшаться.

Воздушные фурмы первоначально работают интенсивно, но по мере потери активности центра работа фурм становится "вялой" с периодическим появлением на фурмах неподготовленных материалов. Разность температур в отдельных газоотодах (температура колошникового газа) увеличивается до 100–150°C при общем уровне 350–400 °C при выплавке передельного чугуна. Термопары, показывающие температуру периферийных газов, фиксируют повышенный нагрев кладки печи – температура возрастает на 100–200 °C по отношению к нормальному режиму плавки.

Из-за ухудшения восстановительной работы газового потока содержание диоксида углерода в колошниковом газе уменьшается, а содержание водорода растет. Содержание диоксида углерода на периферии уменьшается и становится ниже оптимального, а у оси печи возрастает и превышает оптимальное значение. Максимальное значение диоксида углерода на кривой изменения  $\text{CO}_2$  по диаметру (радиусу) колошника перемещается к центру печи. Первоначально наблюдается ускоренный сход шихты, что можно наблюдать по шомпольным диаграммам.

При развитии периферийного газового потока температура жидких продуктов плавки понижается. На диаграмме давления колошникового газа периодически появляются "верхние пики" (мгновенное повышение давления), что не свойственно нормальному ходу плавки. Увеличивается длина "нижних пик".

Меры по ликвидации периферийного потока газа бывают различными в зависимости от степени его развития. При относительно свободном центре мо-

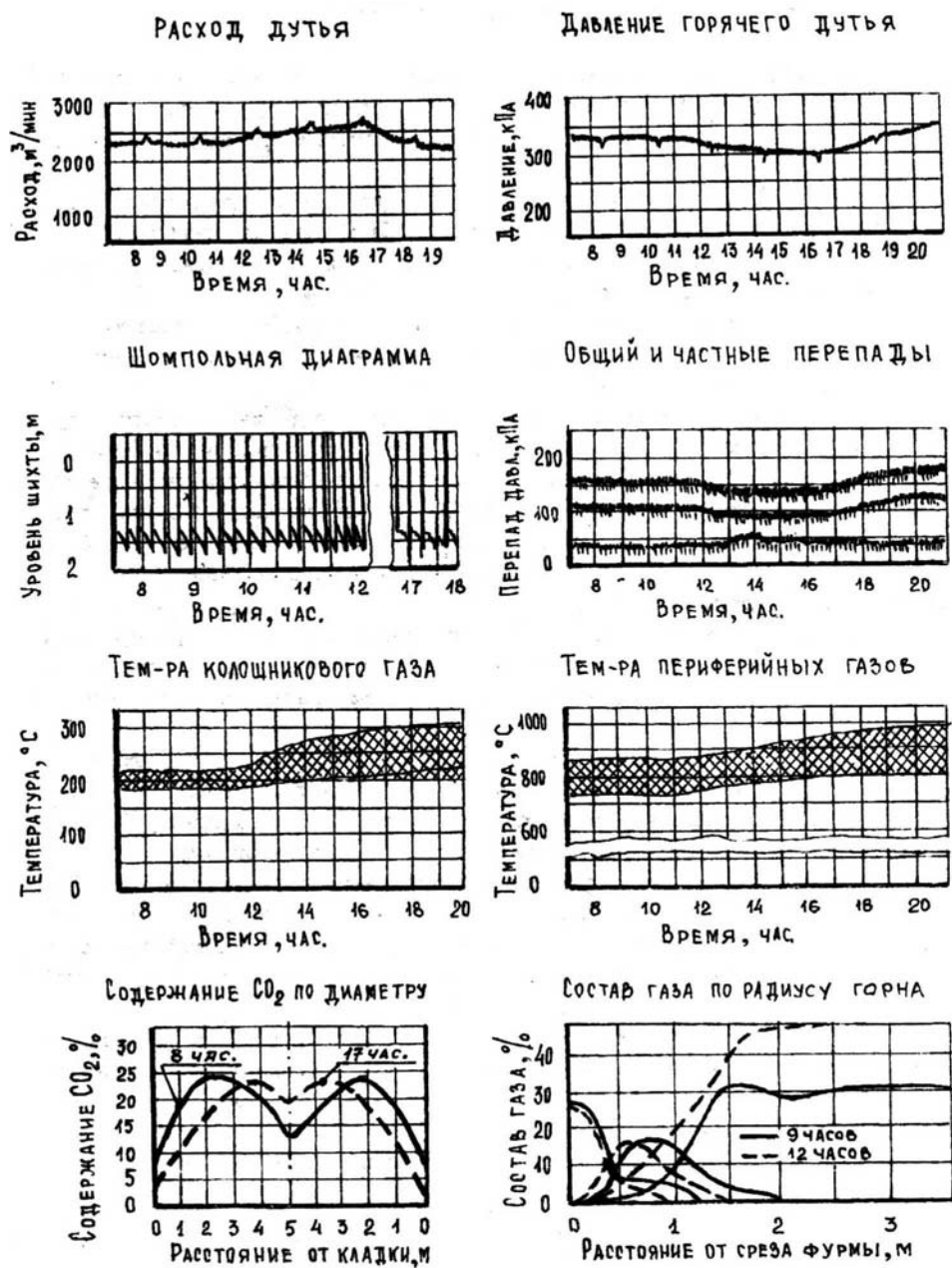


Рис. 8.2. Тренд параметров при развитии периферийного потока газа

жет быть вполне достаточным устранение активности периферии путем увеличения рудной нагрузки в эту зону. При загроможденном центре сокращение периферийного хода газов путем загрузки периферии недопустимо. Это приведет к серьезному подвисанию шихты в печи. В последнем случае (при загроможденном центре) необходимо принять меры для разрыхления центральной зоны столба шихты. После разгрузки центра необходимо осуществление следующих мероприятий.

- 1) уменьшить массу подачи;
- 2) постепенно подгружать периферию путем изменения системы загрузки или путем понижения уровня засыпи;



- 3) понизить концентрацию кислорода в дутье, сокращая при этом расход природного газа (мазута, пылеугольного топлива);
- 4) понизить давление колошникового газа;
- 5) для поддержания достаточного нагрева печи загрузить несколько «холостых подач». При достижении устойчивой работы центральной зоны печи постепенно восстанавливать параметры технологического режима до исходного уровня.

### *Центральный (осевой) поток газа в печи*

Это отклонение от нормального газораспределения, противоположное периферийному газовому потоку. Центральный поток газов является результатом длительной работы доменной печи на системах загрузки, излишне загружающих периферию, при работе печи с малыми подачами, а также с пониженным уровнем засыпи.

Тренд основных параметров доменной плавки при осевом потоке газового потока представлен на **рис. 8.3**.

Температура колошникового газа повышенная, а разброс точек отдельных показаний термопар снижается, при этом полоса показаний термопар приобретает вид узкой ленты. Температура периферийных газов уменьшается до 650-700 °С, а ширина разброса этих температур возрастает. Из-за возрастания газодинамического сопротивления слоя шихты давление горячего и холодного дутья возрастает.

На шомпольных диаграммах меняется вид схода подач: уровнемеры фиксируют замедленное и неравномерное по времени и глубине срабатывание подач с регистрацией подстоев и обрывов шихты. На кривой изменения  $\text{CO}_2$  по диаметру (радиусу) колошника содержание диоксида углерода на периферии возрастает, а в осевой зоне снижается. Воздушные фурмы работают с неровной, но интенсивной циркуляцией и с периодическим появлением кусков неподготовленных материалов. Нагрев чугуна и шлака, как правило, повышенный.

Для устранения чрезмерного движения газового потока в центре печи недопустимо в качестве первоочередной меры использовать приемы для закрытия осевой части шихты железорудным материалом. В этом случае может возникнуть подвисяние шихты. Поэтому последовательность действий технологического персонала по ликвидации осевого хода газа состоит в следующем.

Снизить степень подгрузки периферийных участков шихты железорудными материалами путем изменения режима загрузки печи. Это можно сделать, к примеру, увеличением числа подач "коксом вперед" (ККРР ↓ или КРРК↓).

Уменьшить расход дутья для снижения общего перепада давления газов по высоте столба шихты. Если имеется возможность, то повысить давление газа под колошником печи. При появлении первых признаков разгрузки периферии приступить к подгрузке осевой части столба шихты железорудными материалами.

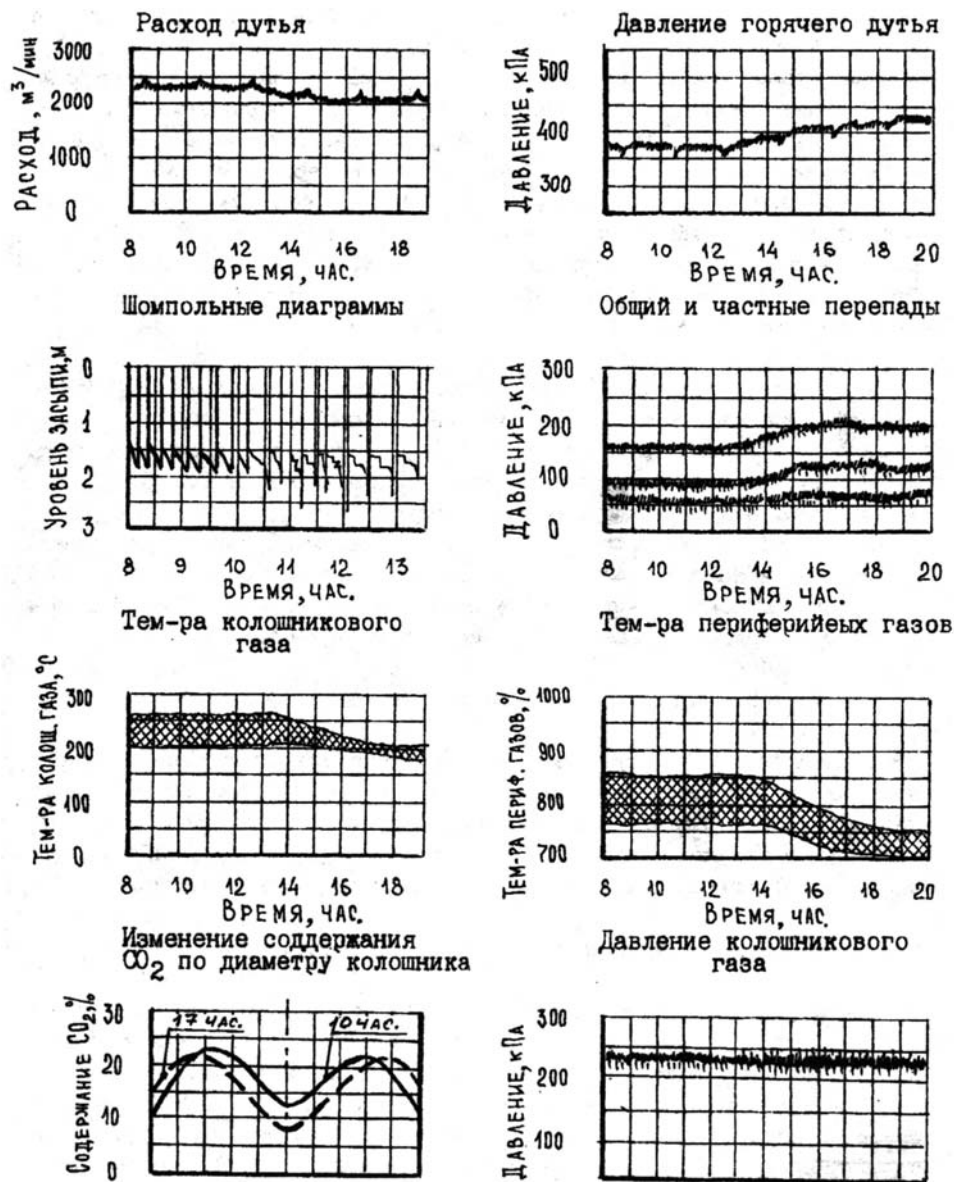


Рис. 8.3. Тренд параметров доменной плавки при развитии центрального потока газа

### Канальный газовый поток

"Канал" – это местный разрыхленный участок в столбе шихтовых материалов, по которому устремляется усиленный поток газов. Этот вид нарушения возникает, как правило, в периферийных участках столба шихты.

При канальном движении газа происходит перевеивание мелких частиц шихты с периферии (вероятного места канала) в центр печи. Это обстоятельство в конечном итоге приводит к еще более значительному нарушению газопроницаемости центральных участков шихты и к частым подвисяниям шихты. Подвисяния шихты и ее обрывы вызывают похолодание печи. Происходит загромождение горна, наступает массовое горение воздушных фурм, что вынуждает де-

лать частые остановки доменной печи. В конечном итоге центральная часть столба шихты становится практически непроницаемой для газов и создается так называемый "химический холодильник".

Причины появления канального газового потока разнообразны: неровный ход доменной плавки, ухудшение качества шихтовых материалов, увеличение мелочи (фракции 0-5 мм) в железорудном материале, снижение механической прочности кокса, несоответствие гранулометрического состава шихты дутьевому режиму плавки, длительная работа доменной печи на воздушных фурмах, частично залитых шлаком, неравномерное распределение дутья по воздушным фурмам, нарушение графика выпусков и т.д.

Тренд основных параметров доменной плавки при канальном газовом потоке представлен на **рис. 8.4**.

Возникновение канального хода газов в печи сопровождается изменением давления горячего дутья – оно становится непостоянным и резко колеблющимся с отклонением от среднего значения в пределах 0,1-0,2 атм (10-20 кПа).

Из-за уменьшения газодинамического сопротивления столба шихты при возникновении канального газового потока расход дутья самопроизвольно возрастает. Показания термопар, регистрирующих температуру колошникового газа в газоотводах, имеют большой разброс (100-200 °С и более). Они перестают пересекаться, при этом среди точек начинают выделяться, смещаясь в область высоких температур, те точки, которые расположены ближе к "каналу".

Нарушается постоянство температур в отдельных точках по периферии печи, увеличивается разность температур периферийных газов, а кривые температур меняют свой температурный уровень и, перемещаясь, могут менять взаимное расположение.

Уровень материалов, фиксируемых зондами, становится непостоянным и различным по сечению печи, а со стороны расположения "канала" появляются провалы зондов: они уходят "без меры".

Давление колошникового газа становится неровным, появляются "верхние пики", возникающие в моменты опускания подач, а также и независимо от этого, вследствие чего на диаграмме исчезают "нижние пики".

Общее содержание  $\text{CO}_2$  в колошниковом газе уменьшается, а на диаграмме содержания  $\text{CO}_2$  по радиусу (диаметру) колошника появляются минимумы (впадины), если плоскость отбора проб газа проходит вблизи канала. Воздушные фурмы в районе канала работают интенсивно, но холодно. На них периодически появляются куски неподготовленного материала. Во время выпуска расплавов из печи наблюдаются значительные колебания нагрева жидких продуктов плавки.

Меры борьбы с канальным ходом газа предусматривают уменьшение загрузки периферийной зоны печи железорудными материалами путем изменения системы загрузки. Постепенно следует увеличивать рудную нагрузку в районе канала путем укладки коксовых гребней по всему оставшемуся периметру колошника. Если принятые меры не приводят к ликвидации канального движения газа, то необходимо осуществить кратковременное глубокое снижение расхода дутья, затем опустить уровень засыпи, догрузить доменную печь коксом до за-

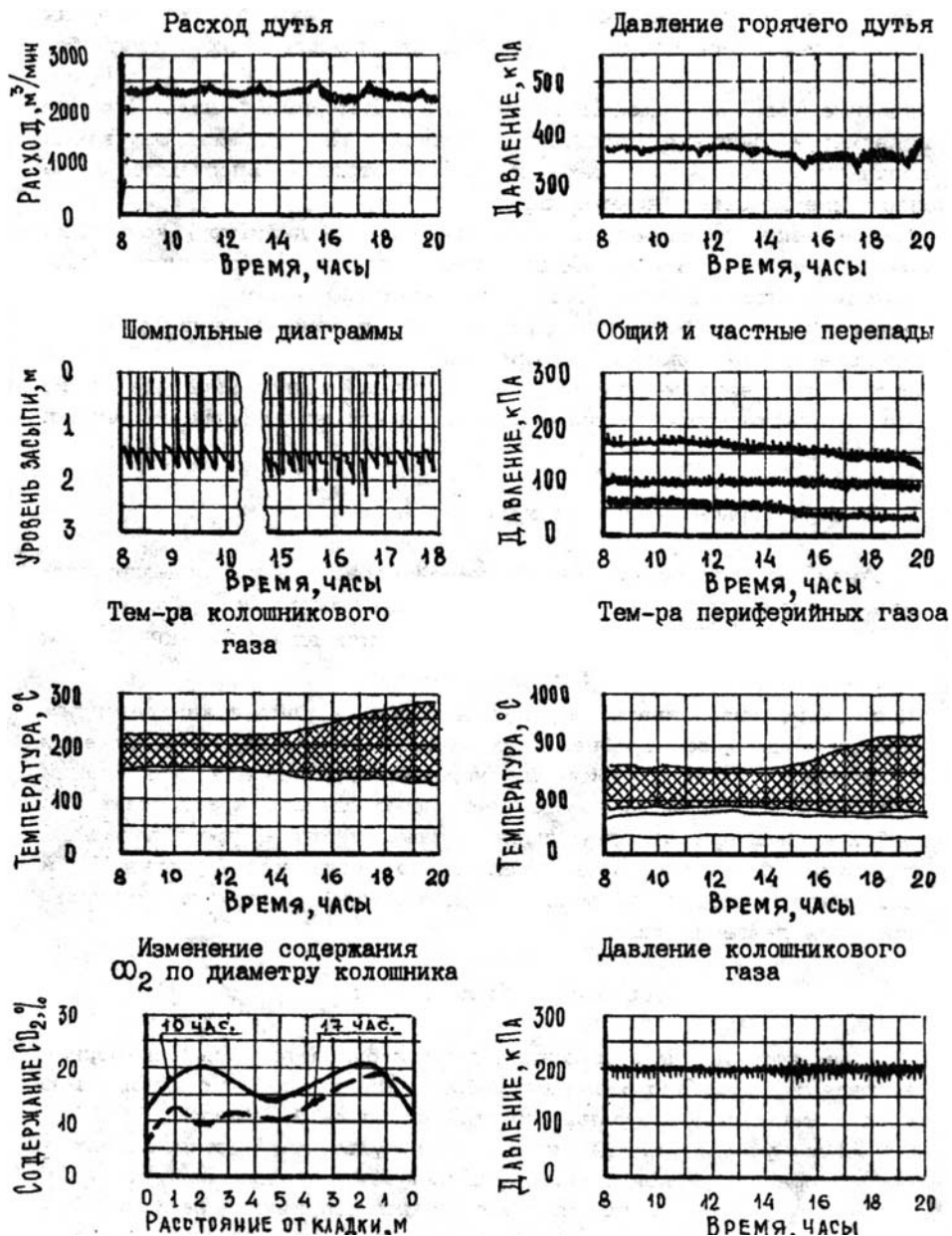


Рис. 8.4. Тренд параметров при развитии канального газового потока

данного уровня и постепенно раздувать печь с пониженным (против обычного) перепадом давления газа.

#### Перекося поверхности шихты на колошнике печи

При перекося шихты в доменной печи наблюдается неравномерная тепловая и восстановительная работа газа по сечению печи, происходит также односторонний разгар огнеупорной кладки.

Перекося уровня шихты является результатом установившегося неравномерного распределения материалов и газов по сечению печи. Это может быть вызвано рядом причин: неправильной центровкой загрузочного устройства, длительной работой печи с неисправной вращающейся воронкой распределителя

шихты (ВРШ), неравномерным распределением дутья и природного газа по воздушным фурмам, заливкой шлаком воздушных фурм и др.

На шомпольных диаграммах фиксируются различные уровни шихты. Увеличивается разность температур в показаниях термопар, регистрирующих температуру периферийных газов. Кривые температур газа в газоотводах перестают перемешиваться и идут обособленными линиями. Ширина ленты разброса температур колошниковога газа значительно увеличивается. Кривые изменения диоксида углерода по радиусу колошника имеют значительные отличия. При водяном охлаждении шахты в местах интенсивного развития газового потока температура выходящей из холодильников воды повышается.

Обычно меры борьбы с перекосом шихты в печи сводятся к следующему. Очистить залитые шлаком воздушные фурмы. Установить углы поворота ВРШ таким образом, чтобы гребни материалов, сформированные на малом и большом конусе, попадали в сторону пониженного уровня шихты. Если вышеприведенные меры не дадут положительного результата, то на выпуске чугуна следует приостановить загрузку печи на 5–10 минут, затем снизить давление дутья (до 0,2–0,3 атм), догрузить печь до нормального уровня и осуществлять медленную раздувку печи.

### *Нарушения теплового состояния доменной плавки*

#### *Признаки и методы борьбы с холодным ходом доменной плавки*

Похолодание печи обусловлено уменьшением прихода тепла или факторами, связанными с дополнительным расходом тепла или снижением степени использования тепловой энергии. Такими причинами могут быть, в частности: ухудшение восстановимости железорудного материала или увеличение содержания в нем мелочи (фракция 0–5 мм), ухудшение качества кокса, применение нерациональной загрузки печи, приводящее к периферийному или каналному газовому потоку, длительная работа печи с пониженным уровнем засыпи, систематическое попадание воды в печь, ошибки при наборе и взвешивании компонентов шихты, длительная работа печи с осадками шихты и др.

Основные признаки холодного хода доменной плавки сводятся к следующему (рис. 8.5).

- Давление горячего дутья плавно уменьшается до определенного предела, а затем резко увеличивается из-за снижения нагрева печи, нарушения режима шлакообразования.
- Первоначально при развитии холодного хода доменной плавки наблюдается снижение нижнего перепада давления газа.
- Расход дутья первоначально имеет тенденцию самопроизвольного повышения, а затем резкого снижения.
- Скорость опускания шихты на колошнике (число срабатываемых подач) первоначально увеличивается из-за самопроизвольного роста расхода дутья и из-за увеличения расхода кокса на прямое восстановление оксидов железа вследствие похолодания и возрастания степени прямого восстановления.

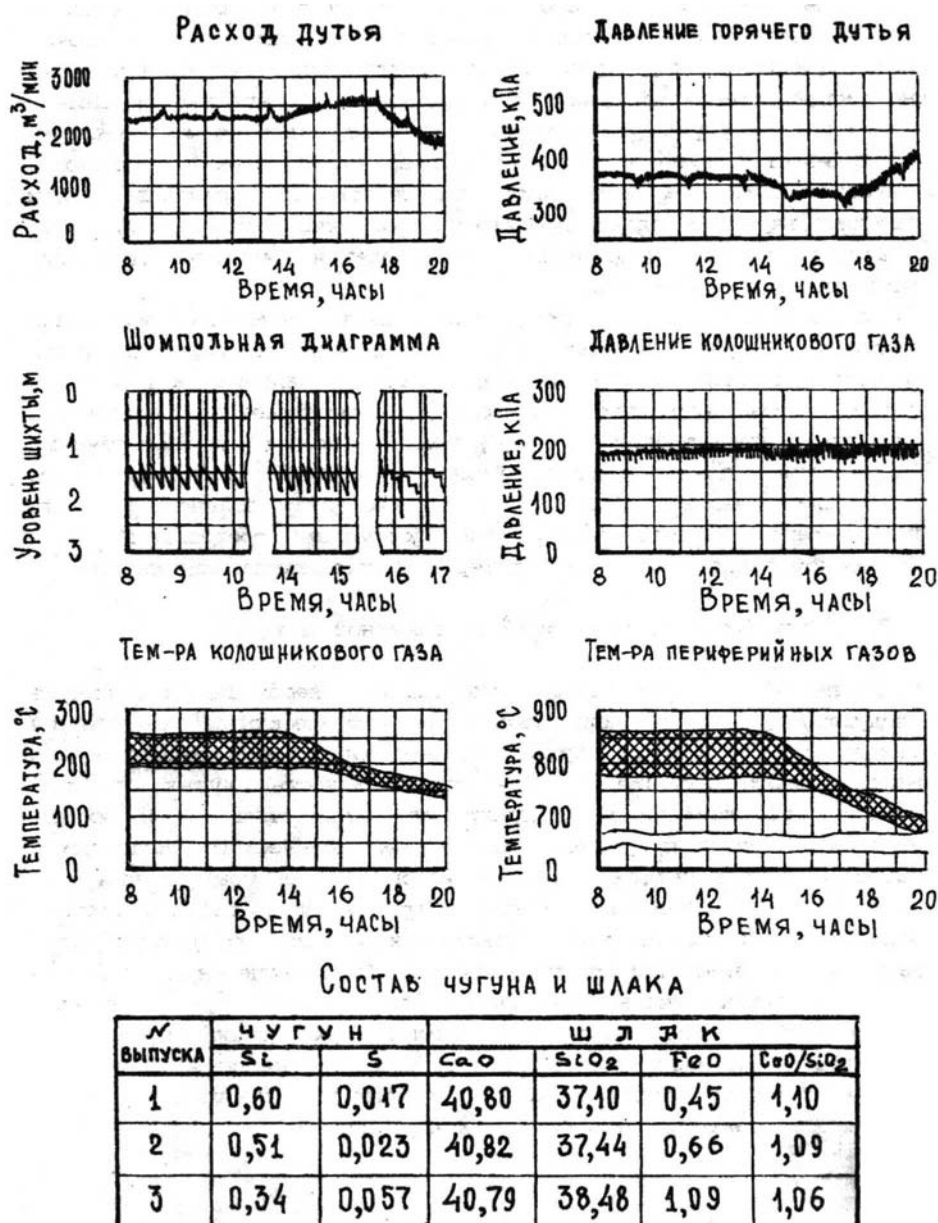


Рис. 8.5. Тренд параметров при похолодании доменной печи

Затем на шомпольных диаграммах фиксируется снижение числа сработанных подач, появление подстоев шихты с последующим ее обрывом.

- Температура колошникового и периферийных газов снижается. Вынос колошниковой пыли вследствие уменьшения температуры газа, снижение его объема и скорости движения сокращаются.
- Нагрев шлака снижается. Понижается его основность, он меняет свой цвет, становится черным вследствие возрастания в нем монооксида железа (FeO). При взятии шлака на "пику" наблюдается образование длинных нитей.
- На воздушных фурмах периодически появляются неподготовленные шихтовые материалы, состоящие из невосстановленных оксидов железа, первичного шлакового расплава и мелких кусков кокса.

- Содержание кремния и марганца в чугуна уменьшается, а содержание серы возрастает. Увеличивается густота искр над поверхностью зеркала чугуна и уменьшается высота этих искр над поверхностью расплава. При выпуске верхнего шлака наблюдается наличие мелких искр, что указывает на ухудшение процесса разделения металла и шлака из-за роста его вязкости.
- При дальнейшем развитии похолодания шихтовые материалы начинают непрерывно подвисать, обрываясь на ходу доменной плавки. В период подвисяний очень опасно появление на воздушных фурмах железистого шлака, поскольку увеличивается вероятность заливки фурм шлаком с прогаром сопел и с неизбежными остановками доменной печи. Главной заботой технологического персонала, ведущего доменную плавку, должно быть предотвращение подвисяний всеми доступными средствами, в частности ценой большой временной потери производительности печи.

При первых признаках похолодания необходимо принять следующие меры:

- Если имеется возможность повышения температуры горячего дутья, то необходимо ее увеличение до максимального значения.
- Уменьшить влажность дутья вплоть до полного прекращения подачи пара.
- Уменьшить расход природного газа и технологического кислорода с расчетом выхода значений теоретической температуры горения не выше 2200 °С.
- Уменьшить рудную нагрузку путем добавки кокса в подачу. Если похолодание развивается дальше, то целесообразно дать кокс в виде "холостых" подач.
- Для предотвращения холодных подвисяний запрещается изменение системы загрузки, уровня засыпи и иные приемы регулирования хода доменной плавки "сверху". Запрещается также увеличивать расход известняка.

#### *Признаки и меры борьбы с горячим ходом доменной плавки*

Причины излишнего нагрева печи прямо противоположны похолоданию доменной плавки. Отклонение доменной плавки в сторону разогрева менее опасно, чем похолодание. Но, тем не менее, – это нарушение нормального режима плавки, которое приводит к уменьшению производительности печи и росту удельного расхода топлива.

Основные признаки разогрева доменной печи сводятся к следующему (рис. 8.6):

- Постепенно повышаются общий и нижний перепады давления газов, а также и давление горячего дутья.
- Сход шихтовых материалов в печи замедляется, и если не принимать мер к снижению перепада давления газа, то появляются обрывы шихты и неустойчивое положение уровня шихты в печи.

- На диаграмме давления колошникового газа появляются "верхние пики", что является признаком прорывов газа через слой шихты в моменты ее подвиганий и обрывов.

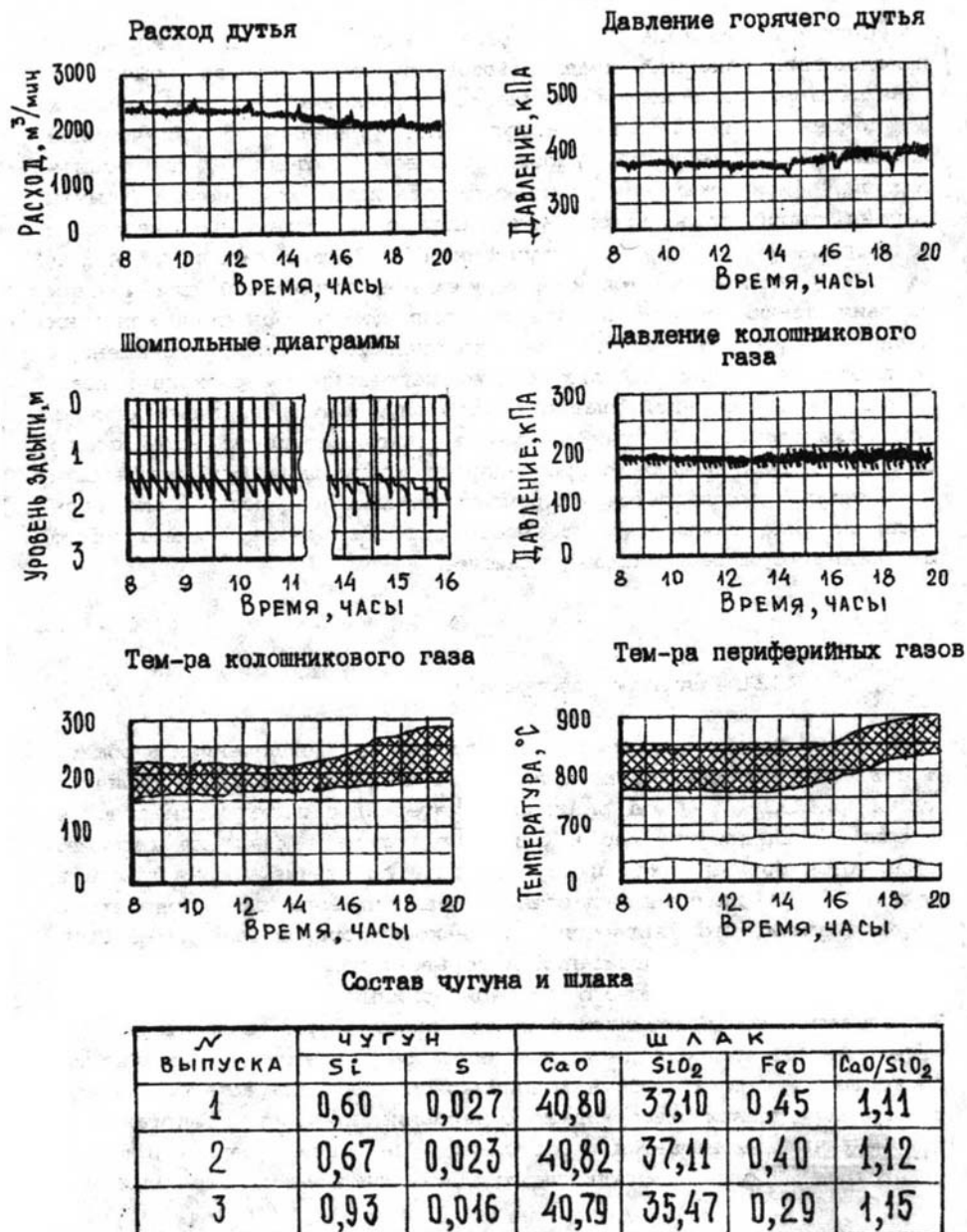


Рис. 8.6. Тренд параметров при разогреве доменной печи

- Температуры колошникового и периферийных газов увеличиваются. Яркость свечения фурм возрастает, но интенсивность циркуляции кусков кокса замедляется. Содержание диоксида углерода (CO<sub>2</sub>) в колошниковом газе возрастает.
- Возрастает содержание кремния и марганца в чугуне, уменьшается содержание в нем серы, возрастает температура и основность шлака.



При признаках разогрева необходимо принять следующие меры:

- Если нарушение плавки обнаружено своевременно, то для ликвидации разогрева печи достаточно ограничиться увеличением влажности дутья до установленного верхнего предела. Это является полезным и для предотвращения подвисаний шихты.
- Если увеличение влажности дутья недостаточно, то для ликвидации разогрева печи рекомендуется временное снижение температуры дутья на 50°C и более с постепенным ее восстановлением по мере исчезновения признаков подвисания.
- Если установлено, что действие фактора, вызвавшее разогрев печи, имеет длительный характер, то необходимо увеличить рудную нагрузку.

### *Нарушение ровного схода шихты в доменной печи*

#### *Подвисание шихтовых материалов в доменной печи (рис. 8.7)*

Подвисанием шихты в доменной печи называют прекращение ее движения (схода) или временную задержку с последующим возобновлением движения. Подвисание шихты – одно из самых распространенных видов расстройства доменной плавки, часто имеющих тяжелые последствия (потеря производительности печи, выпуск бракованной продукции). Тяжелые случаи подвисаний могут быть причиной повреждения отдельных конструкций доменной печи. Подвисание шихты является следствием нарушения нормального распределения материалов и газов в печи, теплового состояния и шлакового режима плавки вследствие ухудшения качества железорудных материалов и кокса, перегрузки периферийной зоны печи рудной составляющей, чрезмерной основности шлака. Это может быть следствием разогрева или значительного похолодания доменной плавки, загромождения горна, а также развития канального газового потока. Подвисанию материалов в печи обычно предшествует неровный ход печи. По своему характеру подвисания разделяются на "верхние" и "нижние".

#### *Верхние подвисания шихты в доменной печи*

Эти подвисания, как правило, связаны с увеличением мелочи в железорудных материалах, а также с ухудшением механической прочности кокса.

Признаками верхнего подвисания являются (см. **рис. 8.7, а**):

- возрастание верхнего и общего перепадов давления газа;
- резкое возрастание давления горячего дутья на 0,1-0,3 атм (10-30 кПа) против нормального;
- резкое замедление или полное прекращение схода шихты после нескольких обрывов;
- увеличение содержания диоксида углерода в колошниковом газе;
- температура газов в газоотводах первоначально снижается, затем медленно растет;

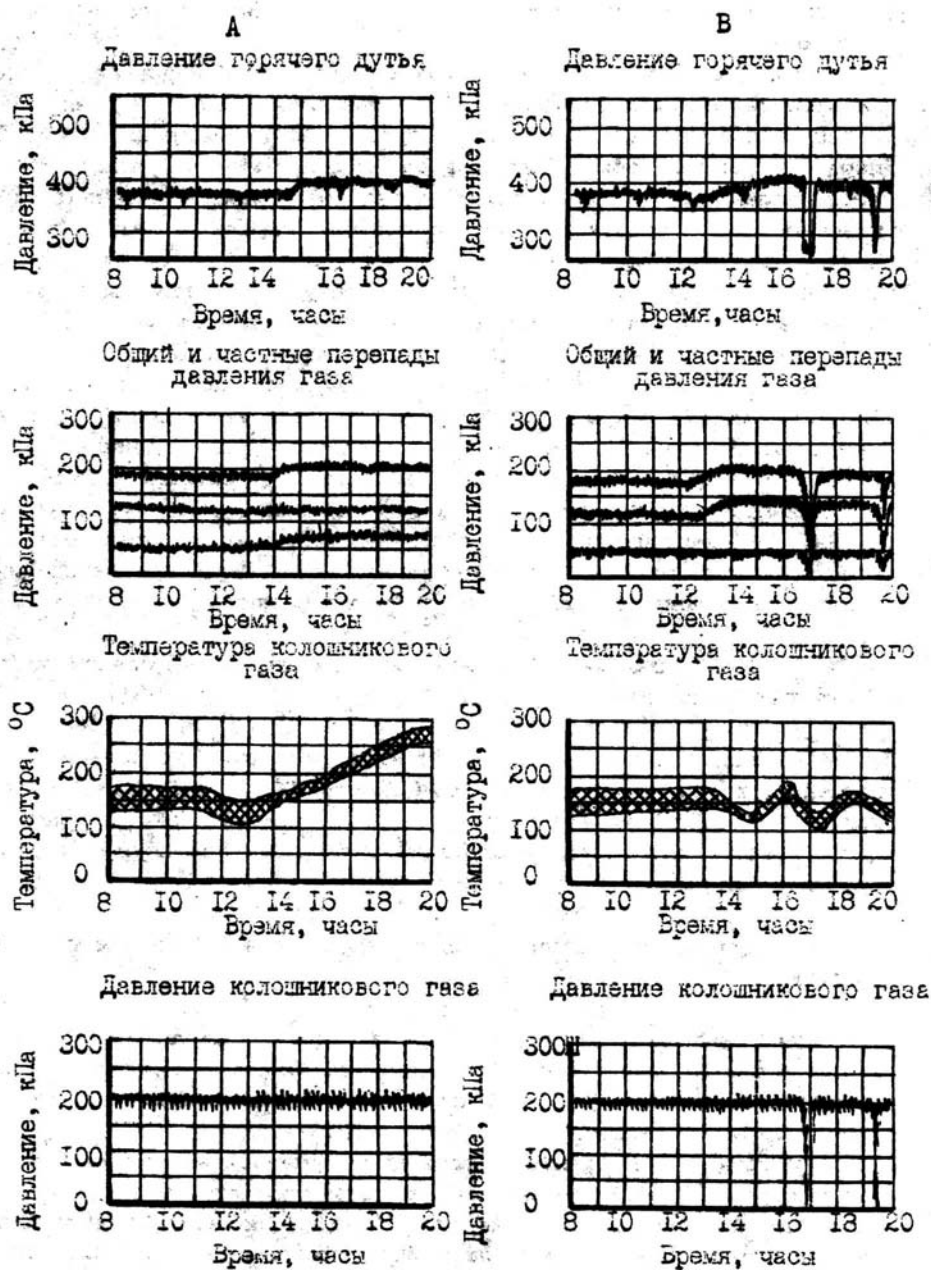


Рис. 8.7. Тренд параметров при верхнем (а), нижнем (б) подвисании шахты

- увеличение температуры и уменьшение разброса показаний периферийных термопар;
- воздушные фурмы работают интенсивно и в большинстве случаев равномерно.

Меры по ликвидации верхнего подвисяния шихты заключаются в следующем:

- снизить температуру горячего дутья на 50–100 °С;
- уменьшить расход дутья для установления давления горячего дутья на 0,1 атм (10 кПа) меньше по сравнению с его давлением до момента подвисяния;

- если несмотря на принятые меры сход шихтовых материалов не возобновляется, то через 10–15 минут от начала подвисяния сделать искусственную осадку шихты;
- после восстановления схода шихты в печи следует постепенно восстановить количество дутья, поддерживая общий перепад давления газов на 0,1–0,2 атм (10–20 кПа) ниже рабочего.

#### *Нижние подвисяния шихты в доменной печи*

Нижние подвисяния чаще всего являются следствием нарушения теплового и шлакового режимов, в результате чего происходит ухудшение газопроницаемости зоны шлакообразования вследствие потери подвижности шлака, возрастания объема зоны вязкопластичных масс и образования неблагоприятной формы зоны когезии.

Признаки нижнего подвисяния шихты в доменной печи следующие (см. **рис. 8.7, б**):

- повышается общий и нижний перепад давления газов. Медленно повышается давление горячего дутья в течение 1–1,5 часа перед подвисянием;
- шомпольные диаграммы регистрируют многочисленные подстои шихты с последующей осадкой материалов. В конечном итоге происходит полное прекращение схода материалов в печи;
- на диаграмме давления колошникового газа появляются "верхние пики";
- кривая температуры колошникового газа имеет вид узкой зигзагообразной ленты;
- воздушные фурмы во время нижних подвисяний работают вяло и неравномерно;
- движение шихты прекращается после одного или нескольких обрывов, при этом давление дутья быстро возрастает до величины на 0,2–0,3 атм (20–30 кПа) выше нормального;

В зависимости от нагрева печи подвисяние может быть на горячем и холодном ходу доменной плавки.

Ликвидации нижних подвисяний на горячем ходу сводится к следующему:

- уменьшить расход дутья с таким расчетом, чтобы снизить давление горячего дутья на 0,1–0,2 атм (10–20 кПа) ниже того, которое было до момента подвисяния;
- увеличить влажность дутья;
- изменить систему загрузки для развития периферийного газового потока.

Холодные подвисяния являются наиболее тяжелыми видами расстройства хода печи. Поэтому необходимо прилагать все усилия для того, чтобы заблаговременно при появлении первых признаков этого расстройства предотвратить его.

Ликвидация нижних подвисаний на холодном ходу сводится к следующему:

- при первых же признаках необходимо уменьшить расход дутья таким образом, чтобы уменьшить давление горячего дутья на 0,1 атм (10 кПа) против того, которое было до момента подвисания. Уменьшение расхода дутья должно быть произведено в максимально короткий срок;
- загрузить в доменную печь несколько холостых подач в зависимости от теплового состояния печи;
- снизить рудную нагрузку;
- если несмотря на принятые меры возобновление схода шихты не происходит, то через 15-20 минут сделать осадку шихты, непрерывно наблюдая за состоянием воздушных фурм, не допуская заливки их шлаком;
- при упорных подвисаниях шихты загрузить 5-8 холостых подач, поддерживая сниженный расход дутья. После повышения нагрева печи и появления устойчивого схода шихты постепенно увеличивать расход дутья, поддерживая перепад давления по высоте слоя шихты на 0,1-0,2 атм (10-20 кПа) ниже рабочего значения.

#### *Тугой ход доменной печи*

Чрезмерное сокращение периферийного потока газов при недостаточной газопроницаемости осевой зоны обычно становится причиной неустойчивой работы доменной печи, склонной к подвисаниям и подстоям. Подобное нерациональное газораспределение является результатом работы печи с пониженным уровнем засыпи, применения излишне больших подач, нерациональной системы загрузки, загрузки в печь шихтовых материалов с большим количеством мелочи.

Признаки тугого хода доменной печи сводятся к следующему (**рис. 8.8**):

- давление горячего дутья повышается и становится неустойчивым;
- расход дутья уменьшается;
- давление колошникового газа становится неустойчивым, на диаграмме появляются "верхние пики";
- возрастает общий перепад давления, а диапазон колебаний перепада давления увеличивается;
- кривые температуры газа по газоотводам сходятся в узкую ленту, отдельные точки практически накладываются друг на друга. Температура периферийных газов понижается;
- сход шихты становится неравномерным, имеют место подстои шихты и провалы шомполов;
- содержание диоксида углерода как на периферии, так и в осевой части печи на 2-4 % выше оптимального;
- интенсивность циркуляции кусков кокса на воздушных фурмах уменьшается, и фурмы работают "вяло".

Для восстановления нормального хода печи необходимо перейти на систему загрузки, разгружающую периферию без увеличения нагрузки на осевую зону столба шихты.

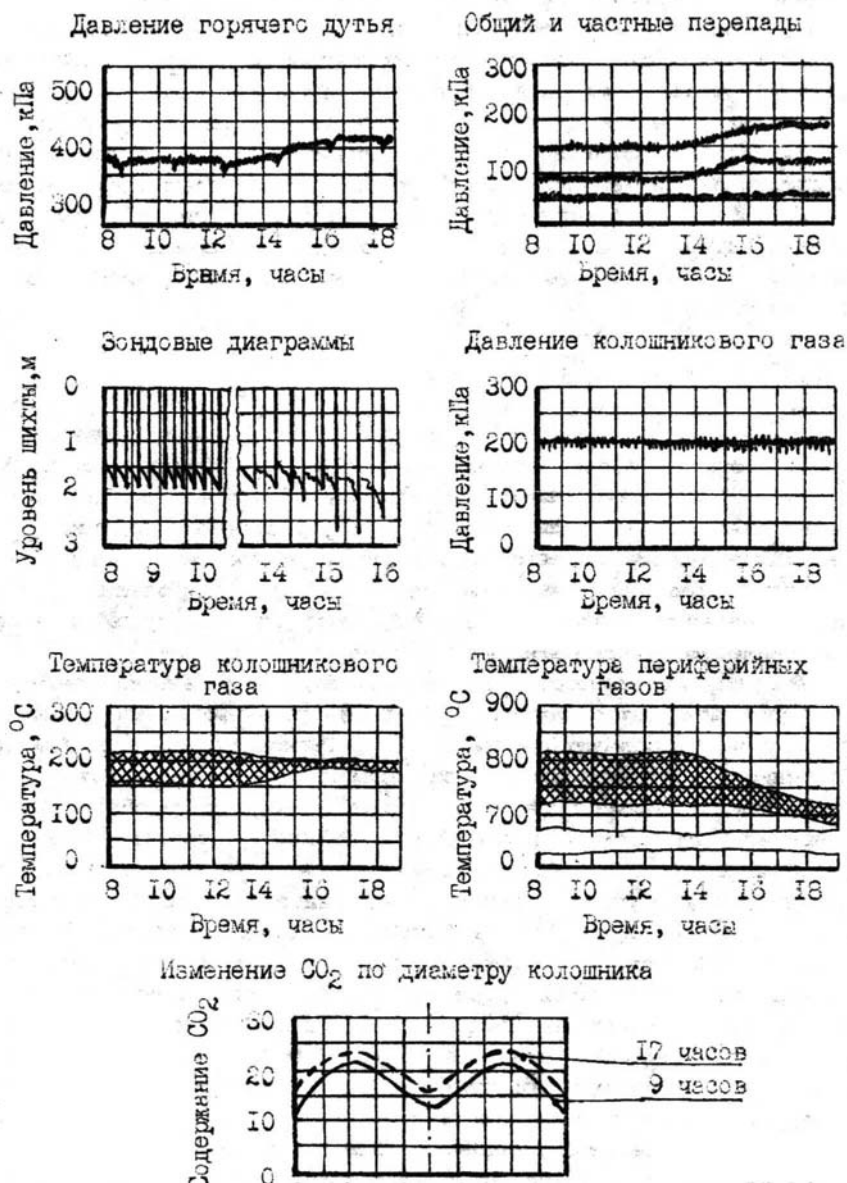


Рис. 8.8. Тренд параметров при тугом ходе доменной печи

### 8.3. Общая характеристика ЭС «Интеллект доменщика»

Учитывая крупные достижения в последнее время в области искусственного интеллекта в решении задач диагностики и прогнозирования, с одной стороны, и появление мощных систем реального времени, с другой, появилась возможность создания по сути дела автоматизированной системы управления доменным процессом нового поколения на базе экспертных систем.

В общем случае такая система должна иметь «навыки суждений» о ходе доменной печи по различным признакам, правильно анализировать их и свое-

временно предупреждать вероятные отклонения от нормального режима, применяя различные приемы воздействия на ход доменной плавки.

Предварительно отметим *основные группы параметров, характеризующих ход доменной печи и используемых в ЭС:*

- химический состав чугуна и шлака, температура чугуна;
- информация с технических средств контроля хода доменной плавки;
- состояние фурменных зон;
- состояние горна доменной печи и, в частности, уровень расплава.

*Основные виды расстройств доменной печи, которые система диагностирует на базе искусственного интеллекта:*

- периферийный ход печи;
- осевой ход печи;
- канальный ход печи;
- перекосящий уровень засыпи;
- тугий ход печи с дальнейшим развитием верхнего или нижнего подвисания;
- горячий ход печи;
- холодный ход печи;
- аварийные ситуации (загромождение горна, образование настывлей и т.п.).

Кроме того, экспертная система непрерывно определяет состояние ровного хода печи и заданного нагрева и прогнозирует их развитие или нарушение.

В процессе экспертизы хода доменной плавки также выявляются *факторы, являющиеся причиной нарушений хода доменной плавки.*

К ним относятся:

- изменение свойств шихтовых материалов (качество железорудных материалов, содержание мелочи в шихте, прочность кокса);
- изменение параметров дутьевого режима (расход дутья, температура и влажность);
- изменение режимов загрузки (система загрузки, уровень засыпи);
- изменение вязкости шлака;
- оптимальность рудной нагрузки;
- изменение параметров комбинированного дутья (расход инжектируемого топлива (природного газа), технологического кислорода на обогащение дутья);
- состояние горна (уровень жидких продуктов плавки).

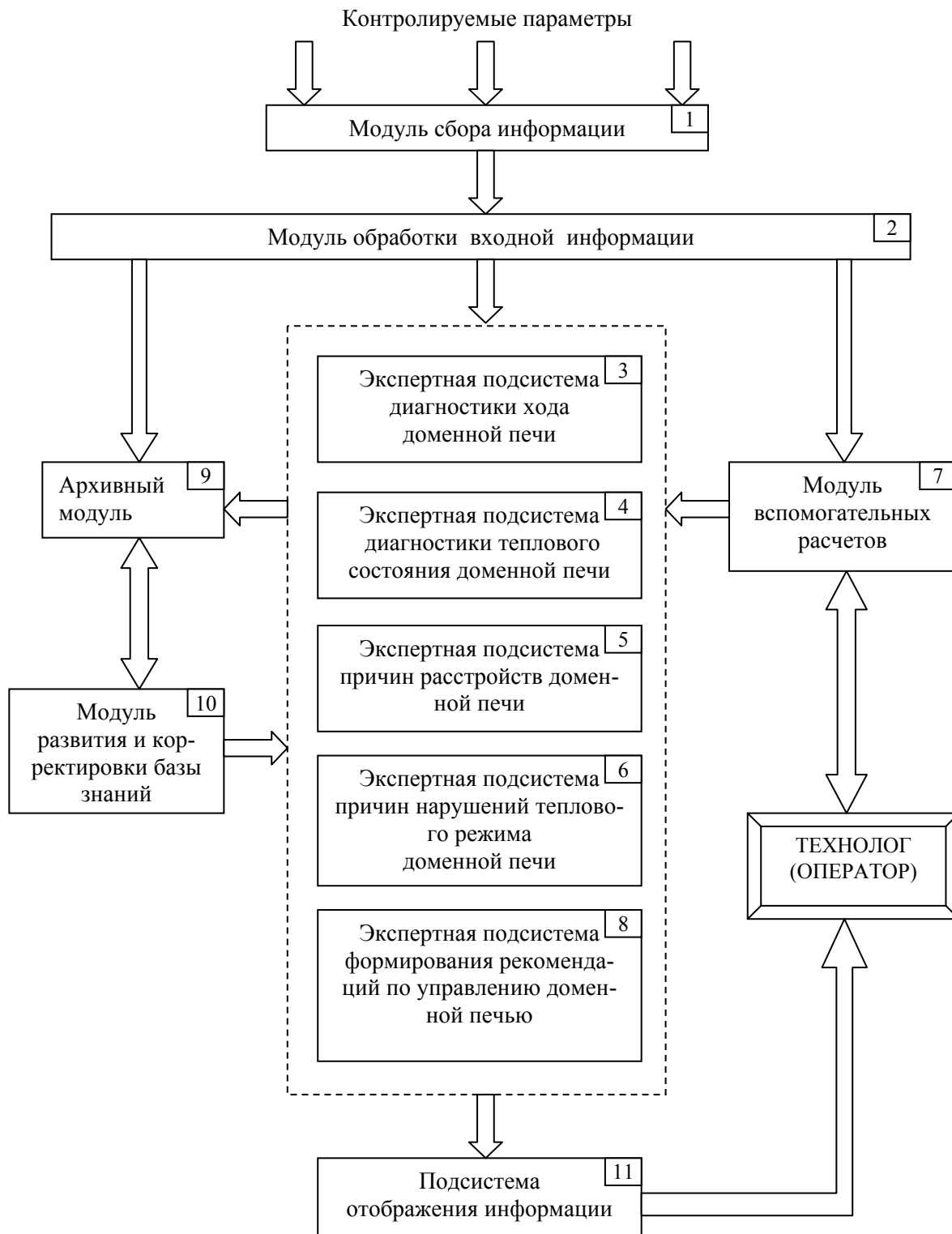
*Основные способы воздействия на ход доменной плавки с целью устранения возникающих отклонений сводятся к следующему:*

- изменение рудной нагрузки (в том числе загрузка в печь дополнительных скипов кокса и холостых подач);
- изменение режима загрузки материалов (система загрузки и уровень засыпи);
- изменение дутьевых параметров (расход, температура и влажность дутья, расход природного газа и технологического кислорода);
- изменение в подачу масс основных и кислых добавок;

- изменение давления колошникового газа;
- изменение количества и состава шлака;
- искусственные (принудительные) осадки шихты.

Управляющие воздействия реализуются обычно путем выдачи рекомендаций технологическому персоналу на их изменение.

Функциональная структура экспертной системы представлена на **рис. 8.9** и включает следующие подсистемы.



**Рис. 8.9.** Функциональная структура экспертной системы

1. *Модуль сбора информации* о параметрах доменной плавки. Этот модуль считывает данные, поступающие в режиме реального времени, генерируемые как датчиками технологических параметров, так и вводимые вручную технологическим персоналом доменной печи. Цикл считывания сигналов датчиков составляет 5–30 секунд.
2. *Модуль обработки входной информации*. Производит оценку считываемых значений параметров (сглаживание, интерполяцию) и выделение характерных изменений параметров на временном интервале 8 часов по специальным математическим процедурам.
3. *Экспертная подсистема диагностики хода доменной печи*. Представляет программный модуль, предназначенный для раннего обнаружения намечающихся расстройств доменной печи (канального хода, периферийного хода, тугого хода с дальнейшим прогнозом места подвисяния шихты, горячего и холодного хода). Экспертная система обеспечивает возможность идентификации ровного хода печи и диагностирует его развитие или нарушение. Экспертная система выдает в количественной форме вероятность появления и развития того или иного вида расстройств хода доменной печи.
4. *Экспертная подсистема диагностики теплового состояния доменной печи*. Представляет программный продукт, обеспечивающий непрерывную диагностику теплового состояния доменной печи по показаниям технических средств контроля. Она выдает с учетом хода, определяемого системой по п.3, количественные данные вероятности того или иного теплового состояния печи (похолодания, разогрева и заданного нагрева печи).
5. *Экспертная подсистема причин расстройств хода доменной печи*. Представляет собой программное средство, предназначенное для информирования технологического персонала о причинах, вызвавших отклонение от нормального хода печи.
6. *Экспертная подсистема причин нарушений теплового режима доменной печи*. Представляет собой программное средство для определения факторов, вызывающих нарушение теплового режима доменной плавки.
7. *Модуль вспомогательных расчетов* - является программным модулем и предназначен для произведения расчетов, необходимых технологическому персоналу при управлении доменным производством. В состав этого модуля входят, в частности, следующие программы:
  - проверочный расчет шихты;
  - расчет задувочной шихты после стоянки печи различной продолжительности;
  - расчет теоретической температуры горения;
  - оперативный расчет расходов технологического кислорода, естественного и обогащенного дутья и др.
8. *Экспертная подсистема формирования рекомендаций по управляющим воздействиям* предназначен для выдачи рекомендаций технологическому персоналу по применению одного из нескольких способов



воздействия на ход доменной печи в случае обнаружения возникающего отклонения от нормального хода или наметившегося отклонения теплового состояния печи от заданного. При этом предусмотрена возможность регулирования теплового режима доменной печи как снизу (температура и влажность, расход природного газа и кислорода) при кратковременном (до 8 часов) нарушении, так и сверху (рудная нагрузка, режим загрузки) при длительных (более 8 часов и до нескольких суток) прогнозируемых отклонениях от заданного значения содержания кремния в чугуне.

9. *Архивный модуль* представляет собой программное средство, обеспечивающее хранение непрерывных реализаций трендов параметров состояния, основных контролируемых параметров, расчетных и фактических значений управляющих воздействий. Этот модуль также сохраняет результаты работы подсистем хода и теплового состояния доменной печи.

10. *Модуль развития и корректировки базы знаний*. Является программным модулем и предназначен:

- для включения новых порождающих правил;
- для установки и корректировки коэффициентов уверенности для порождающих правил;
- для тестирования базы знаний на контрольных сценариях развития расстройств печи.

11. *Подсистема отображения информации*. Модуль предназначен для представления технологическому персоналу необходимой информации о ходе доменной плавки. При этом поддерживается два режима:

- автоматический (постоянный вывод информации на монитор компьютера);
- диалоговый – по запросу пользователя.

Система отображения обеспечивает:

- текущее значение вероятностей появления и развития того или иного хода доменной печи и ее теплового состояния;
- графическое представление текущей тенденции развития вероятностей того или иного хода печи за последние 24 часа;
- необходимые разъяснения по заключениям экспертных систем по запросу технологического персонала;
- вывод на экран монитора экстренных сообщений (перекос уровня засыпи, отклонение теоретической температуры горения от заданной, неисправность датчиков, прекращение поступления данных в систему и т.п.);
- графическое представление изменения теоретической температуры горения;
- графическое отображение газодинамических характеристик столба шихты (текущих и предельных).

## 8.4. Обработка входной информации

В основу алгоритмов обработки входной информации положен подход, при котором сигнал описывается:

- диапазоном его изменения (допустимое значение, больше или меньше допустимого значения);
- тенденцией изменения сигнала (увеличивается, уменьшается, остается неизменным, низкочастотные колебания);
- амплитудой высокочастотных колебаний (равна установленному значению, больше установленного значения).

Считывание информации с датчиков производится дискретно с периодом  $\Delta T$ , определяемым, с одной стороны, временем обработки всех датчиков, а с другой – из условия  $\Delta T < 1/2F$ , где  $F$  – низкочастотная составляющая полезного сигнала.

Входная информация разделена на специфические блоки.

1. *Блок задач быстрых входных данных.* В этот блок включены параметры, требующие периодичности опроса состояния 5 секунд (критические параметры), такие как верхние пики давления колошникового газа, анализ схода шихтовых материалов, время срабатывания подач, перепады давления по высоте печи, давление колошникового газа, горячего дутья, работа ВРШ, дискретные сигналы и т.д.

2. *Блок задач остальных входных данных, автоматически контролируемых непосредственно на печи:* анализ состава колошникового газа, температуры периферийных термопар, колошникового газа по газоотводам, влажность, температура, расход и состав горячего дутья, расходы природного газа и технологического кислорода, расходы кокса и железорудных материалов и т.п. Периодичность опроса состояния этих остальных автоматически контролируемых параметров составляет 30 секунд.

3. *Автоматический или полуавтоматический ввод контролируемых данных:* химический состав железорудного сырья, его механическая прочность, содержание мелочи; технический анализ кокса, его прочность; состав жидких продуктов плавки. Поступающая из отдела сырья и топлива информация о химическом составе и физических свойствах очередной партии сырья обновляется по мере поступления. При этом производится предварительная обработка информации, слежение за прохождением партии сырья и передача информации о составе сырья в компьютер экспертной системы, когда данная партия сырья подается на колошник доменной печи. Поступающая из химической лаборатории доменного цеха информация о результатах химического анализа чугуна и шлака становится известной примерно через 2 часа после очередного выпуска. В экспертную систему передается также оперативная информация о содержании кремния в чугуне и основности шлака на выпусках по оценке мастера печи.

4. *Блок информации о ходе доменного процесса по оценке технолога-доменщика.* К ней, в первую очередь, относится та ее часть, которая в настоящее время не контролируется автоматически, но несет большую информационную нагрузку о развитии процессов в доменной печи. К этой информации относятся следующие данные:

- состояние фурменных зон, оцениваемое на основании визуального контроля: яркость и неравномерность свечения фурменной зоны, циркуляция кокса перед фурмами и т.п.;
- состояние горна доменной печи;
- оценка внешнего вида чугуна и шлака в жидком и твердом состояниях.

5. Блок определения уровня расплава в горне и характера выпусков чугуна и шлака. Информация определяется на основании данных начала и конца выпуска, времени срабатывания подач, массы выпуска чугуна и визуального наблюдения за ходом отработки жидких продуктов плавки.

Информация блоков 4 и 5: данные визуального контроля работы фурменной зоны, программа системы загрузки, начала и конца отработки жидких продуктов плавки и т.п. обновляются по мере поступления и ввода информации.

Обновление выходной информации на экране монитора (результат диагностирования состояния) происходит с периодичностью 5 минут, а при поступлении экстренных сообщений – сразу. В процессе обработки параметров доменной плавки возникает необходимость:

- выделения различных характеристик их изменений: увеличение, уменьшение, наличие колебаний, наличие положительных и отрицательных пиков на определенном отрезке времени;
- фильтрации, сглаживания, интерполяции и экстраполяции.

Для автоматических датчиков в основном производится периодическое считывание сигналов, их предварительная обработка по различным алгоритмам, а также запоминание оценок параметров обрабатываемых сигналов за 5-минутный интервал времени.

### *Алгоритмы сглаживания технологической информации*

Первичная обработка данных контроля и показаний измерительных приборов включает процедуру экспоненциального сглаживания первого или второго порядка, фильтрацию методом медианы по 3–5 точкам.

Рассмотрим основные математические процедуры, обеспечивающие решение указанных задач при обработке информации о состоянии технологического агрегата.

**Алгоритм сглаживания среднего.** Алгоритм сглаживания среднего представляется в виде

$$Z^*(i) = \frac{1}{M} \sum_{j=i-M+1}^i Z(j), \quad (8.1)$$

где  $Z^*(i)$  – сглаженное значение контролируемой переменной  $Z(i)$  для  $i$ -го момента времени;

$M$  – интервал сглаживания («конечная память»), выраженная в числе точек.

Величина  $M$  может выбираться из условия  $M < M_{\min}^s$ , где  $M_{\min}^s$  – минимальная длительность постоянного уровня полезного сигнала. Такой выбор оправдан с точки зрения получения в конце переходного процесса практически точного отслеживания любого ступенчатого изменения полезного сигнала.

**Алгоритм экспоненциального сглаживания первого порядка.** Алгоритм экспоненциального сглаживания первого порядка имеет вид

$$Z^*(i) = Z^*(i-1) + \alpha[Z(i) - Z^*(i-1)]$$

или

$$Z^*(i) = (1 - \alpha)Z^*(i-1) + \alpha Z(i). \quad (8.2)$$

Параметр сглаживания  $\alpha$  выбирается так, чтобы длительность переходного режима была менее  $M_{\min}$ , т.е.  $\alpha = 2/(M+1)$ ,  $M \leq M_{\min}$ , где  $M$  – интервал сглаживания («конечная память»), выраженная в числе точек;

$i$  – порядковый номер цикла (интервала) считывания;

$Z^*(i)$  – сглаженное значение сигнала на  $i$ -м считывании;

$Z(i)$  – фактически считанное значение сигнала на  $i$ -м считывании.

Как видно из выражения (8.2), экспоненциальное сглаживание не пропускает ни высокочастотную составляющую, ни отдельные кратковременные (меньше цикла считывания) импульсы считываемого сигнала.

**Алгоритм релейного экспоненциального сглаживания.** Алгоритм релейного экспоненциального сглаживания имеет вид

$$Z^*(i) = Z^*(i-1) + \alpha \varphi[Z(i) - Z(i-1)] = Z(i-1) + \alpha \varphi(X), \quad (8.3)$$

в котором функция  $\varphi(x)$  принимает значения

$$\varphi(x) = \begin{cases} x, & \text{если } |x| \leq \gamma, \\ \gamma \text{Sign}(x), & \text{если } |x| > \gamma, \end{cases} \quad (8.4)$$

параметр  $\alpha$  выбирается, как и для алгоритма (8.2);

$\gamma > 0$  – некоторый параметр, определяющий свойства защищенности алгоритма (8.3) от влияния грубых помех. Если обычная помеха  $\{N^H(i)\}$  имеет нормальное распределение с нулевым средним и дисперсией  $\sigma_{N^H}^2$ , то параметр  $\gamma$  можно выбирать в виде

$$\gamma = \Delta^S + 2,5 \sigma_{N^H}, \quad (8.5)$$

где  $\Delta^S$  – модуль вероятного превращения полезного сигнала на соседних отсчетах. Как видно из формул (8.3) и (8.4), фильтр релейного экспоненциального сглаживания не пропускает высокочастотную составляющую с амплитудой  $< \gamma$  и пропускает без запаздывания изменение сигнала, если это приращение больше  $\gamma$ .

Для исключения пропускания отдельных выбросов (пиков) сигнала или определения появления устойчивого приращения при оценке  $Z(i)$  целесообразно выбирать либо из надежных физических представлений, либо путем обработки нескольких, например трех-пяти последовательных, отсчетов наблюдаемой переменной. Сигнал об устойчивом изменении параметра  $B$  формируется из условия

$$B_{\text{sign}}(Z) = \begin{cases} 1, & \text{если } Z^*(i), Z^*(i+1), Z^*(i+2) \geq \Theta, \\ 0, & \text{в остальных случаях.} \end{cases} \quad (8.6)$$

При неизменяемой переменной

$$B(Z) = \begin{cases} 1, & \text{если } Z^*(i), Z^*(i+1), Z^*(i+2) < \Theta, \\ 0, & \text{в остальных случаях.} \end{cases} \quad (8.7)$$

**Алгоритм медианного сглаживания.** Алгоритм медианного сглаживания (med) представлен выражением

$$Z^*(i) = \text{med} \{Z_{i-M-1}, Z_{i-M}, \dots, Z_i\} \quad (8.8)$$

с конечной «памятью» сглаживания, равной числу точек  $M$ .

Для решения проблемы выделения «полочек» и знаков тренда изменения измеряемых параметров остановимся на задаче выделения их на стационарных и локально-стационарных участках, описываемых выражением

$$X(t^* \pm \Theta) = X^d(t^*) + n(t^* \pm \Theta) \quad \text{при } |\Theta| \leq \frac{T^*}{2}, \quad (8.9)$$

где  $X$  – измеренная переменная;  $t^*$  – дискретное или непрерывное время, отнесенное к середине участка длительностью  $T^*$ ;  $\Theta$  – переменный сдвиг по времени вокруг центра  $t^*$ ;  $X^d$  – действительное значение полезного сигнала, характеризующего уровень полочки;  $n$  – измерительная помеха, эквивалентная обычным и грубым искажениям полезного сигнала.

При решении задачи выделения «полочек», участков с положительным и отрицательным трендом может быть использовано релейное экспоненциальное сглаживание первого порядка, в котором базовые значения  $\alpha^*$ ,  $\beta^*$  настроечных коэффициентов следует выбирать по приближенным соотношениям

$$\alpha^* = \frac{4\Delta_t}{T^*}; \quad \beta = 2,5\sigma_n, \quad \text{т.е. в зависимости от длительности участка } T^*,$$

среднеквадратичного отклонения обычной помехи  $\sigma_n$  и шага дискретизации времени  $\Delta_t$  при считывании сглаживаемых данных. При этом оперативно рассчитывается текущее среднее знаковой последовательности отклонений между  $x(t)$  и выдаваемой релейным экспоненциальным сглаживанием оценкой  $x^d(t)$ .

Такой расчет делается рекуррентно:

$$S_g^* = S_g^*(t-1) + a_{S_g} \{ \text{sing} [x(t) - x^d(t)] - S_g^*(t-1) \}, \quad (8.10)$$

где  $S_g^*$  – экспоненциальная оценка при  $1 > \alpha > a^*$  текущего среднего знаковой последовательности  $\text{sing} [\dots]$  в дискретные моменты  $0, 1, \dots, (t-1)$ , отстоящие друг от друга на шаг  $\Delta_t$ .

Если имеет место неравенство  $|S_g^*(t)| > \delta_{Sg}$ , то гипотеза «полочки» отвергается.

### *Алгоритмы формирования управляющих воздействий рекомендаций, корректировок и обучения*

В основу этих алгоритмов положен метод восстановительно-прогнозирующей алгоритмизации. Сущность ее содержательно связана с разнообразной практикой человеческой деятельности, в ходе которой находятся ретроспективно (с запаздыванием) образцовые (идеальные) решения, а затем они используются в целях общего обучения и контрольного управления.

При восстановлении образцовых решений производится:

- *сглаживание последовательностей данных*, с целью подавления измерительных помех посредством алгоритма релейно-

экспоненциального сглаживания согласно выражению (8.3); расчет ошибки управления

$$\delta X(i) = X^{\Phi}(i) - X^3(i), \quad (8.11)$$

где  $X^{\Phi}(i)$  – вектор фактического значения выходного параметра при  $i$ -м отсчете (реализации);

$X^3(i)$  – вектор заданного значения выходного параметра;

- *восстановление образцовых (идеальных) управлений*

$$U^B(i) = U^{\Phi}(i) + 1/K \delta X(i), \quad (8.12)$$

где  $U^B(i)$ ,  $U^{\Phi}(i)$  – векторы соответственно восстановленного и фактического управлений на  $i$ -м отсчете (реализации);

$K$  – коэффициент передачи канала управления;

- *приведение управлений к базовым условиям*

$$U^{\delta}(i) = U^b(i) + \sum_{m=1}^M l_m [Z_m(i) - Z_m^{\sigma}], \quad (8.13)$$

где  $Z_m(i)$  и  $Z_m^{\sigma}$  – нерегистрируемые и базовые значения учитываемых факторов;

$l_m$  – пересчетные коэффициенты;

$M$  – число учитываемых факторов;

- *автопрогнозирование приведенных управлений для условий на предстоящий отсчет (реализацию)*

$$U^{\delta}(i+1) = F[U^{\delta}(i), U^{\delta}(i-1), \dots, U^{\delta}(i-n)], \quad (8.14)$$

где  $F[\ ]$  – математический оператор прогнозатора;

- *пересчет базовых экстраполированных управлений на предполагаемые начальные условия*

$$U(i+1) = U^b(i+1) + \sum_{m=1}^M l_m [Z_m(i) - Z_m^{\sigma}]. \quad (8.15)$$

## 8.5. Характеристика базы знаний

База знаний содержит два основных модуля:

- 1) базу знаний для диагностики нарушений хода доменной печи;
- 2) базу знаний для диагностики нарушений теплового режима.

Кроме этого, имеются еще два модуля, которые используются для определения:

- 1) причин (факторов), вызывающих соответствующие нарушения хода печи;
- 2) причин (факторов), вызывающих соответствующие нарушения теплового режима доменной плавки.

Схемы баз знаний указанных модулей представлены на **рис. 8.10-8.13**, из которых очевидна используемая информация и логика построения экспертных подсистем хода доменной печи. Все базы представляют собой многослойные структуры, в элементах которых производятся вычисления коэффициентов уверенности  $CF_i$ .

Некоторый фрагмент такой многослойной структуры схематически изображен на **рис. 8.14**.

$X_1^1$  – давление горячего дутья (диапазон).

Имеет три альтернативы:

- оптимальное значение;
- выше оптимального значения;
- ниже оптимального значения.

$X_2^1$  – давление горячего дутья (тенденция).

Может принимать следующие значения:

- уменьшается;
- увеличивается;
- не меняется.

$X_3^1$  – давление горячего дутья (колебания):

- меньше или равно 0,05 атм;
- больше 0,05 атм, но меньше или равно 0,2 атм;
- больше 0,2 атм.

Основную часть *первого (левого) слоя* образуют  $N$  блоков входных переменных – результатов  $\{X_i^j\}$  работы блока первичной обработки измерений, причем  $X_i^j$  может принимать несколько значений. При этом в некоторых из модульных баз знаний первый слой может содержать переменные, значения которых определяются после работы других модулей. Каждый блок входных переменных описывает результаты измерений одного из технологических параметров печи. Так, например, группа  $X^1 = \{X_i^1\}$ ,  $i = 1, 2, 3$  служит для описания параметров давления горячего дутья. При этом  $X_1^1$  задает диапазон изменения давления,  $X_2^1$  – тенденции изменения, а  $X_3^1$  описывает колебания давления горячего дутья.

*Второй слой* образуют блоки правил  $P_i^j$  вида

**IF** < условие > **THEN**  $u^i = a$  **ELSE**  $u^i = 0$ .

В качестве условия используются предикаты вида  $u^i = < \text{значение} >$ , объединенные логическим «И». Результатом выполнения правила является значение переменной, имеющее смысл промежуточного коэффициента уверенности. Эти коэффициенты определяются экспертом при вводе правила в базу знаний.  $k$ -й блок второго слоя образует правила, в условиях которых используются переменные из  $k$ -го блока переменных. Фактические элементы  $j$ -го блока правил описывают воздействия  $j$ -го технологического параметра на технологический процесс. Для реализации правила второго слоя используется функция «И» (AND).



Рис. 8.10. Схема базы знаний причин тепловых расстройств доменной печи



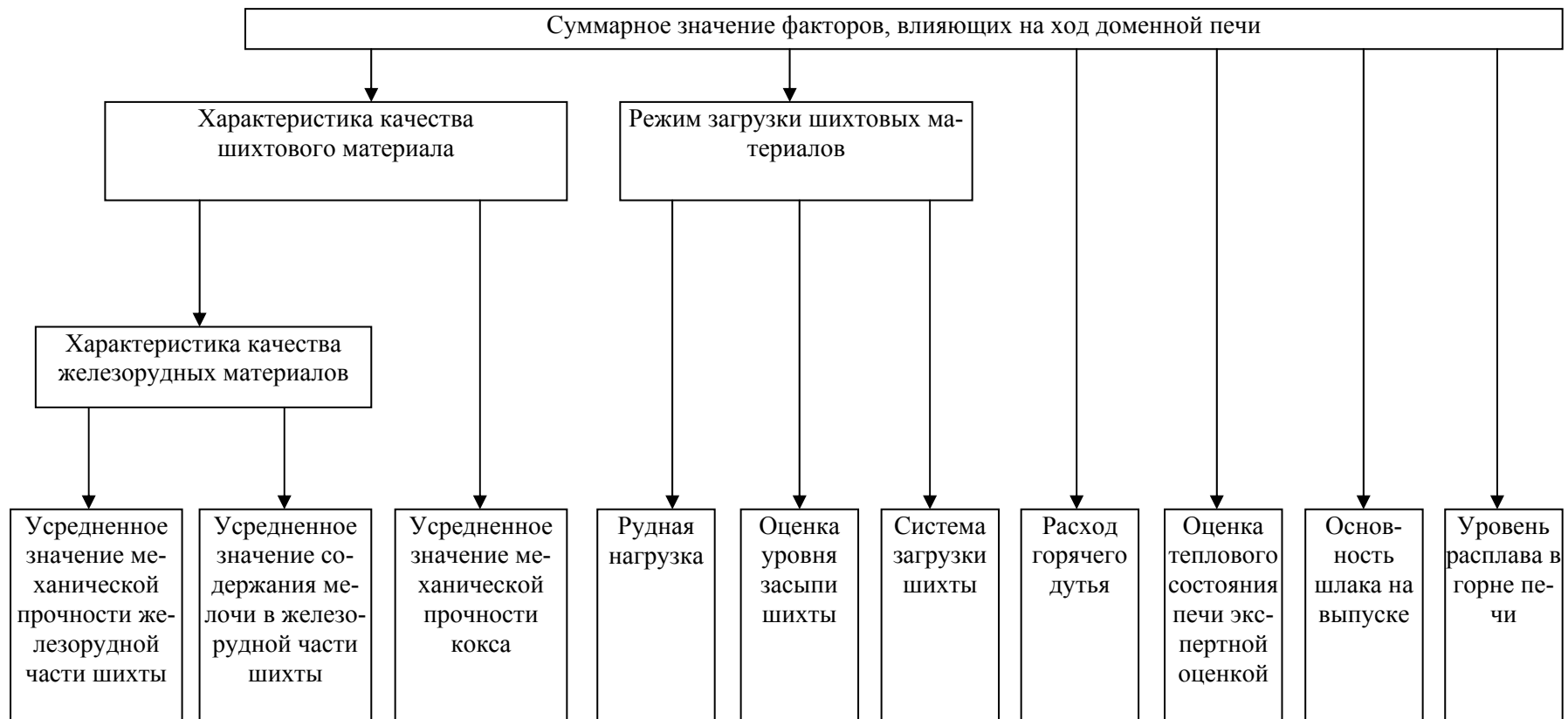


Рис. 8.11. Схема базы знаний причин расстройств ходов доменной печи

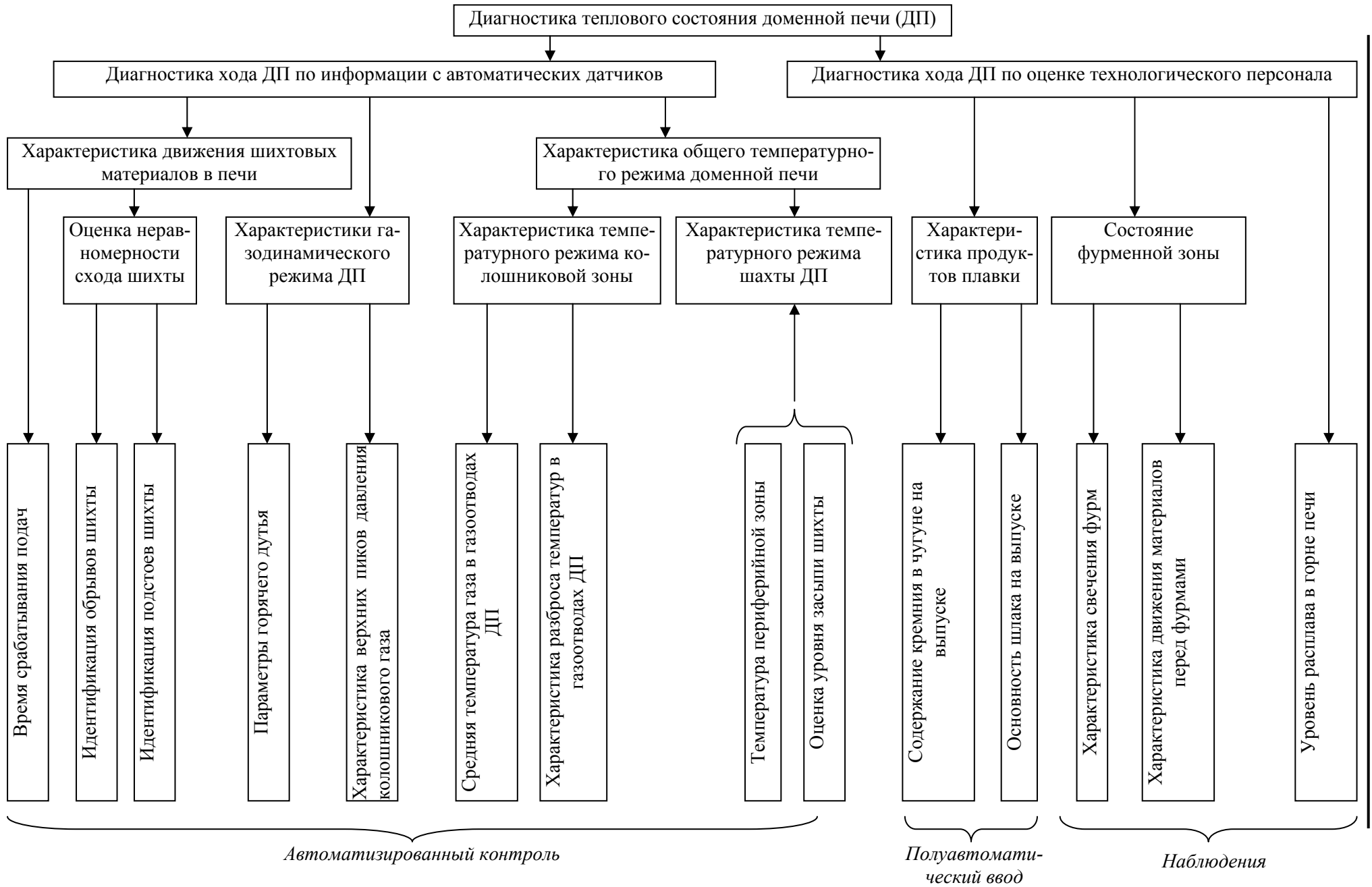


Рис. 8.12. Схема диагностики теплового состояния доменной печи

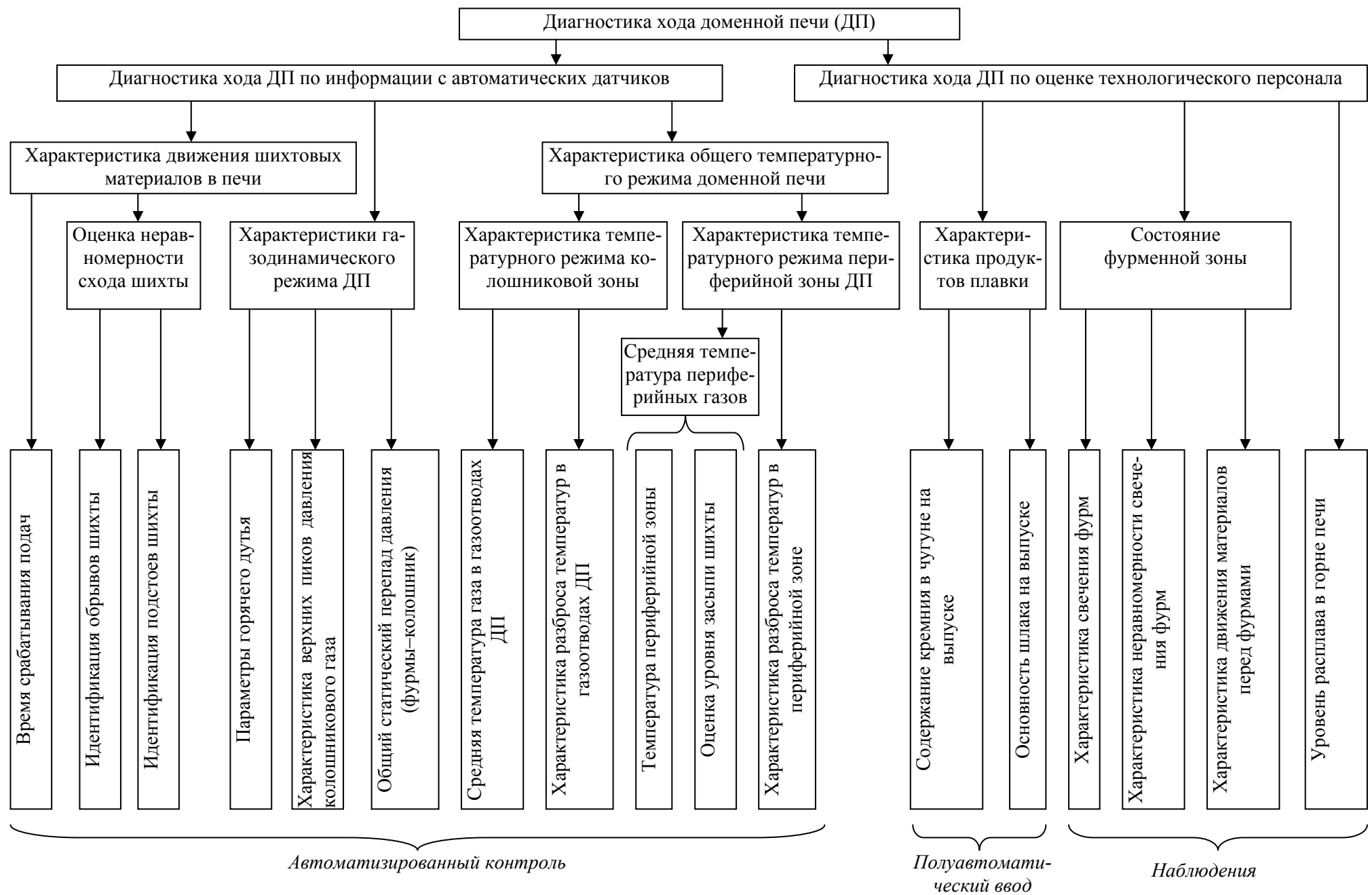


Рис. 8.13. Схема диагностики ходов доменной печи

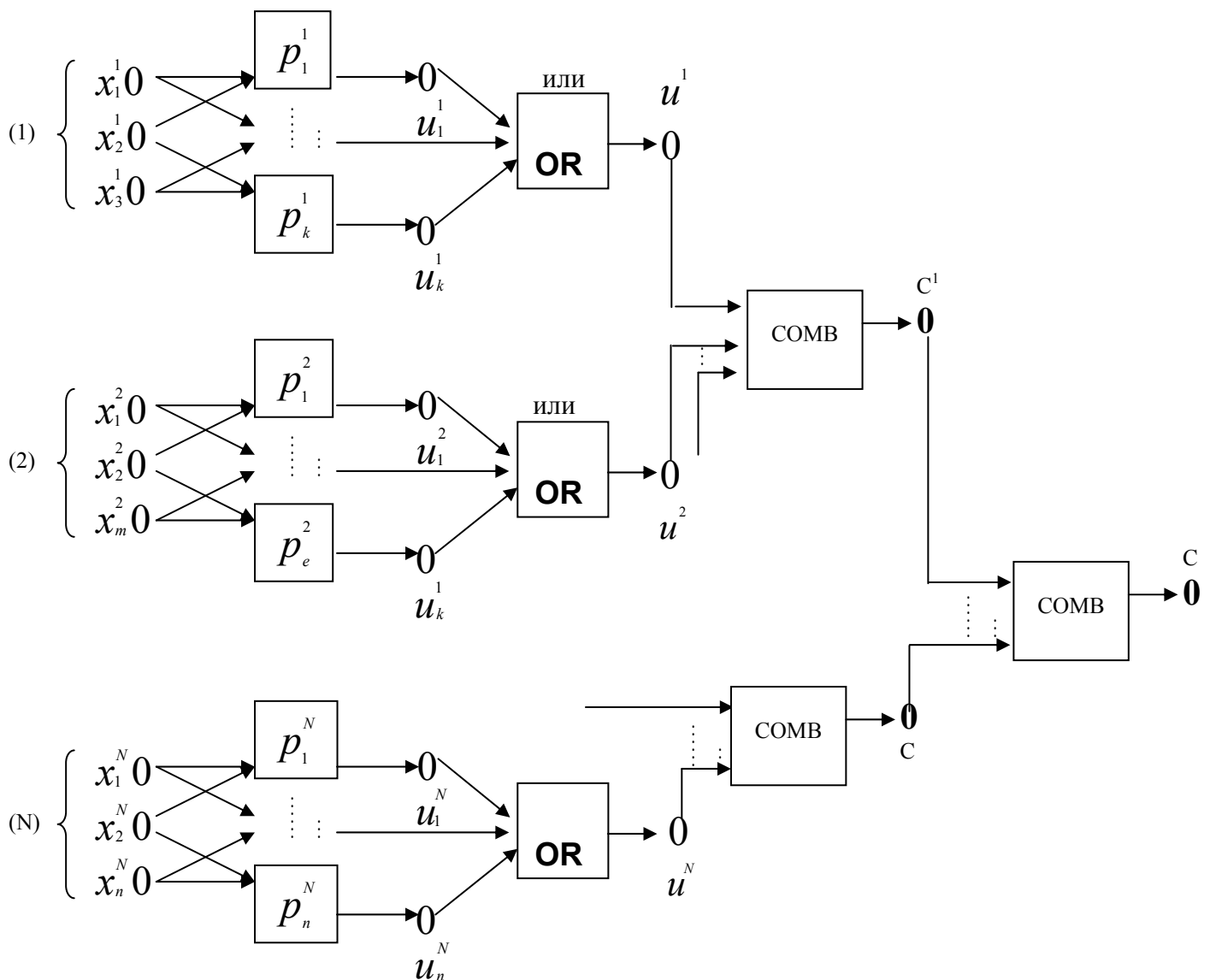


Рис. 8.14. Схема базы данных

Третий слой образуют блоки «ИЛИ» OR-правил. OR-правила выполняются по следующему алгоритму.

Пусть  $S_1, \dots, S_n$  – входные переменные OR-правила, а  $S$  – выходные. Тогда:

- 1) если все  $S_i=0$ , то  $S=0$ ;
- 2) если  $S_i < 0$ , в качестве  $S$  принимается максимальное отрицательное значение  $S_i$ ;
- 3) если  $S_i$  имеют положительные, нулевые и отрицательные значения, то  $S = \max S_i$ .

Как видно из рис. 8.14, «ИЛИ» (OR)-правила используются для объединения результатов, получаемых на втором слое.

Четвертый и последующие слои образуют блоки «КОМБ» (Comb)-правил (см. **рис. 8.14**), в которых вычисляются коэффициенты уверенности по формулам (7.4)–(7.8).

Таким образом, в результате прогноза определяется коэффициент уверенности определения (состояния) хода доменной печи и коэффициент уверенности теплового режима.

Многослойная архитектура базы знаний обладает рядом преимуществ. К ним относятся:

- *простота модификации*. Многослойная структура позволяет легко добавлять, расширять или удалять блоки правил на любом слое. Это свойство особенно важно на двух первых слоях, т.к. позволяет легко учитывать появление новых групп измерений технологических параметров;
- *адаптируемость*. Предложенная структура базы знаний позволяет достаточно просто оценивать воздействия промежуточных коэффициентов уверенности на конечный результат и по результатам таких оценок вносить необходимые коррективы.

Пример отображения информации экспертной системы ведения доменной плавки на мониторе компьютера представлен на **рис. 8.15**.

## 8.6. Контрольные вопросы

1. Чем вызвана необходимость использования экспертных систем в доменном производстве?
2. Почему несмотря на сравнительно большую тепловую инерцию современных доменных печей возможно значительное отклонение режимных параметров плавки от заданных?
3. По каким признакам определяют нормальный режим работы доменной печи?
4. Перечислите параметры, характеризующие тепловое состояние доменной печи, признаки и меры борьбы с холодным и горячим ходами доменной плавки.
5. Перечислите виды нарушения устойчивости газового потока доменной печи. Какими признаками они характеризуются?
6. Перечислите признаки нарушения ровного схода шихты в доменных печах и методы восстановления нормального хода.
7. Какие основные подсистемы включает в себя структура экспертной системы доменной плавки? Укажите назначение основных подсистем.
8. Дайте общую характеристику базы знаний экспертной системы доменной плавки.
9. Какие виды диагностики нарушений доменной плавки определяет экспертная система?

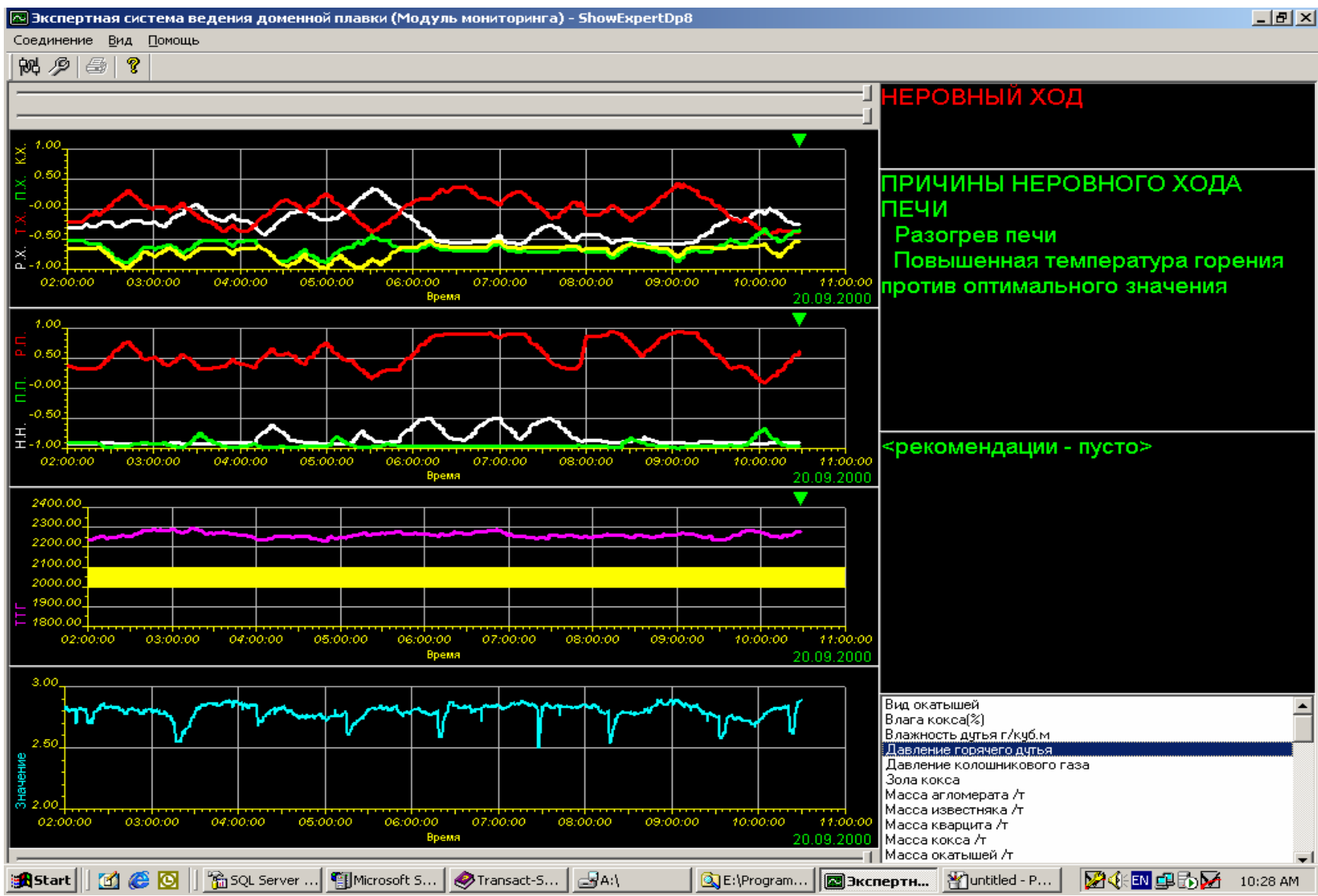


Рис. 8.15. Экспертная система ведения доменной плавки

## Рекомендуемый список литературы

### Основной:

1. Аладьев В.З., Хунт Ю.Я., Шишаков М.Л. Основы информатики: Учебное пособие. – М.: Информационно-издательский дом «Филинь». 1998. – 496 с.; 2-е изд. 1999. – 544 с.
2. Гаврилов Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем: Учебник для вузов. – СПб.: Питер, 2000. – 384 с.
3. Дейт К.Дж. Введение в системы баз данных: Пер. с англ. – 6-е изд. – Киев: Диалектика, 1998. – 784 с.
4. Информатика: Учебник / Под ред. проф. Н.В.Макаровой. – М.: Финансы и статистика, 1997. – 768 с.
5. Metallургия чугуна: Учебник для вузов / Е.Ф.Вегман, Б.Н.Жеребин, А.Н.Похвиснев, Ю.С.Юсфин, В.М.Клемперт. – М.: Metallургия, 1989. – 512с.
6. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Питер, 1999. – 672 с.
7. Острейковский В.А. Информатика: Учебник для вузов. – М.: Высш. шк., 1999. – 511 с.
8. Теплотехника доменного процесса / Б.И.Китаев, Ю.Г.Ярошенко, Е.Л.Суханов и др. – М.: Metallургия, 1978. – 248 с.

### Дополнительный:

9. Абовский Н.П. Творчество: системный подход, законы развития, принятие решений. – М.: СИНТЕГ, 1998. – 312 с.
10. Айвенс К. Эксплуатация Windows NT. Проблемы и решения: Пер. с англ. – СПб.: BHV – Санкт-Петербург, 1998. – 592 с.
11. Безопасность компьютерных сетей на основе Windows NT / В.С.Люцарев, К.В.Ермаков, Е.Б.Рудный, И.В.Ермаков. – М.: Издательский отдел "Русская редакция" ТОО "Channel Trading Ltd.", 1998. – 304 с.
12. Введение в системный анализ теплофизических процессов в металлургии: Учебное пособие для вузов / Н.А.Спирин, В.С. Швыдкий, В.И.Лобанов, В.В.Лавров. Екатеринбург: УГТУ, 1999. – 205 с.
13. Вейскас Д. Эффективная работа Microsoft Access 7.0 для Windows 95: Пер. с англ. – СПб.: Питер, 1997. – 848 с.
14. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика, 1998. – 176 с.
15. Грабер М. Понимание SQL: Пер. с англ. – М.: Наука, 1996. – 420 с.
16. Гук М. Аппаратные средства IBM PC: Энциклопедия. – СПб: Питер Ком, 1999. – 816 с.
17. Дубовик В.Я. Схемотехника устройств связи объект–ЭВМ в металлургии: Учебник для вузов. – Новокузнецк: СибГИУ, 2000. – 300 с.
18. Дьяконов В.П. Математическая система MAPLE V R3/R4/R5. – М.: Солон, 1998. – 399 с.
19. Дьяконов В.П. Справочник по MathCAD PLUS 6.0 PRO. – М.: СК Пресс, 1997. – 336 с.

20. Желтые страницы Internet 98. Русские ресурсы / Ю.Поляк, А.И.Сигалов и др. – СПб.: Питер, 1998. – 600 с.
21. Информатика. Базовый курс: Учебник для вузов / С.В.Симонович и др. – СПб.: Питер, 1999. – 640 с.
22. Использование Internet: Пер с англ. / Дж. Хончкатт, М.Р.Браун, Т.Фронцковяк и др. Киев; М.; СПб.: Издат. дом «Вильямс», 1998. – 592 с.
23. Использование Microsoft Office 97, профессиональный выпуск: Пер с англ./ Д.Бойс, С.Фаллер, Р.Гилген и др. Киев; М.; СПб.: Изд. дом «Вильямс», 1998. – 1120 с.
24. Кенин А.М. Windows 95/NT для пользователей или как научиться владеть компьютером. – Екатеринбург: Планета, 1998. – 432 с.
25. Колпаков С.С., Кузнецова С.Б., Потемкин В.К. Проблемы разработки инструментальных АСУ листопрокатным производством. – М.: Интернет Инжиниринг, 1997. – 272 с.
26. Компьютерные сети: Учебный курс. 2-е изд. (+CD-ROM). – Microsoft Press, 1998. [Русская редакция].
27. Кулаков Ю.А., Луцкий Г.М. Компьютерные сети. – Киев: Юниор, 1998. – 384 с.
28. Кулаков Ю.А., Омелянский С.В. Компьютерные сети: Выбор, установка, использование и администрирование. – Киев: Юниор, 1999. – 544 с.
29. Лепаж И., Яррера П. UNIX. Библиотека системного администратора: Пер. с англ. – М.: Диалектика, 1999. – 528 с.
30. Липаев В.В. Системное проектирование сложных программных средств для информационных систем. – М.: СИНТЕГ, 1999. – 224 с.
31. Лисиенко В.Г. Принципы построения трехуровневых АСУ ТП объектов с распределенными параметрами на примере АСУ нагревом металла: Учебное пособие. – Екатеринбург: УГТУ, 1999. – 73 с.
32. Ложе И. Информационные системы: Пер. с франц. – М.: Мир, 1979. – 632 с.
33. Маклаков С.В. VPwin и ERwin. CASE-средства разработки и проектирования информационных систем. – М.: ДИАЛОГ-МИФИ, 1999. – 256 с.
34. Матвеев Л.А. Компьютерная поддержка решений: Учебник. – СПб.: Специальная литература, 1998. – 472 с.
35. Мейер Д. Теория реляционных баз данных: Пер. с нем. – М.: Мир, 1987. – 608 с.
36. Мешалкин В.П. Экспертные системы в химической технологии. Основы теории, опыт разработки и применения. – М.: Химия, 1995. – 368 с.
37. Назаров С.В. Администрирование локальных сетей Windows NT: Учебное пособие. – М.: Финансы и статистика, 1999. – 336 с.
38. Нильсон Н. Принципы искусственного интеллекта: Пер. с англ. – М.: Радио и связь, 1985. – 376 с.
39. Новиков Ю.В., Карпенко Д.Г. Аппаратура локальных сетей. – М.: ЭКОМ, 1998. – 288 с.
40. Орвис В. Excel для ученых, инженеров и студентов: Пер. с англ. – Киев: Юниор, 1999. – 528 с.
41. Основы современных компьютерных технологий: Учебное пособие / Под ред. проф. А.Д.Хомоненко. – СПб.: КОРОНАпринт, 1998. – 448 с.



42. Першиков В.И., Марков А.С., Савинков В.М. Русско-английский толковый словарь по информатике. – 3-е изд., перераб. – М.: Финансы и статистика, 1999. – 368 с.
43. Плис А.И., Сливина Н.А. MathCAD: Математический практикум для экономистов и инженеров: Учебное пособие. – М.: Финансы и статистика, 2000. – 656 с.
44. Поспелов Г.С. Искусственный интеллект – основа новой информационной технологии. – М.: Наука, 1988. – 200 с.
45. Потемкин В.Г. Система инженерных и научных расчетов MathLAB 5.x.: В 2 т. – М.: ДИАЛОГ-МИФИ, 1999. Т. 1. – 366 с.; Т. 2. – 304 с.
46. Рамм А.Н. Современный доменный процесс. – М.: Metallurgia, 1989. – 304 с.
47. Сайлер Брайан, Спотгс Джефф. Использование Visual Basic: Пер. с англ. Киев.: Изд. дом «Вильямс», 1999. – 832 с.
48. Серов Ю.В. Метрологическое обеспечение технологических процессов черной металлургии (метрология и информатика): Справочник: В 2 кн. – М.: Metallurgia, 1993. Кн. 1. – 272 с.; Кн. 2. – 352 с.
49. Спириин Н.А., Лавров В.В., Шаврин В.С. Оптимизация, идентификация и оценивание теплотехнических процессов в металлургии: Учебное пособие для вузов. – Екатеринбург: УГТУ, 1996. – 188 с.
50. Справочник по искусственному интеллекту: В 3 т. / Под ред. Э.В.Попова и Д.А.Поспелова. – М.: Радио и связь, 1990.
51. Стерн М., Монти Г., Бэчманн В. Сети предприятий на основе Windows NT для профессионалов. – СПб.: Питер Ком, 1999. – 448 с.
52. Теплообмен и повышение эффективности доменной плавки / Н.А.Спириин, Ю.Н.Овчинников, В.С.Швыдкий, Ю.Г.Ярошенко. – Екатеринбург: УГТУ, 1995. – 243 с.
53. Трахтенгерц Э.А. Компьютерная поддержка принятия решений: Научно-практическое издание. – М.: СИНТЕГ, 1998. – 376 с.
54. Хансен Г., Хансен Дж. Базы данных: разработка и управление: Пер. с англ. – М.: ЗАО "Издательство БИНОМ", 2000. – 704 с.
55. Ченцов А.В., Чесноков Ю.А., Шаврин С.В. Балансовая логико-статистическая модель доменного процесса. – М.: Наука, 1991. – 92 с.
56. Шарф Д. HTML 3.2: Справочник СПб: "Питер", 1998. – 224с.
57. Экономическая информатика: Учебник для вузов / Под ред. В.В.Евдокимова. – СПб.: 1997. – 592 с.
58. Экспертные системы: Инструментальные средства разработки: Учебное пособие / Л.А.Керов, А.П.Частиков, Ю.В.Юдин, В.А.Юхтенко; Под ред. Ю.В.Юдина. – СПб.: Политехника, 1996. – 220 с.
59. Элементы теории систем и численные методы моделирования процессов тепломассопереноса: Учебник для вузов / В.С.Швыдкий, Н.А.Спириин, М.Г.Ладыгичев, Ю.Г.Ярошенко, Я.М.Гордон. – М.: Интернет Инжиниринг, 1999. – 520 с.
60. Эндрю А. Искусственный интеллект. – М.: Мир, 1985. – 265 с.
61. Кудинов Г.А. Охлаждение современных доменных печей. – М.: Metallurgia, 1988. – 325 с.

Спирин Николай Александрович  
Лавров Владислав Васильевич

# **ИНФОРМАЦИОННЫЕ СИСТЕМЫ В МЕТАЛЛУРГИИ**

**Конспект лекций  
(отдельные главы из учебника для вузов)**

Редактор издательства Л.Ю.Козьяйчева

ЛР №020315 от 23.12.1996 г.

---

Подписано в печать 03.05.2001

Бумага офсетная

Печать офсетная

Уч.-изд.л. 33,2 Тираж 300 экз. Заказ

Формат 60x84 1/16

Усл.печ.л. 28,72

Цена "С"

---

Издательство УГТУ–УПИ,  
620002, Екатеринбург, ул. Мира, 19

Типография Уральского отделения РАН  
620019, Екатеринбург, ГСП-169, ул. С.Ковалевской, 18