О.Л. Голицына, Н.В. Максимов,
И.И. Попов

Информационные системы и технологии

О.Л. Голицына, Н.В. Максимов,
И.И. Попов

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Рекомендовано Учебно-методическим объединением вузов Российской Федерации по образованию в области прикладной информатики в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению 09.03.03 «Прикладная информатика»

Москва ФОРУМ 2018

УДК 004.02
ББК 32.973-02
Г60

Рецензенты:

А.И. Гусева — доктор технических наук, профессор, профессор кафедры «Экономика и менеджмент в промышленности» ЭАИ НИЯУ МИФИ;
Л.А. Сысоева — кандидат технических наук, доцент, доцент кафедры «Моделирование в экономике и управлении» факультета управления Института экономики, управления и права РГГУ

Голицына О.Л.

Г60 Информационные системы и технологии : учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. — М. : ФОРУМ : ИНФРА-М, 2018. — 400 с. — (Высшее образование).

ISBN 978-5-91134-853-3 (ФОРУМ)
ISBN 978-5-16-009601-8 (ИНФРА-М)

В учебном пособии рассматриваются классификация и структура автоматизированных информационных технологий (АИТ), связанные с ними понятия и определения, роль предметной области. Приводятся базовые АИТ пользователя — обработка текстов, таблиц, мультимедийных данных; смешанные АИТ — распознавание символов, преобразование речи в текст и обратно, машинный перевод. Рассматриваются технологии администратора и разработчика АИС и АИТ — доступ к данным в локальном и сетевом режимах, клиент-серверные архитектуры, средства и технологии информационного поиска.

Предназначено для студентов, обучающихся по направлениям «Прикладная информатика (по областям применения)», «Информационные системы», «Программное обеспечение вычислительной техники и автоматизированных систем», а также для широкого круга специалистов в области информати

УДК 004.02
ББК 32.973-02

ISBN 978-5-91134-853-3 (ФОРУМ)
ISBN 978-5-16-009601-8 (ИНФРА-М)

Человеку свойственно ошибаться, но по настоящему запутывает все только компьютер

Из Законов Мэрфи

Список сокращений

АИПС — автоматизированная информационно-поисковая система
АИС — автоматизированная информационная система
БД — база данных
ЕЯ — естественный язык
ИД — информационная деятельность
ИП — информационная потребность
ИПП — информационная потребность пользователя
ИПС — информационно-поисковая система
ИР — информационный ресурс
ИС — информационная система
ИПЯ — информационно-поисковый язык
ЛО — лингвистическое обеспечение
ОД — основная деятельность
ПО — поисковый образ
ПОЗ — поисковый образ запроса
ПОД — поисковый образ документа
ПрО — предметная область

Введение

С самого начала появления автоматизированных информационных систем все функции контроля, обработки и анализа информации были переданы вычислительным машинам, потому что считалось, что с их помощью в этой области можно сделать все. Однако оказалось, что использование машин отнюдь не всегда эффективно и даже не всегда целесообразно. Традиционные «ручные» методы обработки информации и управления информационными потоками зачастую оказывались намного сложнее для автоматизации, чем это представляли себе специалисты по информатизации. Некоторые информационные потоки, вполне эффективные, пока они проходили от человека к человеку, после перемещения их в машинную среду становились причиной снижения общей эффективности деятельности: данные, для передачи которых раньше использовались прямые «естественные» каналы, теперь должны проходить через процедуры кодирования и обработки ЭВМ. Это приводит не только к заметной задержке, но и требует от человека специальных знаний и навыков.

Другая проблема порождается гигантскими и все увеличивающимися объемами хранимых и перерабатываемых данных, а также появлением все большего числа субъектов, поставляющих, изменяющих и использующих эти данные. Расширяется спектр задач, для которых нужны данные, а вместе с ними растут и потребности вычислительной среды, причем не только в части оборудования — возрастают требования к специализированным программным средствам и технологиям, в том числе и в части их стандартизации и унификации.

На современном этапе развития общества информационные системы (ИС) и технологии (ИТ) становятся теми средствами, которые человек может использовать очень широко. Вместе с тем, в современных употреблениях эти два термина, равно как и основополагающее понятие «информация», являются настолько часто употребляемыми и расхожими, что практически перестали отражать что-либо определенное. К сожалению, в нормативных документах эти базовые поня-

тия определены очень по-разному и скорее отражают специфику области применения документа, чем свойства самого определяемого объекта.

Например, в [1] даны следующие определения:

1) информация — сведения (сообщения, данные) независимо от формы их представления;

2) информационные технологии — процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов;

3) информационная система — совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств.

Эти определения являются достаточно обобщенными: существо определяемого будет зависеть от того, какой смысл будет вкладываться в базовые понятия — сведения, сообщения, данные, процессы, методы, способы, средства. Но главное — они не отражают особенности их применения в ИТ-сфере: для их конструктивного использования при построении и использовании автоматизированных информационных систем (АИС) необходимо определить операционные свойства основного объекта обработки — информации, а также функциональные возможности операционных сред, в которых она обрабатывается.

Пособие обобщает многолетний опыт преподавания авторами дисциплин, относимых к «информационному» блоку, как в технических, так и в гуманитарных вузах, в том числе таких, как НИЯУ МИФИ, РГГУ, РЭУ им. Плеханова. В своей работе авторы руководствовались и тем, что материал должен не только представлять существо конкретной темы, но и подвести читателя к пониманию обоснованности (или условности) того или иного решения. Авторы сознательно избегали описаний языков и технологий, применяемых в конкретных системах, предполагая, что полноценное освоение материала курса так или иначе должно быть связано с практикой и, соответственно, с неизбежным изучением конкретных подходов, языков и технологий, свойственных выбранной системе и изложенных в специальных пособиях, учебниках и руководствах.

Данное пособие написано в предположении, что читатели владеют основами информатики и программирования.

Учебное пособие предназначено для студентов вузов, обучающихся по направлению 230700 «Прикладная информатика», а также для учащихся техникумов по специальности 2203 «Программное обеспечение вычислительной техники и автоматизированных сис-

тем». Пособие обеспечивает формирование следующих профессиональных компетенций бакалавров:

- способность анализировать при решении профессиональных задач социально-экономические проблемы и процессы с применением методов системного анализа и математического моделирования (Б1);
- способность использовать основные законы естественнонаучных дисциплин в профессиональной деятельности и эксплуатировать современное электронное оборудование и информационно-коммуникационные технологии (Б2);
- способность применять системный подход и математические методы в формализации решения прикладных задач (Б3);
- способность осуществлять и обосновывать выбор проектных решений по видам обеспечения информационных систем (Б4);
- способность проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе, участвовать в реинжиниринге прикладных и информационных процессов (Б5);
- способность оценивать и выбирать современные операционные среды и информационно-коммуникационные технологии для информатизации и автоматизации решения прикладных задач и создания ИС (Б8);
- способность ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий (Б10).

В целом курс ориентирован на развитие и таких общепрофессиональных компетенций, как способность понимать роль и значение информации и информационных технологий в развитии современного общества и научного знания, а также способность использовать, обобщать и анализировать информацию, ставить цели и находить пути их достижения в условиях формирования и развития информационного общества.

В **1-й главе** представлены общие вопросы терминологии, понятий и классов объектов и процессов, связанных с проблематикой информационных систем и технологий. С системных позиций вводятся основные понятия: информация, данные, знания. Определены состав и структура информационной системы, рассматриваемой как средство автоматизированной обработки данных. Вводится определение информационной технологии и классов ИТ.

В **главе 2** представлены основные (базовые) типы технологий пользователя: обработка текстовой и табличной информации. Рас-

смотрены определения моделей документа, языки разметки документов, технологии XML, функции текстового процессора MS Word; работа с электронными таблицами на примере MS Excel.

Глава 3 посвящена описанию основных принципов мультимедийных технологий: обработка аудиоинформации; форматы аудиосигнала; технологии и программные средства обработки статических изображений; принципы и элементы технологий цифрового видео.

Глава 4 содержит описание «смешанных» информационных технологий, в том числе: оптическое распознавание текстов; системы распознавания и генерации речи; средства автоматизированного перевода текстов.

В главе 5 рассматриваются технологии доступа к данным: файловые системы, базы данных и СУБД; физическая организация данных в системах управления данными, а также хранилища данных и их использование для анализа информации.

Глава 6 содержит описание сетевых информационных технологий и технологий Internet. Рассмотрены структуры сетей, модель взаимодействия открытых систем, технологии Internet, прикладные протоколы коммуникации Internet, распределенные файловые системы Internet, распределенные информационные системы Internet.

В **главе 7** рассматриваются основные принципы технологий информационного поиска; обобщенный технологический процесс АИПС; методологические основы информационного поиска; компоненты и обобщенная схема информационного поиска.

Глава 8 представляет технологии распределенной обработки информации: распределенные информационные ресурсы; клиент-серверные архитектуры распределенной обработки данных; архитектуры сервера баз данных; схемы размещения и доступа к данным в распределенных БД; объектно-ориентированные технологии распределенной обработки; электронные библиотеки.

В **приложении** приводится глоссарий терминов.

В целом пособие должно рассматриваться и как введение в проблематику автоматизированной обработки информации, хотя, в силу ограниченности объема, не претендующее на полномасштабное изложение материала разделов, каждый из которых представляет отдельную дисциплину и зачастую представлен отдельным полноценным пособием.

Глава 1
ВВЕДЕНИЕ В ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Приведем в дополнение к использованному во введении определению понятия «информационная система» [1] определение, данное в [Криницкий, 1982]: «АИС — это комплекс, состоящий из информационного фонда и процедур: управляющей, обновления, информационного поиска и завершающей обработки, позволяющих накапливать, хранить, корректировать и выдавать информацию»[1]. Данное определение имеет функционально-структурный характер, а неявно присутствующее понятие «системности» отражает существо функциональности: состав и структура ИС определяются исходя из требований к уровню *эффективности обслуживания информационных потребностей* пользователей в контексте задач основной деятельности.

Целесообразно конкретизировать и другое основополагающее понятие — «информационная технология», которое согласно [2] определяется как «практическая деятельность и прикладная наука, имеющие дело с данными и информацией. Примером является сбор, изображение, обработка, обеспечение безопасности, передача, взаимообмен, представление, управление, организация, хранение и восстановление данных и информации». Здесь важно *конструктивно* уточнить это понятие следующим образом: *«Информационная технология* — это представленное в проектной форме (формализованном виде, пригодном для практического использования) концентрированное выражение научных знаний и практического опыта, позво-

[1] Такое определение, безусловно, является более конструктивным по своей сути, но, тем не менее, оно не проводит границы между информацией и данными, которые собственно накапливаются, обрабатываются и передаются.

ляющее рациональным образом организовать тот или иной достаточно часто повторяющийся информационный процесс. При этом достигается экономия затрат труда, энергии людских и материальных ресурсов, необходимых для реализации данного процесса» [Колин, 1995]. Данное определение подчеркивает, что ИТ — это не только программные, технические и другие средства обработки информации и отнюдь не любые способы осуществления процессов и методов, как это иногда представляется, и что только на основе обобщенного подхода к этому понятию можно достигать нового качества, оптимизировать разнообразные информационные процессы, начиная от подготовки и издания печатной продукции и заканчивая информационным моделированием и прогнозированием глобальных процессов развития природы и общества.

Особенностью информационных систем и технологий является то, что они не могут рассматриваться изолированно, вне материальной сферы. Информация является неотъемлемым и, часто, определяющим компонентом практически всех материальных процессов, которые инициирует (или в которых участвует) человек, т. е. эффективность их использования проявляется и может быть оценена только в сфере материального производства.

Другая важная особенность предопределена естественным для больших систем требованием надежности и устойчивости функционирования и развития, а также возможности интеллектуального (человеческого) контроля в условиях большой сложности. Это означает, что сфера информационных технологий должна быть распространена практически на все этапы жизненного цикла продукта.

Третья особенность — это непосредственное или опосредованное участие человека в технологических процессах. Любой автоматизированный и даже автоматический процесс на том или ином этапе связан с необходимостью представления (или получения) информации в форме, удобной для человека. Это породило, в частности, и отдельное направление — технологии человеко-машинного взаимодействия и интерфейсы информационных систем.

Можно сделать вывод, что хотя информационные технологии в значительной степени ориентированы на индустрию, они должны рассматриваться не только как инструмент, умножающий возможности человека, но также и как методологическая платформа, обладающая универсальными парадигмами, моделями, методами, языками для представления, формализации, моделирования, систематизации, обработки прикладных знаний.

1.1. Информация и информационные процессы

Реально любая информационная система всегда является частью какой-либо более общей, например, производственной, социальной, научно-исследовательской системы, а люди или системы, которые были генераторами информации, в другой момент времени будут потребителями этой или какой-либо другой информации. Обобщенная схема информационных потоков и место ИС в процессе управления представлены на рис. 1.1. Здесь информационная система представлена в самом широком понимании как система, обеспечивающая выработку (или реализацию) *информации (*управляющего воздействия*)* путем ее синтеза на основе уже имеющейся информации: сведений о состоянии объекта, окружающей среды, а также об эффективности ранее выработанной информации.

Рис. 1.1. Место ИС в процессе управления

Это позволяет говорить о цикличности *информационного обмена*, когда информация, получаемая в результате одного *созидательного процесса*, так или иначе будет использована в другом процессе. Свойство «созидательности» здесь отражает существенную особенность подхода к рассматриваемому классу ИС: информация не является са-

модостаточным объектом, а ее генерация не является исключительной целью субъекта, управляющего процессом. То есть информация появляется в результате некоторой целенаправленной — *основной* деятельности (ОД) субъекта, обычно тесно связанной с его существованием в окружающей среде, и *используется* в какой-либо последующий момент времени этим или каким-либо другим субъектом в рамках этого или какого-либо другого процесса деятельности.

1.1.1. Данные, информация, знания

Термины «информация», «знания», «данные» в большинстве учебников и пособий практически не различаются. Но различие становится очевидным, если рассмотреть их «функциональную» взаимосвязь.

В стандарте [2] термины «знания», «информация» и «данные» рассматриваются с точки зрения их *обработки,* где они определены как словесно, так и с помощью графической иллюстрации (рис. 1.2) следующим образом.

Знания — организованная, интегрированная совокупность фактов, событий, предположений и правил, организованных для системного использования.

Информация — любой факт, понятие или значение, полученное из данных, а также контекст, выбранный из знаний, или контекст, ассоциированный со знаниями.

Данные — представление информации в некотором формализованном виде, пригодном для передачи, интерпретации или обработки.

Термины «знания», «информация» и «данные» в приведенной выше вербальной форме определены «изолированно» друг от друга (только с точки зрения их *отдельной* обработки), в то время как на рисунке они представлены *в процессе* — до и после обработки. Но здесь надо отметить, что с алгоритмической точки зрения («вход-выход») эта схема все же не дает взаимно корректных определений.

В [Ожегов, 1989] понятие «информация» определяется как «сведения об окружающем мире и протекающих в нем процессах, воспринимаемых человеком или специальным устройством», где *сведения* — это «познания в какой-либо области, известия, сообщения, знания, представление о чем-либо».

В [Винер, 1968] информация определяется уже как «...факты, получаемые из внешнего мира в процессе приспособления, т. е. в процессе взаимодействия с внешним миром».

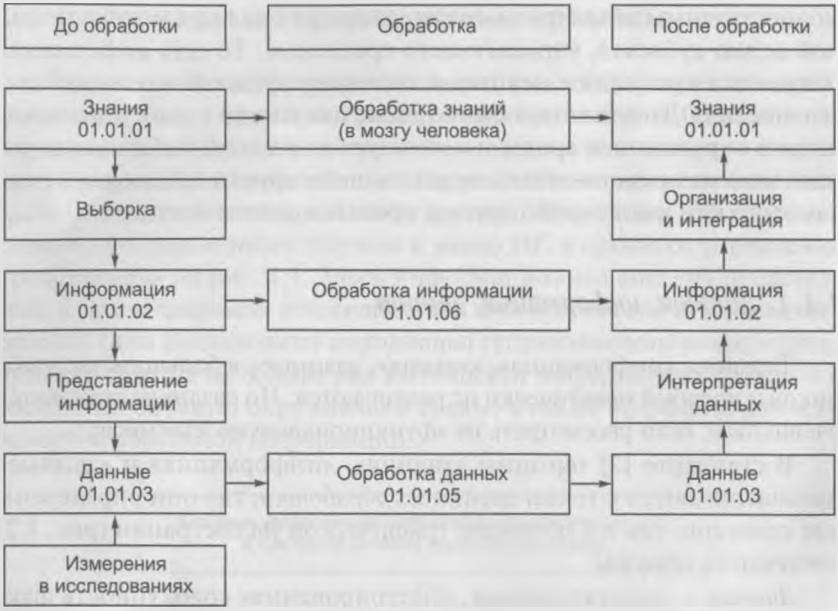


Рис. 1.2. Иллюстрация соотношения между данными, информацией и знаниями в ISO/IEC 2382-1

Такая несогласованность определений приводит к принципиально важному вопросу: имеют ли знания, информация, данные общую природу или это совершенно разные субстанции?

Приведенные выше определения и формулировки подчеркивают «вторичность» этих сущностей и неявно указывают на два принципиальных положения: 1) взаимодействие объектов в природе сопровождается *сопутствующими* образами (сведениями, фактами), которые воздействуют на сам процесс или отражают результат собственно взаимодействия, и 2) взаимодействие всегда так или иначе *ориентировано* и характеризуется, например, направленностью, действенностью, эффективностью и т. д., а их *характеристики* обычно физически и составляют образ. Отсюда также можно вывести предположение, что информация с точки зрения формы и структуры является неатомарным, композиционным объектом.

Рассмотрим простой пример типичного процесса информирования (во взаимодействии пассажира с железнодорожным транспортом) — получения сведений о времени отправления последней электрички до нужной станции. В случае если мы обращаемся к сотрудни-

ку справочной службы вокзала, то лаконичным, но абсолютно ожидаемым, полным, точным (а главное — разрешающим нашу жизненную проблему) ответом будет, например, «23 часа 20 минут». Точно такое же выражение (величину, определяющую время суток) можно было бы обнаружить (или получить) и в других местах, например, в собственной записной книжке, на экране компьютера, в объявлении киносеансов и т. д. Но, воспринимая эти вполне *правильные по величине* данные, мы, естественно, не отправимся на вокзал. Эти данные (*величина*, а не *значение*) для нас не будут *действенными*. Разница между этими ситуациями вполне очевидна, и это позволяет сделать вывод о том, что значение одной и той же величины определяется обстоятельствами (*контекстом*) ее появления или использования. В примере с сотрудником справочной службы такой контекст содержится в *отдельной от ответа* цели заданного вопроса или в том, что этот вопрос *направлен* компетентному лицу. В других случаях контекст может явно содержаться в развернутом ответе или неявно определяться его структурой.

То есть информация может рассматриваться как данные, связанные с определенным контекстом и обладающие свойством действенности.

Такая формулировка подчеркивает, что *контекст* существует как самостоятельный объект и его надо обрабатывать, в том числе и как самостоятельный объект. Система (и человек) *берут* данные (сигналы, величины и т. д.) в общем случае из большого множества и *выбирают* эти данные не столько по величине самих данных, сколько именно по признаку соответствия контексту (только те, которые, необходимы *для решения конкретной задачи)*.

Отсюда следует, что *обрабатываемые данные* должны обладать, а точнее — *должны быть связаны* с контекстом, который ассоциируется с отличительными признаками, также в свою очередь представляющими собой некоторый набор данных. Далее, для получения прагматического результата эти данные обрабатываются прикладной программой (данные связываются с методом обработки, являющимся одной из форм задания контекста), и, в итоге, полученный результат (тоже данные) будет связан со своим контекстом — способом его использования, что реально и обеспечит действенность информации для конечного пользователя. Данное соотношение представлено схемой на рис. 1.3.

В этом смысле комплексность объекта «информация» косвенно, но очень точно иллюстрируется определением свойства документа «пригодность для использования»— наиболее часто используемого

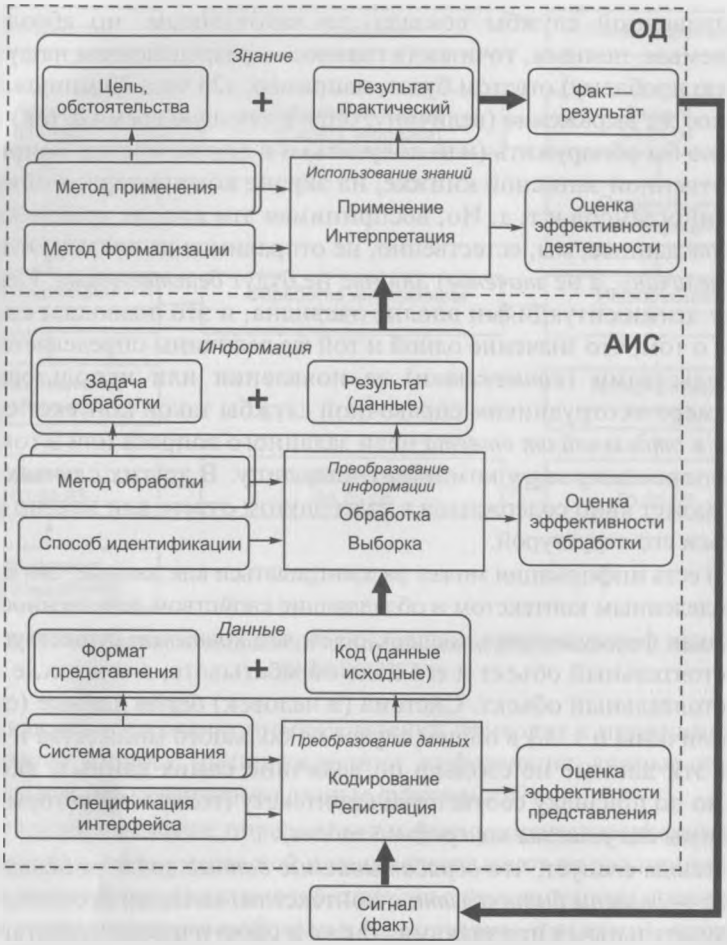


Рис. 1.3. Контекстное окружение данных

в качестве эквивалента понятия «информация». Согласно [3], «...пригодным для использования является документ, который можно локализовать, найти, воспроизвести и интерпретировать. При воспроизведении он должен отражать связь с деловой деятельностью или операцией, в результате которой он был создан. Контекстные ссылки документов должны нести информацию, необходимую для понимания операций деловой деятельности, в которых эти документы были созданы и применялись. Должна быть предоставлена возможность идентифицировать документ в более широком контексте — контексте

деловой деятельности и функций. Связи между документами, фиксирующие последовательность действий, должны быть сохранены».

Отметим, что схема (см. рис. 1.3) предопределяет и еще одно широко используемое в информатике понятие — «интерфейс». Согласно [2], «интерфейс — это разделительная граница, определенная с помощью характеристик этой границы». Ключевым в этом определении является не слово «граница» (в смысле разграничения), а «характеристики» — набор свойств, которыми *должен* обладать воспринимаемый объект (сигнал, данные и т. п.) для того, чтобы он с помощью процедуры, ориентированной на обработку соответствующей формы представления, был передан процедуре собственно функциональной обработки[1]. То есть отдельная обработка как процесс в общем случае включает, по крайней мере, два подпроцесса — функцию преобразования данных, реализующую получение целевого результата, и обеспечивающую ее интерфейсную функцию преобразования во внутреннее представление, соответствующее возможностям целевой функции.

Отметим, что интерфейсные функции связаны с двумя типами задач: 1) *кодирования* — представления содержания в форму, обеспечивающую целевую обработку в конкретной среде, и 2) *идентификации,* обеспечивающей «узнавание» процедурами, внешними по отношению к среде обработки, что собственно и позволяет осуществлять выборку только нужных для целевой обработки данных.

Этой точке зрения отвечает и определение в [Математический словарь, 1988]: «Данные в информатике — факты или идеи, выраженные средствами формальной системы, обеспечивающей возможности их хранения, обработки или передачи. Такую формальную систему называют *языком представления данных*; синтаксис этого языка — *способом представления информации*; его семантику или прагматику — *информацией*». Данное определение подчеркивает, что термины «знания», «информация», «данные» необходимо рассматривать не только с позиций обработки, но и с точки зрения *представления, хранения, поиска и передачи*.

---

[1] В общем случае интерфейсные процедуры должны быть до и после функциональной обработки: до обработки — приводить данные к форме адекватной специфике реализации целевой функции, после — к форме, адекватной особенностям использования (восприятия) результата приемником. На рис. 1.3 процесс интерфейса выхода отсутствует, так как согласование форм реализуется только на стороне принимающего процесса, что, если учесть однородность операционного пространства АИС, вполне рационально.

Рассмотрим понятие «данные», которое в [Информатика, 1999] вводится следующим образом: «Мы живем в материальном мире. Все, что нас окружает и с чем мы сталкиваемся, относится либо к физическим телам, либо к физическим полям. Все объекты находятся в состоянии непрерывного движения и изменения, которое сопровождается обменом энергией и ее переходом из одной формы в другую. Все виды энергообмена сопровождаются появлением *сигналов*. При взаимодействии сигналов с физическими телами в последних возникают определенные изменения свойств — это явление называется *регистрациеи сигналов*. Такие изменения можно наблюдать, измерять или фиксировать теми или иными способами — при этом возникают и *регистрируются* новые сигналы, т. е. образуются *данные*».

Это определение подчеркивает *основное* свойство данных — их объективное и безусловное существование. Они независимы от субъекта наблюдения или управления и сами по себе никак явно не обусловлены связями с будущим: однажды полученные данные могут никогда не использоваться или использоваться совсем иначе, чем это предполагалось при их возникновении. В то же время вторичность данных по отношению к сигналу отражает их искусственную и, следовательно, в некоторой степени субъективную природу: регистрация сигналов (т. е. порождение данных) производится специальным устройством на соответствующем носителе, каждый из которых выбирается или создается некоторым субъектом для решения конкретных задач в конкретное время.

Помимо данных-фактов следует отдельно выделить специфические данные, представляющие задачи, проблемы или гипотезы (объекты, имеющие абстрактную природу). В целенаправленной деятельности (человека) именно формулировки задач выступают в роли данных, инициирующих (определяющих) функциональный процесс, где в качестве ресурса может выступать как материя или энергия, так и данные. Роль и характер используемых данных в целом отражены схемой управляемого функционального процесса, представленной на рис. 1.4.

Информационная система, функциональность которой обусловлена проблемным контекстом (данными, представляющими целевую задачу), фактически преобразует информацию. Потенциально полезные данные, выделенные из общего множества в соответствии с контекстом задачи (исходная информация), в результате использования порождают выходную информацию — актуализированные данные, подтверждающие или отрицающие действенность выбранных исходных данных для решения задачи.

Рис. 1.4. Обобщенная схема функционального процесса, управляемого данными

Для субъекта управления выходная информация выступает в роли основы для формирования личного или общественного *знания* — данных, актуальность и полезность которых подтверждена процессом преобразования ресурса — соответствием цели деятельности субъекта или общества. Особенностью этих данных является то, что они помимо фактов, характеризующих процесс, могут включать и другую разновидность данных — *сигналы* о выявленных проблемах и вновь возникших задачах.

1.1.2. Информация и информационные взаимодействия

В информатике, управлении или в социальных и в физических системах понятие информации присутствует практически везде. И хотя в каждом случае это понятие определяется по-разному, тем не менее, для соответствующей области знаний оно практически всегда бывает достаточно конструктивным. Однако в области информационных систем и технологий необходимо не только формально определить термин, но и понимать природу, что позволит адекватно использовать свойства информации.

Все используемые и изучаемые человеком объекты и их взаимодействия так или иначе связаны с физическим миром[1]. Это отно-

[1] В настоящее время в физике известны четыре *фундаментальных* вида физических взаимодействий: слабое, сильное, электромагнитное, гравитационное. Все остальные виды взаимодействия, в том числе и людей, имеют в своей основе перечисленные. Наблюдаемая действительность показывает, что все функции сознания (получение, интерпретация, генерация информации пр.) также реализуются физическими и химическими процессами.

сится и к абстрактным объектам, которые существуют в сознании «физического» человека, а их взаимодействие с окружающим миром возможно, только если они будут воплощены в изделиях или представлены материальными средствами — языком.

Информация в этом смысле не является исключением. Все объекты, которые выступают в роли информации, представлены вполне определенными физическими формами существования. Однако в некоторых случаях (в частности, в информационной деятельности) проявляются особые свойства информации (по отношению к физическим), такие как эмерджентность, кумулятивность, неассоциативность и т. д.

Понятие информации очень тонкое, и подходы к его определению следует искать через характер и особенности взаимосвязей, аналогично тому, как свойства объекта определяются через его взаимодействия.

Будем называть предметной областью (ПрО) совокупность объектов (и/или их состояний) естественного или искусственного происхождения, существующих либо в виде сложившегося в результате эволюции устойчивого естественного образования, либо выделяемые некоторым субъектом в соответствии с целями его деятельности. Очевидно, что границы таких ПрО будут всегда достаточно условными.

Информационными (ИнфОб) будем считать такие объекты[1], которые обладают свойством (способностью) отражать и, при соблюдении определенных условий, изменять состояние другого физического объекта или его взаимосвязи. Информационные объекты для нас *представляют* «атомы» информации: это та форма существования информации, которая обеспечивает возможность ее хранения и передачи в пространстве и во времени. Информационный объект, порожденный каким-либо другим объектом, существует независимо от него. Такой объект не тождественен ни тому, который был причиной его появления, ни объекту, на который он воздействует, и в то же время он сам может быть объектом информационного воздействия.

Информационной средой (ИСр) предметной области будем считать множество информационных объектов, специально создаваемых либо выделяемых в этой или какой-либо другой ПрО.

Выделение во множестве физических объектов двух, в общем случае, пересекающихся подмножеств (предметной области и информа-

[1] Термин «объект» здесь используется в самом общем смысле, без учета его природы — физической или абстрактной.

ционной среды) достаточно условно. Объект, являющийся «информационным» для одной предметной области, не обязательно будет таковым для другой: для различных состояний системы (как и для различных предметных областей) эти разбиения в общем случае будут разными. Например, восход солнца: (1) в ПрО «Солнечная система» — это одно из состояний движения космических тел, (2) в ПрО «Планета Земля» — начало светового дня. При этом как информационный объект он может: (1) в ИСр «Счисление времени» использоваться как точка отсчета времени; (2) в ИСр «Организация рабочего дня» — как сигнал «Подъем».

Отметим также, что упомянутая условность имеет двойственную природу:

1) объективную, которая связана с фактором времени и отражает неизбежность естественного изменения состояния объекта с течением времени;

2) субъективную, которая связана с функциональностью объекта (рассматриваемого, скорее всего, как часть некоторой системы взаимосвязанных объектов), которая отражает некую внешнюю цель изменения — перевода его в более предпочтительное состояние.

Существенно, что конечной целью здесь является не столько установление степени соответствия информационного сообщения содержанию той или иной предметной области, сколько его потенциальные возможности по изменению состояния данной ПрО, или, другими словами, *действенность* в определенных условиях. Это неизбежно предполагает наличие конкретного *процесса* (и, соответственно, «процессора»), реализация которого, в свою очередь, обусловлена средой взаимодействия.

Будем называть *информационными взаимодействиями* такие процессы, в которых в качестве операционных используются объекты-образы. В отличие от физических взаимодействий, непосредственно происходящих между отдельными телами или силами (оригиналами) в конкретный момент времени, в информационных взаимодействиях будут участвовать и их образы. Причем такие объекты-образы способны вступать между собой во взаимодействие уже без ограничений, присущих оригиналам, что позволяет для развития оригиналов использовать эволюцию их образов.

Такие процессы используют *память* —операционную среду объектов-образов, внешнюю по отношению к объектам-оригиналам. Отсюда следует, что для реализации взаимодействия среда должна иметь

механизмы сопряжения (приведения) форм существования взаимодействующих объектов.

Таким образом, информационные объекты и взаимодействия являются объективной и закономерной действительностью, существующей наряду с физическими взаимодействиями. Информационные взаимодействия отличает нелинейность (необратимость), обусловленная дискретным характером процесса отображения из одного пространства в другое.

С функциональной точки зрения с понятием «информация» связываются два типа действий: (1) коммуникационные — обеспечивающие «перемещение» ИнфОб для последующего взаимодействия в пространстве ИСр; и (2) собственно информационные — реализующие «действенность» информации. Если в первом случае должна быть обеспечена целостность (неизменность содержания) объекта, то действенность информации, реализуемая во время ИнфВ, может быть достигнута только в случае нарушения этой целостности.

Как было отмечено ранее, в процессе информационного взаимодействия появляется объект (образ), физически не зависящий от оригинала. Однако здесь следует обратить внимание на то, что исходный информационный объект, практически не меняя своего физического состояния, явно или неявно обретает новое свойство — «быть использованным в данной ПрО». Очевидно, что при этом характер и результаты взаимодействия его с другими предметными областями не будут такими же. То есть информация, взаимодействуя с конкретной ПрО, принимает одно состояние из множества возможных. Такое состояние информации, *зафиксированное* в виде *контекстно дообусловленного информационного объекта* (связанного особенностями ее применения в выбранной ПрО), и может называться *знаниями*.

1.1.3. Знания и научная[1] информация

Прагматический смысл проявления знаний сводится к тому, что «знать» — это уметь вести себя адекватным образом в ситуациях, связанных с индивидуальными или коллективными взаимодействиями.

---

[1] Здесь и далее будет рассматриваться научная информация, которая, в отличие от производственной, бытовой и т. д., уже при ее генерации ориентирована не только на конкретику текущего момента, но и на гипотетическое будущее, что уже предполагает необходимость ее существования в виде самостоятельного операционного объекта в рамках замкнутой искусственной информационной среды.

Аналогично понятие *научно-техническая информация* с информационно-технологической точки зрения определяется в [Муранивский, 1982] как «*задокументированное научное знание*, введенное в оборот и участвующее в функционировании и развитии общества». Это определение подчеркивает, что знание, не получившее «толчка» для циркуляции в обществе (в общем случае — в среде), не может рассматриваться как информация. Отсюда также следует, что информация не существует без носителя, обеспечивающего ее передачу. При этом существование информации не зависит от вида носителя и формы представления, однако от этого зависят возможность и эффективность ее использования. Например, информация, представленная в знаковой системе, незнакомой получателю, или на носителе, данные на котором не могут быть доступны по причине отсутствия устройства для чтения, не будет использована.

Информация выступает как форма знания, отчужденная от его носителя (сознания субъекта) и обобществляющая его для всеобщего использования. То есть информация — это динамическая форма существования знаний, обеспечивающая их распространение и функционирование. Получая информацию, пользователь превращает ее путем интеллектуального усвоения (информационно-когнитивного процесса) в свои новые личностные знания, т. е. происходит воссоздание знаний на основе информации.

Обобщая, можно сказать следующее. Зафиксированные (воспринятые) сигналы окружающего мира представляют собой объективно существующие *данные*. *Информация* появляется при использовании данных в процессе решения конкретных задач, при этом происходит формирование нового *личного знания* субъекта. Результаты решения задач, обобщения в виде законов, теорий, совокупностей взглядов и представлений, полученные отдельными субъектами и выступающие в этот момент как проверенная информация, образуют *обобществленные знания*, отчужденные от субъектов, их сформировавших. Такие знания представляются обычно в форме *документов* и сообщений, которые существуют объективно и относительно независимо от контекста их получения и в свою очередь могут рассматриваться как данные, в том числе в других предметных областях и с другим контекстом.

Функциональное соотношений этих понятий иллюстрируется схемой, приведенной на рис. 1.5.

Станут ли данные информацией, зависит от того, известен ли предопределенный контекстом метод преобразования (отражения)

данных в новые или уже известные понятия. То есть: чтобы извлечь информацию из данных, необходимо иметь метод получения информации, соответствующий этим данным[1].

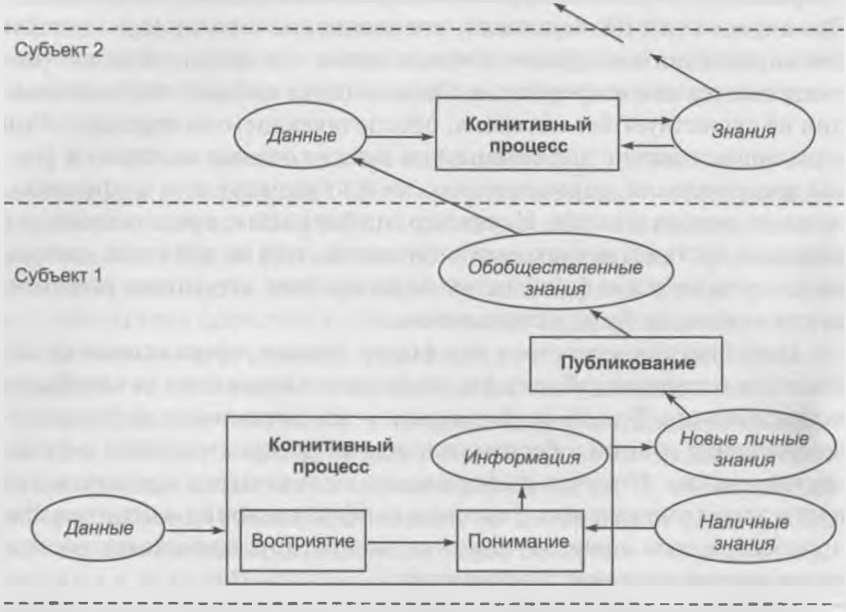


Рис. 1.5. Соотношение понятий «информация», «данные», «знания»

Одни и те же данные могут представлять разную информацию (в итоге — по-разному действовать) в зависимости от взаимодействующих с ними методов, к которым надо отнести и условия ее извлечения (например, уже имеющиеся у субъекта знания). То есть информация возникает и существует в момент взаимодействия *объективных* данных и *субъективно* выбираемых методов. Все прочее время она пребывает в состоянии «потенциальном» и представлена как данные. В отличие от данных — результатов регистрации объективно существующих сигналов, вызванных изменениями в материальных телах или полях, методы могут быть субъективными, поскольку их создание и использование предопределяются целями субъекта.

[1] Данные, составляющие информацию, обычно имеют свойства, однозначно предопределяющие метод ее получения, что иногда и приводит к отождествлению этих понятий.

1.2. Информационные коммуникации и основы формализованного представления информации

1.2.1. Процессы воспроизводства информации и знания

Информационные системы являются искусственными и создаются человеком для конкретных задач в области его основной деятельности. При этом оборот информации, как и всякого продукта в человеческой деятельности, подчиняется естественному циклу «*создание—передача—потребление*». Например, в традиционном цикле информационного обмена документальной информации основной поток идет по цепочке *автор—издательство—библиотека—читатель*, хотя существуют и другие пути, например: *автор—читатель*; *издательство—читатель* (подписка).

Новое знание (сведения о результатах деятельности) воплощается обычно в форме сообщения — документа, фиксирующего знания субъекта, реализующего деятельность. Такая «материализация» знаний обеспечивает унифицированную форму обобществления личного знания, а сравнительно низкая стоимость тиражирования (документов, описывающих процесс в форме, обеспечивающей ее воспроизведение, а не субъекта, способного повторить результат) позволяет существенно расширить сферу потенциальных потребителей. Однако чтобы опубликованное сообщение стало стимулом для построения нового знания приемником, сообщение должно быть не только воспринято (выделено среди других), но также понято (выделен смысл) и вписано в систему наличного знания приемника или сохранено в долговременной памяти.

В свою очередь, формирование сообщения явно или неявно связано с выбором или введением специальной терминологии (знаковой системы или системы кодирования), что часто сказывается на адекватности передачи смысла.

То есть процессы обработки и поиска информации не могут рассматриваться изолированно от процессов основной деятельности, поскольку обусловлены ими, а действенность информации обуславливается «коммуникационными» возможностями ИС (способами и средствами представления информации).

Процесс решения любой научной или практической задачи, где, так или иначе, возникает или используется информация, независимо

от того, автоматизирован он или нет, в общем случае включает следующие этапы.

1. *Поиск сообщений.* Создатель нового знания обращается к информационным ресурсам для получения информации, которая может быть использована (в частности, заимствована в качестве решения) им, например, как концептуальная основа, экспериментальные, вспомогательные или опровергаемые данные и т. д. По отношению к среде он будет выступать в качестве потребителя информации, а информационная среда будет источником сообщений. При этом сообщения могут быть получены либо в виде услуги специальных *информационных систем*, обеспечивающих хранение и поиск информации в различных хранилищах, либо по другим каналам, например, путем личного общения, непосредственным обращением к результатам других исследований, в том числе еще не представленных в виде сообщений, и т. д.

2. *Интерпретация сообщений.* Вследствие уникальности конкретных условий решаемых задач язык полученного сообщения (в общем случае) может не совпадать с «внутренним языком» создателя информации. Данный этап включает адаптацию сообщений, интерпретацию их содержания в терминах «внутреннего языка», а в итоге — извлечение из сообщений сведений, необходимых для решения поставленной задачи. Результат этапа — информационное обеспечение решаемой задачи, которое должно привести к повышению эффективности ее решения.

3. *Решение задачи.* На данном этапе, используя информационное обеспечение, а также собственные знания, прилагая определенные усилия, субъект, решая задачу, создает новую информацию. Эта информация зафиксирована на языке задачи, является достоянием достаточно ограниченного множества лиц (организаций), связанных с конкретной разработкой и, как правило, для использования за пределами конкретной задачи будет требовать дополнительных затрат труда.

4. *Создание сообщений.* На данном этапе осуществляется интерпретация полученного результата на «языке коммуникаций», т. е. подготавливается сообщение в «стандартной» форме, одной из тех, которые приняты на данном этапе развития предметной области вообще и информационных коммуникаций в частности. Результатом этого этапа может быть статья, техническая документация, сообщение по электронной почте и т. д. Новая информация, оформленная как сообщение, уже представляет собой потенциальную обществен-

ную ценность для большого круга пользователей и, соответственно, для решения других задач.

5. *Распространение сообщений*. На этом этапе создатели сообщений вступают во взаимодействие с системой информационных коммуникаций, затрачивая определенные усилия (в основном организационного характера) по «вводу» сообщения в один (или несколько) из доступных каналов коммуникации (публикация, аудиторное выступление и т. д.). Эффективность данного этапа определяется как степенью усилий субъекта познания, так и теми возможностями, которые ему предоставляет система коммуникации и, в частности, АИС.

Общее представление о взаимодействии потребителей-поставщиков информации в процессе воспроизводства знаний иллюстрируется схемой на рис. 1.6. Хотя очевидно, что перечисленные этапы могут реализоваться сложным последовательно-параллельным образом, не обязательно все и в указанной последовательности.

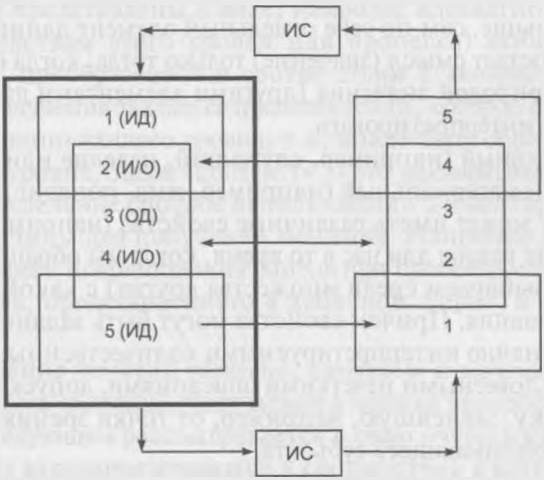


Рис. 1.6. Каналы взаимодействия потребителей-поставщиков информации

Первый и пятый этапы являются этапами собственно информационной деятельности (ИД), поскольку их эффективность во многом определяется свойствами конкретных коммуникаций и информационных систем. Третий этап — собственно основная деятельность. Этапы второй и четвертый носят пограничный, диффузный характер и могут относиться как к основной, так и к информационной деятельности.

Формально на рис. 1.6 представлена схема информационного обмена. Реально же пользователь работает с источником информации по схеме информационного обслуживания, для которой характерна опосредованность, «разорванность» взаимодействия: сообщения «отчуждаются» от автора[1]. Это приводит к «рассеянию» информации, и поэтому для эффективного отыскания публикаций необходимо создавать и использовать специальные справочно-поисковые средства.

1.2.2. Представление информации и организация данных

Если бы назначением информационных систем было только хранение и поиск данных в массивах записей, то структура системы и базы данных была бы простой. Причина сложности в том, что практически любой объект характеризуется не только параметрами-величинами, но и взаимосвязями частей или состояний. Кроме того, как отмечалось выше, сам по себе отдельный элемент данных (его величина) приобретает смысл (значение) только тогда, когда связан с контекстом — природой значения (другими элементами данных), что и позволит его интерпретировать.

Материальный (например, служащий, изделие или населенный пункт) или нематериальный (например, имя, понятие, абстрактная идея) объект может иметь различные свойства (например, цвет, вес, имя), которые важны для нас в то время, когда мы обращаемся к нему (например, выбираем среди множества других) с какой-либо целью его использования. Причем свойства могут быть заданы как отдельными, однозначно интерпретируемыми количественными показателями, так и словесными нечеткими описаниями, допускающими разную трактовку, зависящую, например, от точки зрения и наличных знаний воспринимающего субъекта.

Общим же фактором является то, что человек, работая с информацией, имеет дело с *абстракцией,* представляющей интересующий его фрагмент реального мира, — той совокупностью *характеристических свойств (атрибутов)*, которые важны для решения его прикладной задачи. Абстрагирование — это способ *упрощения* совокупности фактов, относящихся к реальному объекту (по своей сути бесконечно

[1] Соответственно, источник информации ассоциируется уже с сообщением — носителем информации, а не с человеком или системой, которые являются источником в прямом смысле этого слова.

сложному и разнообразному). При этом некоторые свойства объекта игнорируются, поскольку считается, что для решения данной задачи (или совокупности задач) они не являются определяющими и не влияют на конечный результат. То есть задача процесса абстрагирования — это построение конструктивного операбельного описания (рабочей модели), удобного в обработке как для человека, так и для машины, позволяющего спроектировать и реализовать эффективную обработку информации, причем высокопроизводительной должна быть работа не только вычислительной системы, но и взаимодействующего с ней человека.

В общем случае для сложных систем с многоуровневым представлением семантики эффективность обработки достигается через специализацию объектов или процессов путем сведения обрабатываемых объектов к таким, которые имеют однородную природу и форму представления. Это означает, что для реализации эффективного межуровнего (межкомпонентного) взаимодействия (на каждом из которых объекты представлены в виде, наиболее адекватном функциональным средствам этого уровня или процесса) любая величина должна быть преобразована в соответствии с «контекстом» этого уровня для получения такого ее представления, которое будет «значимо» для воспринимающего уровня, т. е. может быть обработано средствами этого уровня. Здесь «контекст» — это декларативное или процедурное определение способа использования элементарных составляющих величины для получения значения. Например, контекстом является порядок использования байтов при преобразовании вещественного числа, представленного в двоичной форме, в символьный формат.

Соотношение понятий *величина*, *контекст* и *значение* приведено на рис. 1.7. Здесь значение, получаемое в первом процессе (на первом уровне), в следующем рассматривается в свою очередь как величина, которая будет интерпретироваться в соответствии с контекстом своего процесса[1] .

Таким образом, можно сказать, что значение определяется парой <*величина, контекст* >. Причем поскольку *контекст* и *величина,* в общем случае, имеют разную природу, они должны быть представлены в вычислительной среде самостоятельными, скорее всего, разнотипными объектами.

---

[1] Соотношение понятий «величина» и «значение» аналогично соотношению понятий «данные» и «информация».

Рис. 1.7. Соотношение понятий «величина», «контекст» и «значение»

В этом смысле (с точки зрения способа представления и, соответственно, восприятия) в отдельный класс можно выделить *фактографическую информацию:* такое представление реально существующих событий и явлений, когда они могут быть описаны как *факты*, задаваемые парой <*имя, значение*>, где *имя* — знак, уникально определяющий (идентифицирующий) факт в заданной предметной области и обычно не нуждающийся в явном определении или доопределении его существа, а *значение* — характеристика, задающая одно из множества возможных состояний.

То есть здесь факт (его значение) задается величиной, например, числовой для физически измеримых параметров и логическими величинами «истина»/«ложь» для указания, свершилось событие или нет[1].

Можно сказать, что особенностью фактографической информации является практическая очевидность (минимальная неопределенность, не требующая использования сложных или нечетких процедур) идентификации и интерпретации «факта» (его имени и состояния). И в этом случае контекст в достаточной степени определяется однозначно понимаемым объявлением о назначении данных и таким именованием полей данных, когда в качестве имени используется общепринятое, не зависящее от прикладных задач *имя свойства* (и таким образом определяются характеристические признаки). Именно такое состояние предопределяет для пользователя возможность адекватного восприятия содержания: способ интерпретации данных в этом случае практически всегда будет однозначным, причем для пользователя *определение способа* происходит обычно *неявно* (не требует от него явных действий для определения и использования контекста). Это,

[1] Следует отметить, что такая форма в наибольшей степени соответствует машинным формам представления информации.

с одной стороны, позволяет свести представление предметной области к точной теоретико-множественной модели, а с другой — обусловливает возможность непосредственного использования данных в задачах обработки (на уровне прикладных программ) для генерации новой информации без участия субъекта (человека, внешнего по отношению к машинной среде), обеспечивающего определение и использование контекста.

Однако большинство задач, решаемых человеком, не могут быть сведены к «фактографическому» представлению и описываются (и, соответственно, представляются в машинной среде) средствами естественного или специализированного языков, оперирующих *лингвистическими переменными*, значение которых может зависеть не только от контекста предметной области, но также и от контекста ближайшего окружения — значения соседних переменных. Причем появление нового смысла (факта) не обязательно приводит к появлению новой переменной: новый факт может быть представлен с помощью уже существующих переменных. Например, словесные определения математических или географических понятий.

В отличие от ранее рассмотренного фактографического представления, для вербальной формы представления факта (выражениями языка, использующего лингвистические переменные) характерно то, что для задания *имени*, *значения* и *контекста* могут использоваться единый способ и средства — элементы одного языка. Например, описание весовых свойств может быть представлено несколькими, но имеющими один смысл вариантами предложений: «Чугунная заготовка весом 29 килограмм» или «Чугунная заготовка имеет свойство $m = 29$, где m — вес в килограммах» и т. п.

Автоматическое приведение такого рода представлений к очевидно наилучшей для этого случая табличной форме потребовало бы применения трудно реализуемых процедур морфологического и семантического анализа. Но, с другой стороны, выделение смысла (и генерация новой информации) обычно производится человеком, сознание которого (как среда преобразования) ориентировано именно на обработку лингвистических переменных.

Случаи, когда информация представляется в форме, не адекватной двоичной форме (требующей дополнительных преобразований), могут быть обусловлены разными факторами. Рассмотрим следующие случаи.

1. Точная (однозначно понимаемая) информация представляется в специальном (в частности, графическом) формате. Например,

структурные химические формулы, конструкторская документация и т. д. В этом случае для автоматической обработки требуются узкоспециализированные средства, что приводит к общей *неунифицированности* представления на уровне данных (в данном примере — графических примитивов).

2. Информация, точная по содержанию, но вариантно представляемая по форме. Например, описание в текстовом виде численно задаваемых параметров изделия. Лингвистические переменные в этом случае имеют точное значение, однако построение универсальной процедуры автоматического выделения факта из текста трудоемко.

3. Слабоструктурированная информация, обычно представляемая в текстовой форме. Например, учебная или научная публикация, где новые понятия строятся на основе ранее определенных. В этом случае значения лингвистических переменных могут принимать новые, ранее не определенные значения, которые определяются контекстом — ближним (словосочетания) или общим (темой сообщения).

Важно также отметить, что структурированность относится не только к форме представления данных (формат, способ хранения), но и к *способу интерпретации значения пользователем*: значение параметра не только представлено в предопределенной форме, но и обычно сопровождается указанием размерности величины, что позволяет пользователю понимать ее смысл без дополнительных комментариев. Таким образом, можно сказать, что фактографические данные предполагают возможность их *непосредственной* интерпретации.

Однако атрибутивный способ в «чистом виде» неэффективен для идентификации *слабоструктурированной информации*, связанной с объектами, имеющими обычно *идеальную* (умозрительную) природу, — категориями, понятиями, знаковыми системами. Такие объекты зачастую определяются опосредованно — через другие объекты, для чего используются естественные или искусственные языки (например, язык математики). Соответственно, для понимания смысла пользователю необходимо использовать соответствующие правила языка и, более того, часто необходимо уже иметь данные, позволяющие идентифицировать и связать получаемую информацию с наличным знанием. То есть процесс интерпретации такого рода данных имеет *опосредованный* характер и требует использования дополнительной информации, которая, в общем случае, не обязательно присутствует в формализованном виде вместе с сообщением (данными), собственно являющимся информацией. Можно сказать, что основным

отличием документальных ИС является опосредованный способ интерпретации данных, а не их организация.

Поэтому физическому размещению данных (и, соответственно, определению структуры физической записи) должно предшествовать описание логической структуры предметной области — построение *модели* соответствующего фрагмента реального мира, выделяющей только те объекты, которые будут нужны будущим пользователям и представленные только теми параметрами, которые будут значимы при решении прикладных задач. Такая модель имеет очень мало физического сходства с реальностью, но будет полезна как *представление* пользователя о реальном мире. Важно, что это представление описывается *удобными для пользователя* средствами, но при этом будет обеспечивать возможности манипулирования в *неадекватной человеку* жесткой вычислительной среде с числовым представлением информации.

Таким образом, прежде чем описывать физическую реализацию объектов и связей между ними, необходимо определить:

1) способ, с помощью которого пользователи представляют (описывают) объекты и связи;

2) форму и методы внутримашинного представления элементов данных и взаимосвязей;

3) средства, обеспечивающие взаимно однозначные преобразования внешнего и внутримашинного представлений.

Такой подход является компромиссом за счет *предварительно определяемого множества абстракций*, общих для большинства задач обработки данных, обеспечивающих возможность построения *надежных* программ обработки. Пользователь, используя *ограниченное множество формальных, но достаточно знакомых понятий*, выделяя сущности и связи, описывает объекты и связи предметной области; программист, используя такие *типовые абстрактные понятия* (как, например, числа, множества, агрегаты данных), определяет соответствующие информационные структуры. Система управления данными, используя *двоичные формы представления типизированных* данных, обеспечивает эффективные процедуры хранения и обработки данных.

На всех этих уровнях используется абстрактное понятие *структура*:

- *структура информации* — схематичная форма (обеспечивающая переход к атрибутивная форме) представления сложных композиционных объектов и связей реальной предметной области,

выделяемых как актуально необходимые для решения прикладных задач, в общем случае без учета того, будут ли для ее решения использованы средства программирования и вычислительные машины. В случае слабоструктурированного (документальная форма) представления ПрО необходимо рассматривать еще и *структуру текста*, обеспечивающую возможность выделения семантически значимых компонентов текста и определения их роли. Эффективность здесь определяется уровнем абстрагирования, а также полнотой и точностью представления свойств посредством выбранной системы характеристик;

- *структура данных* — атрибутивная форма представления свойств и связей ПрО, ориентированная на выражение описания данных средствами формальных языков (т. е. учитывающая возможности и ограничения конкретных средств с целью сведения описаний к стандартным типам и регулярным связям). Эффективность в этом случае связывается с процессом построения программы («решателя» прикладной задачи) и, в каком-то смысле, с эффективностью работы программиста;
- *структура записей* — учитывающая особенности физической среды реализация способов хранения данных и организации доступа к ним как на уровне отдельных записей, так и их элементов. Эффективность в этом случае связывается с процессами обмена между устройствами оперативной и внешней памяти и обеспечивается избыточностью данных, искусственно вводимой для обеспечения функциональной эффективности отдельных операций (например, поиска по ключам).

Структура является общепринятым и удобным инструментом, одинаково эффективно используемым как на уровне сознания человека при работе с абстрактными понятиями, так и на уровне логики машинных алгоритмов. Структура позволяет простыми способами свести многомерность содержательного описания к линейной последовательности записей, а в ряде случаев именно описание структуры представляет собой контекст использования данных.

1.2.3. Идентификация и поиск данных

В задачах обработки информации, и в первую очередь в алгоритмизации и программировании, атрибуты *именуют* (обозначают) и присваивают им *значения*.

При обработке информации мы имеем дело с совокупностью объектов, *информацию о свойствах* каждого из которых надо сохранять (записывать в некоторую ограниченную адресуемую область носителя) как *данные*, чтобы при решении задач их можно было найти и выполнить необходимые преобразования.

Таким образом, как представлено на рис. 1.8, любое состояние объекта характеризуется совокупностью актуализированных (имеющих некоторое значение в этот момент времени) атрибутов[1], которые фиксируются на некотором материальном носителе в виде *записи — группы* (совокупности) формализованных *элементов данных* (значений атрибутов, представленных в том или ином формате).

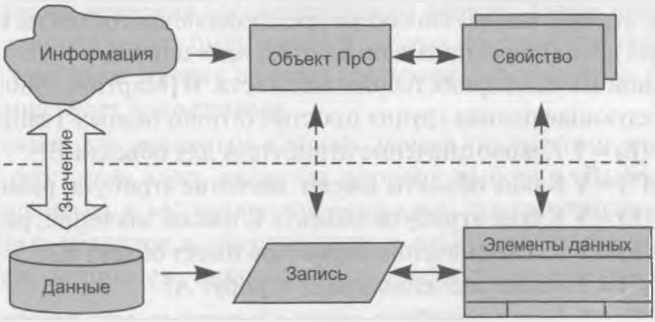


Рис. 1.8. Атрибутивный способ идентификации

В контексте задач хранения и поиска можно говорить, что значение атрибута *идентифицирует* объект: использование значения в качестве поискового признака позволяет реализовать простой критерий отбора по условию сравнения[2]. В общем случае можно говорить, что отдельный объект уникален (уже хотя бы потому, что мы *именно его* выделяем среди других). Соответственно, запись, содержащая данные о нем, также должна быть узнаваема однозначно (по крайней мере, в рамках предметной области), т. е. иметь уникальный идентификатор, причем никакой другой объект не должен иметь такой же идентифи-

[1] В общем случае объект может описываться совокупностью записей, относящихся к его составным частям или отражающих динамику изменения состояния.

[2] Следует отметить некоторые семантические проблемы идентификации через значение атрибута. Значение атрибута идентифицирует запись о *состоянии* объекта (а не о нем самом), и в случае изменения значения, например, табельного номера служащего, будет невозможно ответить на вопрос: идет ли речь о том же служащем или о новом.

катор. Поэтому если в качестве идентификатора используется значение элемента данных, то в некоторых случаях для обеспечения уникальности требуется использовать более одного элемента. Например, для однозначной идентификации записей о дисциплинах учебного плана необходимо использовать элементы СЕМЕСТР и НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ, так как одна дисциплина может быть прочитана в разных семестрах.

Таким образом, атрибутивный способ идентификации предопределяет, что в отличие от логики сознания человека, где вопросы могут иметь форму «как», «почему», предполагающую развернутую форму ответа, машинная логика вопроса может обрабатывать только вопросы типа «есть ли» (выполняется ли условие, сводящееся к соотнесению величин, атрибутивным способом представляющих объекты в вычислительной среде), предполагающая ответ «да» или «нет», т. е. в форме, основанной на категориях теории множеств. В [Мартин, 1980] приводится следующая полная группа простых (атрибутивных[1]) запросов:

1). **A(E) = ?** Каково значение атрибута A для объекта E?

2). **A(?) = V** Какие объекты имеют значение атрибута, равное V?

3). **?(E) = V** Какие атрибуты объекта E имеют значение, равное V?

4). **?(E) = ?** Какие значения атрибутов имеет объект E?

5). **A(?) = ?** Какие значения имеет атрибут A?

6). **?(?) = V** Какие атрибуты каких объектов имеют значение, равное V?

Соответственно, технологии (алгоритмы) поиска основываются на двух типах *организации массива* объектов поиска — *прямой* и *инвертированной*.

В случае прямой организации массива записей (хранилища документов) документы обычно размещаются в порядке их поступления. При прямой организации поиск в больших массивах будет требовать достаточно много времени, так как для сравнения с запросом надо последовательно выбирать *все*[2] документы из хранилища, по той простой причине, что не обратившись к документу, мы не можем судить о его содержании.

---

[1] Здесь в запросах типов 2, 3, 6 вместо оператора равенства может быть использован другой оператор сравнения (*больше, меньше, не равно* или другие).

[2] Конечно, перебор можно завершить, когда будет получен документ, отвечающий в рамках известного и даже уже ставшего традиционным подхода на все вопросы реальной ИП, однако вполне возможно предположить, что мы при этом уже не дойдем до документов, опубликовавших новейшие достижения, опровергающие ранее принятое решение.

Инвертированная форма организации массива[1] предполагает, что записи могут быть упорядочены, например, разбиты на подмножества, которые, в свою очередь, упорядочены в соответствии с величиной атрибута. Такое упорядочение записей (обычно виртуальное, так как нерационально переписывать и сортировать исходный массив) сопровождается построением вспомогательной структуры — *инвертированного справочника*, в котором с каждым *индексом* (*ключом* — идентификатором подмножества) связан список ссылок на документы, отнесенные к этому подмножеству. Например, систематический или авторский каталоги библиотеки имеют типично инвертированную организацию: карточки с шифром раздела представляют собой индекс, а инвертированный список — это распложенные за ней карточки с шифрами хранения соответствующих единиц. Таких инвертированных списков может быть много и разных по атрибуту упорядочения, но при этом массив единиц хранения будет один и порядок размещения будет тоже единым.

Неявным, но основным с точки зрения реализации алгоритма поиска, фактором здесь является порядок выборки. Выборка может проводиться в «естественном» порядке, соответствующем расположению объектов в массиве, или в «искусственном», соответствующем, например, некоторой классификации предметной области.

И в том и в другом случае имеем дело с *перебором* объектов, выбираемых для сравнения из хранилища. Таким образом, рациональность построения процедуры поиска зависит от длины перебора, что в свою очередь определяется как характеристиками хранимых объектов (в первую очередь — размерами записей), так и характером запросов. Соответственно оптимизация достигается сокращением перебора — длины *последовательно* проверяемого массива.

В качестве ключа, обеспечивающего доступ к записи, можно использовать идентификатор — отдельный элемент данных, представляющий соответствующий атрибут или совокупность элементов (составной ключ).

---

[1] Определяющим в понятии «*прямая организация*» является не характер размещения записей — единиц хранения, а размещение *содержания (величин атрибутов)*, которое представлено изначальной «естественной» *последовательностью величин*, а в случае инвертированной организации содержание массива будет представлено теми же величинами (но самостоятельно, отдельно от контекста), упорядоченными, например, по возрастанию (по алфавиту), при этом положение в записи будет указано параметром.

При этом ключ может храниться в составе записи или отдельно. Например, ключ для записей, имеющих неуникальные значения атрибутов, для устранения избыточности целесообразно хранить отдельно. На рис. 1.9 приведены два таких способа хранения ключей и атрибутов для набора простейшей структуры.

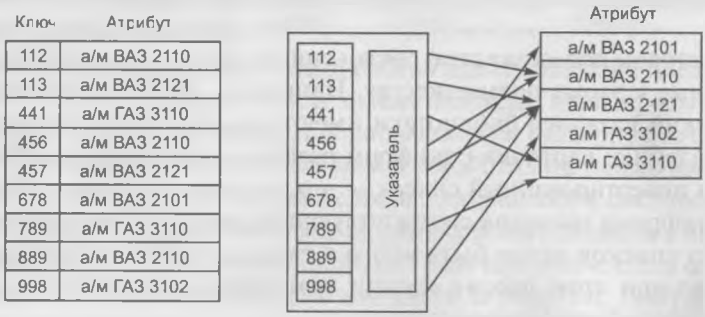


Рис. 1.9. Способы хранения ключа и атрибута

Один из способов использования ключа в качестве входа — организация инвертированного списка, каждый вход которого содержит значение ключа вместе со списком идентификаторов соответствующих записей. Данные в инвертированном списке располагаются обычно в возрастающем порядке, поэтому алгоритм нахождения нужного значения довольно прост и эффективен. После нахождения значения запись локализуется по указателю физического расположения. Недостатком индекса является то, что он занимает дополнительное пространство и его надо обновлять каждый раз, когда удаляется, обновляется или добавляется запись. На рис. 1.10 приведен инвертированный список для предыдущего примера.

В общем случае инвертированный список может быть построен для любого ключа, в том числе составного.

Пример, приведенный на рис. 1.9, представляет прямую организацию массива. Второй способ является инверсией первого, он соответствует рис. 1.10. Прямая организация массива удобна для поиска

а/м ВАЗ 2110	678
а/м ВАЗ 2110	112, 456, 889
а/м ВАЗ 2121	113, 457
а/м ГАЗ 3102	998
а/м ГАЗ 3110	441, 789

Рис. 1.10. Инвертированный список для ключа «Марка автомобиля»

по условию «Каковы свойства указанного объекта?», а инвертированная — для поиска по условию «Какие объекты обладают указанным свойством?»

Запросы типа 1 выполняются поиском по «прямому» массиву: доступ к записи производится по первичному ключу. Запросы типа 2 выполняются поиском по инвертированному списку: доступ к записи(ям) производится по указателю, выбираемому из списка по значению вторичного ключа. Ответом в этих случаях будет *значение* атрибута или идентификатора. Запросы типа 3 имеют ответ — *имя* атрибута.

Запросы типа 2, 5, 6 относятся к нескольким атрибутам, и в этом случае могут быть построены несколько индексов, облегчающих поиск по этим ключам.

1.2.4. Методы и средства идентификации объектов

Как отмечалось ранее, объект имеет различные свойства (например, цвет, вес, имя), которые важны в то время, когда к объекту обращаются с целью какого-либо его использования. Причем свойства могут быть заданы как отдельными, однозначно интерпретируемыми количественными показателями, так и словесными нечеткими описаниями, допускающими разную трактовку, иногда зависящую от точки зрения и наличных знаний воспринимающего субъекта.

В различных ситуациях возникает необходимость идентификации конкретного объекта либо группы объектов, обладающих некоторым множеством общих свойств.

На рис. 1.11 приведена типология задач идентификации объектов.

Задачи уникальной идентификации объекта можно условно разделить на две группы: выделить объект для определения или описания его персональных (индивидуальных) характеристических свойств в рамках конкретной предметной области (назовем такую идентифи-

Рис. 1.11. Типология задач идентификации объекта

кацию сущностной) и выделить объект, выполняющий в данное конкретное время некоторую уникальную функцию (назовем такую идентификацию функциональной).

При решении задач первой группы глубину и способ уникальной идентификации связывают с предметной областью, в рамках которой рассматривается объект: например, серия и номер паспорта могут однозначно идентифицировать любого (достигшего определенного возраста) гражданина данного государства; студента данного вуза однозначно идентифицирует номер студенческого билета; вкладчика банка — номер счета в этом банке. В общем случае это может быть одно и то же физическое лицо, поэтому обычно стремятся использовать минимальный (необходимый и достаточный) набор характеристических свойств, позволяющий построить уникальный идентификатор. В то же время такой идентификатор должен быть минимально подвержен изменениям.

Задачи функциональной уникальной идентификации возникают обычно в том случае, когда необходимо однозначно указать объект в некотором упорядоченном множестве ему подобных (например, номер вагона в отходящем по расписанию поезде). Такая идентификация не является уникальной идентификацией объекта в полном смысле этого слова, так как поезд каждый раз может быть составлен заново, и на месте вагона с номером N реально могут стоять разные физические вагоны. Тем не менее она должна позволять однозначно ориентироваться при выборе объекта. Задачи функциональной идентификации чаще всего решаются путем присваивания объекту порядкового номера в соответствии с установленным отношением порядка, что обеспечивает простую и короткую идентификацию объекта.

Основной недостаток любой уникальной идентификации — ее неинформативность, т. е. отсутствие каких-либо явных признаков (атрибутов), характеризующих объект с содержательной стороны.

В основе идентификации групп объектов может использоваться один из следующих методов:

- классификационный;
- описательный;
- смешанный.

Классификационная идентификация ориентирована на применение специализированных условных обозначений для объектов, у которых выделенные свойства имеют одинаковые значения. В основе такой идентификации лежит использование мнемонических или классификационных кодов, однозначно характеризующих объект.

Мнемонический код предполагает однозначную расшифровку значений выделенных свойств объекта. Например, условное обозначение «Электронасос ГНОМ 100-25» наряду с обозначением объекта «электронасос» включает мнемоническое обозначение следующих свойств: Г — для грязной воды, Н — насос, О — одноступенчатый, М — моноблочный, 100 — с подачей 100 $м^3$ /ч, 25 — с напором 25 м.

Классификационный код устанавливает взаимно-однозначное соответствие характеристики объекта стандартным кодификаторам и классификаторам. Например, по специальному кодификатору организаций некоторой продукции может быть присвоен четырехзначный буквенный код организации-разработчика; по классификатору конструкторской документации — код классификационной характеристики и т. п.

Таким образом, классификационные методы обеспечивают систематизацию объектов в соответствии с некоторой заданной классификационной схемой. Код, присвоенный отдельному классу (равно как и мнемоническое обозначение), обеспечивает его полную идентификацию в рамках конкретного классификатора.

Описательные методы идентификации используются, как правило, в тех случаях, когда необходимо идентифицировать конкретный объект или группу объектов путем описания произвольного набора его характеристик. Описательный метод предполагает наряду с указанием классификационных характеристик выделение дополнительных наборов свойств, углубляющих характеристику объекта и сужающих область поиска.

В ряде случаев для идентификации объектов используются ссылки на нормативные документы, содержащие описания конкретных характеристик (свойств, показателей, отличительных признаков). Тогда идентификация объекта включает наименование объекта и ссылку на документ, содержащий требования к этому объекту, например, «Кислота соляная по ГОСТ 3118—77»; «Шины пневматические для легковых автомобилей по ГОСТ 4754—80».

Одним из основных преимуществ описательного метода идентификации является возможность осуществления сопоставительного анализа однородных (родственных) объектов путем сравнения характеристик, вошедших в их идентификацию. Такое сравнение позволяет выбрать объект, обладающий наилучшими характеристиками для заданных условий применения или обеспечивающий полную замену другого.

Описательные методы идентификации широко используются в медицине (описание течения и симптомов болезней в медицинской

карте), в криминалистике (описание преступника и характера преступления), в геологии (описание минерала) и т. п.

Смешанные методы идентификации предполагают использование при характеристике предмета возможностей и преимуществ как классификационных, так и описательных методов. Наряду с произвольным многосторонним описанием объекта могут быть заданы его атрибуты, определяющие принадлежность к определенному классу некоторой классификационной схемы, а также ссылки на нормативный документ, где помещены его характеристики.

1.3. Концептуальные основы, состав и структура информационной системы

1.3.1. Основные понятия и определения

Для определения состава и взаимосвязей компонентов системы приведем предварительно определения следующих основных понятий.

Система (от греч. *systema* — целое, соединение, составленное из частей) — совокупность элементов, взаимодействующих друг с другом и образующих определенную целостность.

Целостность системы — проявление свойства *эмерджентности*, отражающего принципиальную несводимость свойств системы к сумме свойств отдельных ее элементов и в то же время зависимость свойств каждого элемента от его места и функции внутри системы.

Элемент системы — часть системы, имеющая определенное функциональное назначение. При этом отдельный элемент какой-либо системы (как и сама система) может также быть элементом другой системы. Сложные элементы систем, в свою очередь состоящие из взаимосвязанных более простых элементов, называют *подсистемами*.

Организация системы — внутренняя упорядоченность, согласованность взаимодействия элементов системы, проявляющаяся, в частности, в ограничении разнообразия состояний элементов системы.

Состояние системы — множество существенных свойств, которыми обладает система.

Структура системы — состав, порядок и принципы взаимодействия элементов системы, определяющие основные свойства системы.

Структура — это та часть свойств, которая остается в системе неизменной при изменении ее состояния.

Архитектура системы — совокупность свойств системы, существенных для организации взаимодействия ее составляющих.

С точки зрения формы существования системы выделяют материальные и абстрактные системы.

Материальные системы — энергоматериальные системы, обеспечивающие реальную обработку информации, представленную на материальных носителях. Подразделяются на технические, эргатические и эргатехнические (смешанные). Именно эргатехнические системы — материальные системы «человек—машина», состоящие из эргатического элемента (человека) и технического элемента (машины), будут составлять основной предмет изучения.

Абстрактные системы — это системы, которые имеют в качестве операционных объектов преимущественно идеализированные, например, знания, теории, гипотезы.

Информационная система (ИС) — материальная система, организующая, хранящая и преобразующая информацию. Это система, основным предметом и продуктом функционирования которой является информация.

Система обработки данных (СОД) — комплекс взаимосвязанных методов и средств преобразования данных, представленных в формализованном виде, пригодном для автоматической обработки при возможном участии человека.

Системы обработки знаний (СОЗ) — автоматизированные ИС, имеющие специальное программное обеспечение для логической (семантической) обработки информации.

Интеллектуальная информационная система — информационная система, обладающая способностями упорядочивать массив сведений по степени существенности; извлекать из массива все возможные сведения как следствия, выводимые посредством логики; формировать новые типы вопросов в ответ на получение новой информации из внешнего мира; обладать «рефлексией», т. е. способностью к оценке хранимых сведений [Смирнов, 1981].

В основу построения автоматизированных ИС, основным назначением которых является обеспечение эффективности *специализированной* обработки данных, положены следующие принципы:

- *принцип интеграции* — обрабатываемые данные, однажды введенные в систему, многократно используются для решения воз-

можно большего числа задач, чем максимально устраняется дублирование данных и операций их преобразования;

- *принцип системности* — возможность обработки данных в различных «разрезах» с целью получения информации, необходимой для принятия решений на всех уровнях и во всех функциональных подсистемах;
- *принцип комплексности* — автоматизация процедур преобразования данных на всех стадиях технологического процесса.

1.3.2. Классификация информационных систем

Понятие автоматизированной системы тесно связано с эффективностью обработки, в основе которой — специализация компонентов и процессов. Такая специализация обычно обусловлена свойствами обрабатываемых объектов и задачами, предопределенными целями системы.

С точки зрения назначения и применения ИС могут классифицироваться по следующим признакам:

1) по характеру использования результатной информации:
   - информационно-поисковые, обеспечивающие сбор, хранение, выдачу информации по запросу пользователя;
   - информационно-советующие, используемые в качестве систем поддержки принятия решений;
   - информационно-управляющие, реализующие непосредственное управление процессом или сложным объектом;
2) по области (сфере) применения:
   - производственные;
   - научно-исследовательские;
   - библиотечные АИС, финансовые, офисные ИС и т. д.;
3) по объектам управления:
   - автоматизированного проектирования;
   - управления технологическими процессами;
   - управления предприятием[1] и т. д.;
4) по степени автоматизации процессов обработки:
   - с ручной обработкой информации;
   - механизированной обработки информации;

[1] Обычно функциями ИС, управляющей крупным предприятием, являются следующие: вычислительная, коммуникационная, запоминающая, следящая, регулирующая, оптимизационная, прогнозирующая, анализирующая, контролирующая, документирующая.

- автоматизированной обработки информации;
- автоматической обработки информации;

5) по степени специализированности возможного применения:

- универсальные;
- специализированные (проблемно-ориентированные).

Другим аспектом типологии ИС является собственно «информационный», где признаки классификации определяются особенностями технологий обработки информации, свойства которой и переносятся на системы. С этой точки зрения информационные системы могут классифицироваться по следующим признакам.

1. По типу хранимой информации можно выделить (исключая мультимедийную информацию) фактографические, документальные, лексикографические ИС.

Фактографические системы ориентированы на обработку данных, контекст использования которых предопределен и обычно зафиксирован в схеме данных или в процедурах обработки.

Документальные ИС подразделяются по уровню представления информации — *полнотекстовые* (обрабатывающие так называемые «первичные» документы) и *библиографическо-реферативные* (обрабатывающие «вторичные» документы, отражающие на адресном и содержательном уровне первичный документ). Контекст использования данных в документальных ИС может быть иным, чем тот, который был определен при ее создании.

Лексикографические — это классификаторы, кодификаторы, словари основ слов, тезаурусы, рубрикаторы и т. д., которые обычно используются в качестве справочных совместно с документальными или фактографическими БД и позволяют в том числе доопределить контекст данных.

2. По типу модели данных СУБД, используемой для реализации информационной системы, выделяют традиционно три класса: *иерархические, сетевые, реляционные*.

3. По топологии хранения данных различают *локальные* и *распределенные* ИС.

4. По оперативности использования данных можно выделить *операционные* и *справочно-информационные*. К последним можно отнести АИС ретроспективной информации (электронные каталоги библиотек, БД статистической информации и т. д.), которые используются для поддержки основной деятельности и обычно не предполагают внесение изменений в уже существующие записи. Операционные ИС предназначены для оперативного отражения состояния и управления

объектами и технологическими процессами ПрО. В этом случае данные не только извлекаются из БД, но также изменяются и добавляются, в том числе в результате их использования.

5. По степени доступности информации ИС можно подразделить на *общедоступные* и те, которые имеют *ограничения на доступ* пользователей к ресурсам системы. В последнем случае говорят об авторизованном доступе, индивидуально определяющем не только набор данных, но и операции, которые доступны конкретному пользователю.

Следует отметить, что представленная классификация не является исчерпывающей; она скорее отражает исторически сложившееся состояние дел в сфере деятельности, связанной с разработкой и применением информационных технологий и систем.

1.3.3. Основные компоненты информационной системы

Основной и определяющей составляющей любой системы являются функционально взаимосвязанные *комплексы данных и процедур* их обработки. Отметим, что эти комплексы ни по отдельности, ни вместе еще не создают той *целостности*, которая свойственна системам. Системные свойства проявляются, когда ИС рассматривается в динамике взаимосвязи со средой, т. е. когда существенными становятся факторы управляемости и адаптивности к изменяющимся внешним условиям, устойчивости во времени. Именно поэтому любая система, помимо функциональных компонент — основных с точки зрения назначения системы, необходимо включает организационные и обеспечивающие компоненты, назначением которых является создание необходимых условий для функционирования, и в том числе формирование субъектов управления. В свою очередь, ИС — это составная часть некоторой большей системы, обеспечивающей достижение какой-либо конкретной цели в деятельности человека.

Практически все современные ИС включают в свой состав вычислительные комплексы, которые составляют *физический компонент* системы. Такими компонентами являются как внешняя память, так и технические и вычислительные средства, обеспечивающие непосредственно обработку и взаимодействие пользователя с ИС.

Второй компонент — это *программные средства* (процедуры) и технологии, обеспечивающие функционирование системы. Здесь

обычно отдельно выделяют подсистему общего *управления данными*, а также процедуры *специализированной функциональной* обработки, отражающие требования предметной области.

Однако в наибольшей степени существо АИС выражается третьим компонентом — *информационным фондом*, который характеризуется не только содержащейся информацией, но и способом ее организации (*модель данных*), а также формой представления, которая, в свою очередь, определяется возможностями *лингвистического обеспечения* — языками представления и управления информацией. Именно лингвистическое обеспечение представляет существо (функциональные возможности и управляемость) АИС, обеспечивая «диффузный» слой между «естественной», обычно энергоматериальной средой ПрО, и информационной средой, имеющей преимущественно абстрактную искусственную природу.

Примерный организационно-функциональный состав АИС приведен на рис. 1.12.

Функциональные подсистемы реализуют и поддерживают модели, методы и алгоритмы обработки информации и формирования управляющих воздействий в рамках задач предметной области, т. е. состав и назначение функциональных подсистем зависит от предметной области особенностей использования ИС. На рис. 1.12 перечислены некоторые области, функциональность которых кажется достаточно очевидной. Отметим только, что подсистема *информационной поддержки* так или иначе есть в составе любой деятельности, так как именно она определяет качество выполнения научно-исследовательских (в том числе маркетинговых) работ, конструкторскую и технологическую подготовку производства.

Состав *обеспечивающих подсистем* достаточно стабилен и обычно мало зависит от предметной области использования ИС. Отметим следующие компоненты:

- *программное обеспечение* — совокупность программных компонент регулярного применения, необходимых для решения функциональных задач и программ, позволяющих наиболее эффективно использовать вычислительную технику, обеспечивая пользователям наибольшие удобства в работе;
- *математическое обеспечение* — совокупность методов, моделей и алгоритмов обработки информации, используемых в системе;
- *лингвистическое обеспечение* (ЛО) — это совокупность языковых средств, обеспечивающих гибкость и многоуровневость представления и обработки информации в АИС. Обычно ЛО вклю-

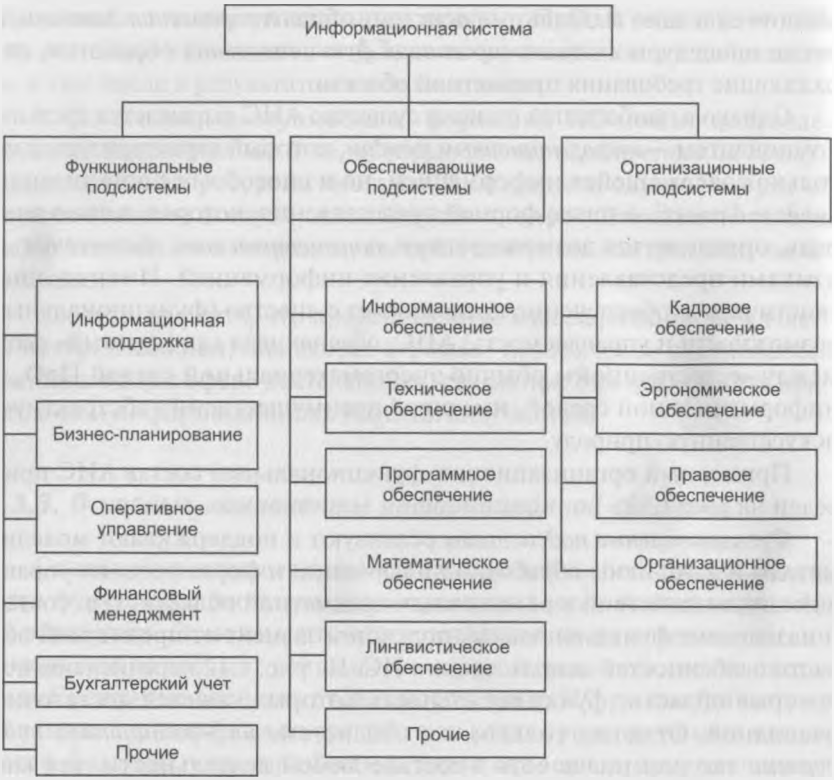


Рис. 1.12. Организационно-функциональный состав ИС

чает языки запросов и отчетов, специальные языки определения и управления данными, обеспечивающие адекватность внутреннего представления и согласование внутреннего и внешнего представлений. ЛО в наибольшей степени зависит от особенностей предметной области.

Организационные подсистемы также относятся к обеспечивающим, но направлены в первую очередь на обеспечение эффективной работы персонала и системы в целом, поэтому могут быть выделены отдельно. Отметим, что разработка ИС должна начинаться именно с организационного обеспечения: обоснования целесообразности системы, экономических показателей, определяющих ее деятельность, состава функциональных подсистем, организационной структуры управления, технологических схем преобразования информации, порядка проведения работ и т. д.

1.4. Информационная технология

1.4.1. Основные понятия

Для информационных технологий, в отличие от производственных, нацеленных в основном на создание новых продуктов, характерно относительное смещение функциональности. Здесь преобладают такие функции, как сбор, хранение, поиск, накопление, анализ, передача и распространение данных, информации и знаний. Информационная технология направлена на обработку и/или переработку «сырья» (в качестве которого выступают данные) путем использования соответствующих «машин», «механизмов» и «организационно-технологических приемов» (в качестве которых выступают аппаратные, программные, а также организационно-методические средства). Причем с точки зрения свойств, рассмотренных в первой главе, информация как объект имеет двойственную природу: она рассматривается и как целостный (неделимый) объект, имеющий *форму* существования, и как *содержание*, отражающее ее действенность.

Технология обработки информации зависит от характера решаемых задач, используемых средств вычислительной техники, числа пользователей, систем контроля процесса обработки информации и т. д. Технология, как процесс, всегда присутствует в любой предметной области, особенности которой, в свою очередь, оказывают существенное влияние на компоненты соответствующих технологий. Обработка происходит в процессе реализации *технологического процесса*, предопределяемого требованиями предметной области.

Для определения содержания и места информационных технологий рассмотрим следующие основополагающие понятия.

Методология[1] — это объединенная единым философским подходом совокупность методов, применяемых для получения проектного (целевого, заданного) результата.

Технология — это представленное в инструктивной форме выражение знаний и опыта, позволяющее *рационально* организовать получение проектного результата путем выполнения некоторого процесса

[1] Термины «методология» и «технология» используются здесь в узком *прикладном* смысле, а не в смысле науки, изучающей соответствующие объекты. Информационная *технология* в этом смысле изучает процессы создания, хранения и обработки информации, а также порядок их выполнения с использованием методов и средств информатики.

с использованием тех или иных средств, реализующих соответствующий метод.

Технологический процесс — направленная на создание заданного (проектного) объекта последовательность технологических операций (действий), согласованных в том числе с условиями выполнения и использующих соответствующие средства.

Технологическая операция представляет собой одно или несколько действий, направленных в рамках технологии на изменение состояния объекта или его взаимосвязи с окружением. Технологическая операция характеризуется наличием одного или нескольких входных объектов; выходного объекта — результата обработки; субъекта и средств управления обработкой. Практически любой конкретный технологический процесс можно рассматривать как часть более сложного процесса и в то же время как совокупность менее сложных (в пределе — элементарных) технологических процессов.

Элементарным технологическим процессом можно назвать такой, дальнейшая декомпозиция которого приводит к потере признаков, характерных для метода, положенного в основу данной технологии. В этом смысле технологическая операция может рассматриваться как элементарный технологический процесс.

В каждом из перечисленных определений явно или неявно присутствует понятие *метод*, имеющее общефилософское значение как путь исследования или преобразования действительности, основанный на знании закономерностей развития этой действительности. Метод предполагает наличие *средства* — того, *с помощью чего* осуществляется действие, реализующее метод, а также *способов* — то, *каким образом* осуществляется действие. Следует отметить, что методы и средства могут использоваться в разных процессах и, следовательно, разных технологиях.

1.4.2. Классификация информационных технологий

В рамках системного анализа сложные системы изучаются посредством разбиения на элементы: предполагается, что сложная система есть целое, состоящее из взаимосвязанных частей, которые не могут быть определены априорно, а строятся или выбираются в процессе декомпозиции (физической или концептуальной) исходной системы. Образующиеся в результате декомпозиции элементы обычно являются центрами некоторой активности (деятельности) и пото-

му называются *элементами деятельности*. При рассмотрении сложных систем наиболее часто выделяют *функциональные элементы/подсистемы* (однородные группы решаемых задач или технологических процессов) и *организационные* (обособленные, автономные и централизованно управляемые подсистемы сложной структуры).

Автоматизированные информационные технологии (АИТ) могут представлять собой как развитие неавтоматизированных (предметных) технологий, так и новые способы и процессы обработки информации. Большинство АИТ являются композициями четырех тесно взаимосвязанных и взаимозаменяемых компонент: интеллектуальных усилий и навыков пользователя; технических средств обработки данных; программного обеспечения; информационных ресурсов.

На рис. 1.13 представлена обобщенная схема абстрактного технологического процесса, рассматриваемого с «информационной» точки зрения.

Целевая обработка — это функционально-ориентированное преобразование входных или хранимых объектов обработки, обеспечивающее получение проектного результата под управлением субъекта (в качестве которого так или иначе выступает человек). Объектом и

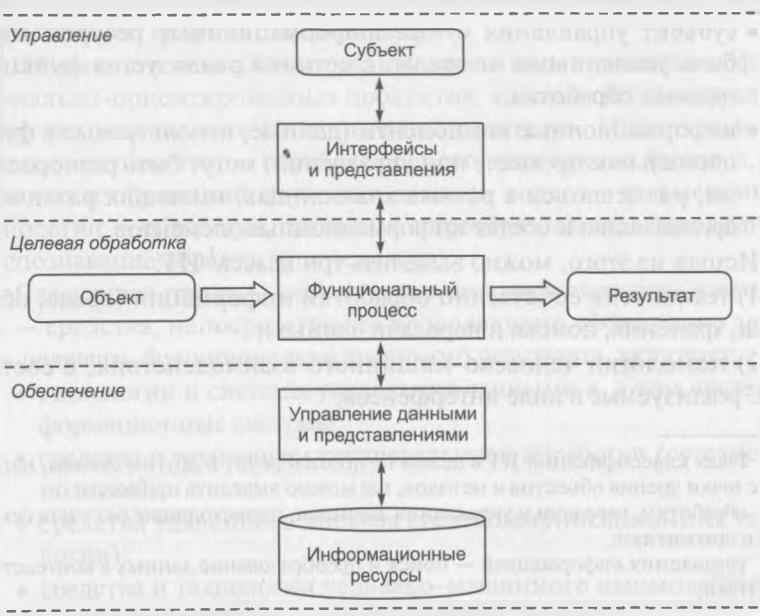


Рис. 1.13. Обобщенная схема абстрактного технологического процесса

результатом обработки может быть и информация: это соответствует понятию информационной деятельности. В этом случае ИТ, непосредственно реализующие уже какую-либо *целевую* функциональную технологию, представляют собой прикладные информационные технологии.

Информационные ресурсы — это внешние (по отношению к функциональному процессу) источники информации, использование которых обычно позволяет обеспечить эффективность целевой обработки.

Интерфейсные средства реализуют тот или иной способ (режим) взаимодействия субъекта с компонентами функциональной обработки.

Приведенная обобщенная схема включает основные «ролевые» компоненты технологического процесса, но не отражает следующие весьма существенные особенности их реализации:

- выполнение функциональной обработки может быть распределенным, например, распараллеленным;
- в технологиях высокой сложности человеку (как субъекту и потребителю результата) в силу, например, большой размерности данных необходимо предоставлять информацию в агрегированном виде, обеспечивающем удобство ее восприятия;
- субъект управления и/или информационные ресурсы могут быть удаленными от среды, в которой реализуется функциональная обработка;
- информационные компоненты (данные, используемые в функциональном процессе или управлении) могут быть разнородными, размещаться в разных хранилищах, имеющих различную организацию и состав информационных элементов.

Исходя из этого, можно выделить три класса[1] ИТ:

1) технологии собственно обработки информации (ввода, обработки, хранения, поиска и передачи данных);

2) технологии человеко-машинного взаимодействия, в составе АИС реализуемые в виде интерфейсов;

---

[1] Такая классификация ИТ в целом не противоречит и другим схемам, например, с точки зрения объектов и методов, где можно выделить процессы:

обработки, передачи и управления данными, происходящие без учета семантики и прагматики;

управления информацией — поиск и преобразование данных в контексте их семантики;

управления взаимодействием с человеком, реализующим человеко-машинный диалог.

3) инструментальные и другие вспомогательные технологии, позволяющие эффективно создавать и развивать ИТ предшествующих классов.

Отметим, что такое разделение, отражающее *специализированность* используемых методов и средств, соответствует и «специализации» пользователей соответствующих технологий, где давно сложилось разделение на «разработчиков», «конечных пользователей» и «администраторов». И с точки зрения этой «специализации» представляется целесообразным подразделять технологии на «базовые», «обеспечивающие» и «инструментальные».

Базовыми информационными технологиями (т. е. используемыми практически в любом процессе) являются те, которые в значительной степени определяются требованиями «архитектурного» уровня (в итоге — принципами фон Неймана). Обработка разнородной по форме информации, представляемой разнотипными данными, предопределяет соответствующий ряд средств и технологий, *ориентированных на форму* представления информации и виды операций, как, например: системы числовой обработки; системы и технологии обработки текстов (текстовые процессоры, системы распознавания текстов); средства обработки мультимедийной информации (например, растровой или векторной графики, звука, видео).

Обычно эти технологии реализуются в виде прикладных функционально-ориентированных продуктов, которые ассоциируются с понятием «технологии конечного пользователя». Можно выделить «чистые» технологии — обработка одного типа данных (текстов, статических изображений, звукового сигнала, видео) — и «смешанные» технологии — обработка, связанная с преобразованием типов данных (распознавание образов, чтение текста).

Следующая группа — *«обеспечивающие» информационные технологии* — средства, непосредственно позволяющие эффективно достигать целевого, функционально значимого результата, включает:

- технологии и системы управления данными и, в том числе, информационные системы;
- средства и технологии распределенной обработки (сетевые технологии);
- средства удаленного доступа (телекоммуникационные технологии);
- средства и технологии человеко-машинного взаимодействия и интерфейсы конечного пользователя;
- средства информационного поиска и управления знаниями.

Отметим, что перечисленные технологии являются, безусловно, важнейшими, но они относятся к «обеспечивающим», поскольку необходимость или необязательность их использования обусловлены характером задач пользователя или средой функционирования.

Третью группу составляют «инструментальные» технологии, обеспечивающие жизненный цикл самих ИТ, как, например:

- технологии проектирования и инструментальные средства разработки программного обеспечения;
- технологии проектирования баз данных;
- технологии реинжиниринга информационных систем.

Такая схема разделения ИТ на «базовые», «обеспечивающие» и «инструментальные» в целом не противоречит и другой классификации ИТ — с точки зрения объектов и методов. Здесь можно выделить следующие «страты»:

- процессов обработки, передачи и управления данными (ввод, хранение, поиск, манипулирование), происходящих в основном без учета семантики и прагматики;
- управления информацией — представление, извлечение, поиск, преобразование данных (ее представляющих) в контексте семантики и прагматики (в том числе для субъекта обработки — это получение, передача и использование знаний);
- управления взаимодействием с человеком (представление информации предметной области и результатов обработки, человеко-машинный диалог). Для случая инструментальных технологий (создания и использования целесообразных средств решения прикладных задач) — это методы и средства связывания технологий обработки данных и технологий обработки информации.

Контрольные вопросы

1. Дайте определение понятия «информация».
2. Охарактеризуйте соотношение понятий «информация», «данные», «знания».
3. Приведите примеры, которые отражают цикличность информационного обмена.
4. Определите атрибутивные свойства информации.
5. Охарактеризуйте прагматические свойства информации.

6. Опишите, как проявляется накопительный характер информации.
7. Назовите и охарактеризуйте формы концентрации информации.
8. Приведите примеры проявления свойства эмерджентности информации.
9. Приведите примеры проявления свойства старения информации.
10. Охарактеризуйте свойство рассеяния информации.
11. Дайте определение понятия «информационный объект». Приведите примеры и опишите свойства информационных объектов.
12. Опишите процесс взаимодействия объектов информационной среды и предметной области.
13. Дайте определение понятия «информация» на основе понятия «информационный объект».
14. Охарактеризуйте соотношение понятий «данные» и «информация» с точки зрения формы представления в информационной среде.
15. Можно ли трактовать термины «данные» и «информация» как синонимы?
16. Назовите и охарактеризуйте типы информационных объектов с точки зрения их «назначения».
17. Приведите примеры абстрактных систем.
18. Приведите примеры материальных систем.
19. Дайте определение понятия «информационная система».
20. Дайте определение информационной технологии.
21. Охарактеризуйте и классифицируйте информацию как основной объект обработки в ИС.
22. Приведите классификацию ИС.
23. Охарактеризуйте основные компоненты ИС.
24. Перечислите и охарактеризуйте основные обеспечивающие подсистемы ИС.
25. Определите понятие «информационная деятельность».

Глава 2
ТЕХНОЛОГИИ ОБРАБОТКИ ДОКУМЕНТОВ

Технологии работы с документами на компьютерах весьма распространены и часто отождествляются с информационными технологиями вообще. Более того, преподавание информационных технологий в учебных заведениях (средних, да и высших) зачастую исчерпывается обучением навыкам работы с текстовыми редакторами (наподобие MS Word) и табличными процессорами (MS Excell). Ни в коей мере не умаляя важности и необходимости владения данным инструментарием, авторы попытались основное внимание сосредоточить на вопросах структур документов, их представлениях и описаниях.

2.1. Основы представления документальной информации и технологий ее обработки

Человечеству свойственна потребность выражать, запоминать и передавать информацию о себе и об окружающем мире — так появились письменность, книгопечатание, живопись, фотография, радио, телевидение.

В истории развития цивилизации можно выделить несколько информационных революций — преобразование общественных отношений, так или иначе связанных с кардинальными изменениями в сфере обработки информации.

Первая информационная революция связана с появлением письменности, обеспечившей возможность сохранения знаний для их передачи другим субъектам, в том числе и последующим поколениям.

Вторая (конец XVI в.) вызвана изобретением книгопечатания, обеспечившим тиражирование знаний, которое радикальным образом изменило общество и культуру.

Третья (конец XIX в.) обусловлена изобретением электросвязи, благодаря чему появились телеграф, телефон, радио, обеспечившие не только массовость, но и предельную оперативность передачи информации.

Четвертая революция (70-е годы XX в.) связана с созданием персональных компьютеров, обеспечивших не только гибкие и управляемые коммуникационные функции, но и перенос части функций, в том числе и интеллектуальных, в вычислительную среду.

2.1.1. К истории документальных технологий[1]

Первая технология печати появилась в Древнем Китае к концу II в. К этому времени у китайцев уже были три необходимых элемента этой технологии: во-первых, бумага; во-вторых, краска; и, в-третьих, умение вырезать (или выгравировывать) тексты на различных поверхностях. Это, например, были буддийские изречения, вырезанные на мраморных колоннах буддийских храмов. Легенды гласят, что паломники смачивали выступающие части букв краской, а затем прикладывали к ним увлажненные листы бумаги.

Примерно в 1041—1048 гг. китайский алхимик Пи-Шен создал первый в истории сменный шрифт, сделав литеры из обожженной смеси глины и клея. Он набирал текст, помещая литеры вплотную одна к другой на металлическую пластину, покрытую смесью резины, воска и бумажного пепла. Пластина нагревалась, смесь расплавлялась и затем, остывая, прочно прикрепляла набор к пластине. Снять литеры было можно, снова нагрев пластину.

Таким образом впервые было найдено универсальное решение многих проблем: была разработана *технология* производства, набора и повторного использования шрифта. Примерно в 1313 г. чиновник по имени Ван-Чен приказал мастерам вырезать более чем 60 тысяч иероглифов на деревянных блоках для печати исторической монографии. Этому человеку также приписывают изобретение горизонтальных рамок-«касс», вращающихся вокруг вертикальной оси, что упрощало процесс набора.

Металлографическая печать считается прямой предшественницей полиграфии. Процесс изготовления печатных форм, скорее все-

[1] Приводится по [Зарождение печати http://www.gosreglament.ru/article/history.shtml].

го, состоял из трех этапов: 1) создавался набор литер — медных или бронзовых пресс-форм, на каждой из которых выгравировывалась определенная буква алфавита; 2) с помощью этих пресс-форм шрифт выдавливался на глиняной матрице; 3) в углубления глины заливался свинец, который, застывая, превращался в литеры.

Набранный шрифт (вставленный в металлическую рамку-форму) покрывался краской, сверху на него помещали лист бумаги, а затем все это вместе зажималось в тиски.

В стремлении повысить скорость и эффективность печатных процессов полиграфисты неизбежно сталкивались с необходимостью механизировать и даже автоматизировать набор. В 1822 г. Уильям Черч запатентовал наборную машину, представляющую из себя ячейки с литерами и клавиатуру. Нажатием клавиши соответствующая литера высвобождалась и опускалась в магазин. Выравнивание литер внутри магазина производилось вручную. С другой стороны, развитие телетайпного оборудования позволило к 1929 г. создать оборудование, полностью использующее принцип разделения функций человека и машины. Оператор изготавливал перфоленту, на которой каждый символ был представлен комбинацией отверстий, затем лента заряжалась в считывающее устройство, которое управляло отливом целых строк. В 1950-е годы во Франции была создана первая система программного набора. Оператор по-прежнему изготавливал перфоленту, но задачи определения длины строки, расстановки переносов, исправления орфографии и даже воспроизведения текста на основе шаблона верстки — все это брал на себя компьютер.

Основой любого из этих способов, а по существу — технологий сохранения и передачи информации, является принцип, предполагающий использование некоторого носителя и соответствующего этой среде (его свойствам) способ переноса и фиксирования информации на нем. Существенно, что природа и свойства такого носителя отличаются от свойственных источнику именно тем, что обеспечивают более эффективную (скорость, массовость и т. п.) передачу и/или сохранность (надежность, компактность, долговременность) информации. Причем перенос и фиксирование означает, что представление отдельного информационного элемента (например, знака) на исходном носителе должно быть сопоставлено представлению этого элемента на другом (например, в процессе кодирования). Эффективность определяет и способ фиксирования представления. Например, знак (букву, иероглиф) можно писать пером или кисточкой, а можно использовать штамп (в том числе и для целых текстов).

Сочетание этих составляющих (знак, носитель, способ фиксирования) и составляет технологию. Именно поэтому в следующих пунктах сравнительно большое внимание будет уделено способам представления информации. Будут рассмотрены как формы фиксирования элементов основных типов (символы, тексты, цифры, числа), так и языки разметки документов от команд управления размещением информации на экране или при печати до технологий XML.

2.1.2. Основы машинного представления информации

Представление аналоговой информации. Исторически первой машинной (электронной) формой передачи и хранения информации являлось аналоговое (непрерывное) представление звукового, оптического, электрического или другого сигнала (сообщения). Известнейшими примерами являются магнитная аудио- и видеозапись, фотографирование, запись на шеллачные или виниловые грампластинки, проводное и радиовещание — они представляют собой основные способы хранения и передачи информации в аналоговой форме (рис. 2.1).

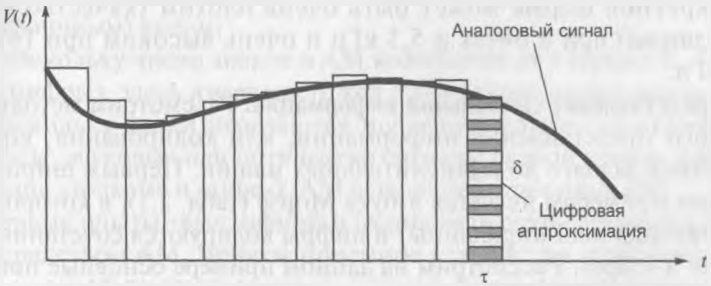


Рис. 2.1. Аналоговый сигнал и его дискретная (цифровая) аппроксимация (оцифровка):
τ — период дискретизации (sampling rate); δ — уровни квантования по амплитуде

Дискретное представление непрерывного сигнала осуществляется аналого-цифровым преобразованием (АЦП), которое заключается в формировании последовательностей n-разрядных двоичных слов, представляющих с заданной точностью аналоговые сигналы. Соответственно, осуществляется и обратное — дискретно-аналоговое (цифро-аналоговое преобразование — ЦАП, DAC).

Более чем тридцатилетнее развитие теории и практики ЭВМ приводит к вытеснению (в том числе и на бытовом уровне) аналоговых

устройств и сигналов цифровыми. Наиболее популярным примером является аудиокомпактдиск (digital audio CD).

В этом случае звуковой сигнал (см. рис. 2.1) сначала преобразуется в дискретную аппроксимацию («многоуровневый ступенчатый сигнал»), при этом происходят квантование по времени и измерение в дискретные моменты параметра аналогового сигнала.

При квантовании по амплитуде каждая ступенька представляется последовательностью бинарных цифровых сигналов. Принятый в настоящее время стандарт CD использует так называемый «16-разрядный звук с частотой сканирования 44 кГц». Для рис. 2.1 это означает, что «длина ступеньки» (τ) равна 1/44 100 с, а «высота ступеньки» (δ) составляет 1/65 536 от максимальной громкости сигнала (поскольку $2^{16} = 65\,536$). При этом частотный диапазон воспроизведения составляет 0—22 кГц, а динамический диапазон — 96 децибел (что составляет совершенно недостижимую для магнитной или механической звукозаписи характеристику качества).

Количество выборок в секунду, т. е. частота дискретизации аналогового звукового сигнала, также может принимать различные значения: 5,5, 11, 22 и 44 кГц. Таким образом, например, качество звука в дискретной форме может быть очень плохим (качество радиотрансляции) при 8 битах и 5,5 кГц и очень высоким при 16 битах и 44 кГц.

Представление символьной информации. Рассмотрим методы дискретного представления информации, или кодирования, которые появились задолго до вычислительных машин. Первым широко известным примером является азбука Морзе (табл. 2.1), в которой буквы латиницы (или кириллицы) и цифры кодируются сочетаниями из «точек» и «тире». Рассмотрим на данном примере основные понятия, связанные с кодированием.

Кодируемые (обозначаемые) элементы входного алфавита называются *символами*; кодирующие (обозначающие) элементы выходного алфавита — *знаками*; количество различных знаков в выходном алфавите назовем *значностью* (*-арностью*); количество знаков в кодирующей последовательности для одного символа — *разрядностью кода*.

Последовательным кодом является такой, в котором знаки следуют один за другим во времени (например, радио- или оптические сигналы, либо передача по двум проводам, двужильному кабелю), *параллельным* — тот, в котором знаки передаются одновременно (например, по четырем проводам), т. е. символ передается одномоментно, в один прием.

Таблица 2.1. **Фрагмент кода Морзе**

Символ входного алфавита	Мнемоническое обозначение по МСС	Кодовая (знаковая) комбинация
A	alfa	.-
B	bravo	-...
C	charlie	-.-.
D	delta	-..
E	echo	.
		...
Y	yankee	-.--
Z	zulu	--..
1	one	.----
		...
9	nine	----.

Применительно к азбуке Морзе (АМ):

- символами являются элементы алфавита (буквы A—Z или А—Я) и цифры (0—9);
- знаками — точка и тире (либо + и –. либо 1 и 0, а в общем случае — два любых, но разных знака);
- поскольку знаков два, АМ является *двузначным* (бинарным, двоичным) кодом;
- поскольку число знаков в АМ колеблется от 1 (буквы Е, Т) до 5 (цифры), здесь имеет место код с *переменной разрядностью*.

Поскольку знаки передаются последовательно (электрические импульсы, звуковые или оптические сигналы разной длины, соответствующие «точкам» и «тире»), АМ есть *последовательный код*.

Первые опыты телеграфной и радиосвязи осуществлялись именно посредством АМ, причем приемное устройство записывало импульсы разной длины («точки» и «тире») на движущуюся телеграфную ленту, однако уже в начале XX в. был осуществлен переход на 5-разрядный (5-битовый) телеграфный код.

Сигналы, реализующие коды, могут быть реализованы одним из следующих способов:

- униполярный код (значения сигнала равны 0, +1 либо 0, –1);
- полярный код (значения сигнала равны –1, +1);
- биполярный код (значения равны –1, 0, +1).

Именно биполярные коды часто используются в каналах передачи данных (рис. 2.2). Здесь единицы представляются чередующимися положительными и отрицательными импульсами. Отсутствие импульсов определяет состояние «нуль». Биполярное кодирование обес-

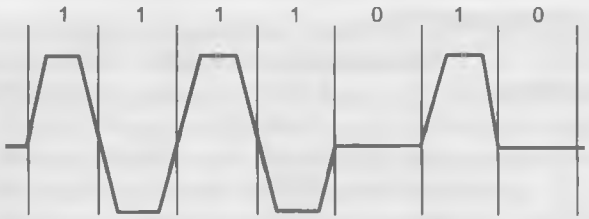


Рис. 2.2. Биполярное кодирование

печивает обнаружение одиночной ошибки. Так, если вместо нуля появится единица либо единица ошибочно сменится на нуль, то это легко обнаруживается. В обоих случаях нарушается чередование полярности импульсов.

Приведем перечень наиболее известных кодов, некоторые из них использовались первоначально для связи, а затем — для представления информации в ЭВМ:

- код Бодо — 5-разрядный код, бывший в прошлом европейским стандартом для телеграфной связи (другое название — IA-1 — international alphabet #1);
- М-2 (российское обозначение) или IA-2 (международное обозначение) — телеграфный код, предложенный Международным консультационным комитетом по телефонии и телеграфии (МККТТ) и заменивший код Бодо;
- ASCII (American Standard Code for Information Interchange) — стандартный 7-битовый код для передачи данных, поддерживает 128 символов, включающих заглавные и строчные символы латиницы, цифры, специальные значки и управляющие символы;
- ASCII-8 — 8-разрядный код, принятый для внутреннего и внешнего представления данных в вычислительных системах. Включает стандартную часть (128 символов) и национальную (128 символов). Соответственно, в зависимости от национальной части, кодовые таблицы различаются (например, CP-852 — стандарт IBM для греческого алфавита, CP-866 — стандарт IBM для русской кириллицы);
- EBCDIC (Expanded Binary Coded Decimal Information Code) — 8-разрядный код, предложенный фирмой IBM для машин серий IBM/360-375;
- код Холлерита, 12-разрядный код, предложенный в 1913 г. и затем использовавшийся для кодирования информации на перфокартах;

- UNICODE — стандарт кодирования символов, позволяющий представить знаки практически всех письменных языков, в том числе письменности как с направлением написания слева направо, так и с написанием справа налево, например, арабское письмо. Юникод имеет несколько форматов представления (*Unicode transformation format, UTF*): UTF-16 (16-разрядный), UTF-32 (32-разрядный) и UTF-8. Причем UTF-8 — это представление, обеспечивающее совместимость со старыми 8-разрядными системами кодирования: символы с кодом меньше 128 в UTF-8 соответствуют символам в ASCII. Остальные символы представляются последовательностями длиной от 2 до 6 байт (на деле только до 4 байт, поскольку в Юникоде нет символов с кодом больше 10FFFF, и вводить их не планируется), в которых первый байт всегда имеет вид 11xxxxxx, а остальные — 10xxxxxx.

В табл. 2.2 приводится пример кодирования нескольких символов в наиболее известных кодах.

Таблица 2.2. **Фрагменты некоторых кодовых таблиц**

Символ	IA-2	Бодо	ISO-7	EBCDIC	ASCII-8	Unicode
A	03	10	41	C1	A1	0041
B	19	06	42	C2	A2	0042
C	0E	16	43	C3	A3	0043
D	09	1E	44	C4	A4	0044
a			61	81	E1	0061
b			62	82	E2	0062
c			63	83	E3	0063
d			64	84	E4	0064
. (точка)	1C	05	2E	4B	4E	002E
, (запятая)	0C	09	2C	6B	4C	002C
:(двоеточие)	1E		3B	5E	5B	002A
? (вопрос)	10	0D	3F	6F	5F	003F

Системы счисления. Кроме кодирования символов в ЭВМ очевидное и важное значение имеет кодирование и представление чисел. Но если кодирование символов имеет целью, преимущественно, сохранение и передачу (не преобразование) этого символа, то кодирование цифр и чисел должно обеспечить еще и их обработку, например, выполнение арифметических операций. То есть представление в этом

случае будет определяться не только размерностью, но и *системой счисления*.

Мы привыкли считать предметы десятками, сотнями: десять единиц образуют десяток, десять десятков — сотню, десять сотен — тысячу и т. д. Это система счисления десятичная. Но десятичная система не единственно возможная. Существуют, например, двенадцатеричная система счисления (счет идет на дюжины) или римская система счисления.

Наиболее естественный способ представления числа в компьютерной системе заключается в использовании строки битов, называемой двоичным числом — числом в двоичной системе счисления (символ текста тоже может быть представлен строкой битов, называемой кодом символа).

Система счисления — способ именования и изображения чисел с помощью символов, имеющих определенные количественные значения. В зависимости от способа изображения чисел системы счисления делятся на:

- непозиционные;
- позиционные.

В *непозиционной* системе счисления цифры не меняют своего количественного значения при изменении их расположения в числе.

Приведем примеры непозиционных систем счисления.

1. Самый простой и очевидный пример — система счисления, где количество обозначается I (палочкой/единицей):

1 = I;
2 = I I;
5 = I I I I I;
10 = I I I I I I I I I I.

2. Пусть следующие символы (цифры в гипотетической системе счисления) соответствуют числам (десятичной системе счисления):

♊ — 1;
♎ — 6;
♒ —12;
♐ — 24;
♍ — 60;
♌ — 365,

и пусть есть правило, по которому любое число можно записать любой комбинацией таких символов, чтобы сумма обозначаемых ими чисел была равна заданному числу.

Тогда 444 можно записать как

𓏺𓎆𓍢𓂭𓐍 (365 + 60 + 12 + 6 + 1);

𓂭𓐍𓏺𓎆𓍢 (6 + 1 + 365 + 60 + 12),

т. е. 𓏺𓎆𓍢𓂭𓐍 = 𓂭𓐍𓏺𓎆𓍢

Такая система счисления является непозиционной, так как цифры не меняют своего количественного значения при изменении их расположения в числе.

В *позиционной* системе счисления количественное значение каждой цифры зависит от ее места (позиции) в числе.

Десятичная система счисления является позиционной, так как значение каждой цифры зависит от ее места (позиции) в числе.

Например:

$23 = 2 \times 10 + 3;$

$32 = 3 \times 10 + 2$

и $23 \neq 32$.

Римская система счисления является смешанной, так как значение каждой цифры частично зависит от ее места (позиции) в числе. Так в числах: **VII, VI, IV — V** обозначает 5, а **I** обозначает 1. Но, с другой стороны, важно, как цифры расположены относительно друг друга:

VII = 5 + 1 + 1 = 7;

VI = 5 + 1 = 6;

IV = 5 – 1 = 4.

Основание системы счисления — количество (P) различных цифр, используемых для изображения числа в позиционной системе счисления. Значения цифр лежат в пределах от 0 до $P - 1$.

В общем случае запись любого числа N в системе счисления с основанием P будет представлять собой ряд (многочлен) вида:

$$N = a_{m-1} \times P^{m-1} + a_{m-2} \times P^{m-2} + ... + a_k \times P^k + ...$$

$$... + a_1 \times P^1 + a_0 \times P^0 + ... + a_{-1} \times P^{-1} + a_{-2} \times P^{-2} + ... + a_{-s} \times P^s. \quad (1)$$

Нижние индексы определяют местоположение цифры в числе (разряд):

- положительные значения индексов — для целой части числа (m разрядов);
- отрицательные значения — для дробной (s разрядов).

Максимальное целое число, которое может быть представлено в m разрядах:

$$N_{max} = P^m - 1$$

Минимальное значащее, не равное 0 число, которое можно записать в s разрядах дробной части:

$$N_{min} = P^{-s}.$$

Имея в целой части числа m, а в дробной — s разрядов, можно записать P^{m+s} разных чисел.

Двоичная система счисления имеет основание $P = 2$ и использует для представления информации две цифры: 0 и 1.

Существуют правила перевода чисел из одной системы счисления в другую, основанные, в том числе, и на выражении (1).

Например, двоичное число 101110,101 равно десятичному числу 46,625:

$$101110{,}101_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 +$$

$$+ 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 46{,}625_{10}.$$

Кроме двоичной и десятичной при работе с компьютером часто используются также *двоично-десятичная* и *шестнадцатеричная* системы счисления (табл. 2.3).

Шестнадцатеричная система счисления часто используется при программировании. Перевод чисел из шестнадцатеричной системы счисления в двоичную систему счисления весьма прост — он выполняется поразрядно.

Для изображения цифр, больших 9, в шестнадцатеричной системе счисления применяются буквы A = 10, B = 11, C = 12, D = 13, E = 14, F = 15. Например, шестнадцатеричное число F17B в двоичной системе выглядит так: 1111000101111011, а в десятичной — 61819.

Двоично-десятичная система счисления используется там, где основное внимание уделяется не простоте технического построения машины, а удобству работы пользователя. В этой системе счисления все десятичные цифры отдельно кодируются четырьмя двоичными цифрами и в таком виде записываются последовательно друг за другом. Двоично-десятичная система не экономична с точки зрения реализации технического построения машины (примерно на 20 % увеличивается требуемое оборудование), но очень удобна при подготовке задач и при программировании. В двоично-десятичной системе счисления основанием системы счисления является число 10, но каждая десятичная цифра (0, 1, ..., 9) изображается при помощи двоичных цифр, т. е. кодируется двоичными цифрами.

Таблица 2.3. **Перевод цифр в разные системы счисления**

Триада	Восьмеричная цифра	Тетрада	Шестнадцатеричная цифра	Десятичное число	Двоично-десятичная запись
000	0	0000	0	0	0000-0000
001	1	0001	1	1	0000-0001
010	2	0010	2	2	0000-0010
011	3	0011	3	3	0000-0011
100	4	0100	4	3	0000-0100
101	5	0101	5	5	0000-0101
110	6	0110	6	6	0000-0110
111	7	0111	7	7	0000-0111
		1000	8	8	0000-1000
		1001	9	9	0000-1001
		1010	A	10	0001-0000
		1011	B	11	0001-0001
		1100	C	12	0001-0010
		1101	D	13	0001-0011
		1110	E	14	0001-0100
		1111	F	15	0001-0101

Представление чисел в ЭВМ. В современных ЭВМ, как известно, применяется двоичная система счисления, которая обладает рядом значительных для машинной обработки свойств (простота выполнения арифметических действий, применение двузначной логики и т. д.), что делает целесообразным ее применение в этой области.

В вычислительных машинах применяются две формы представления чисел:

- естественная форма, или форма с фиксированной запятой (точкой) — ФЗ (ФТ);
- нормальная форма, или форма с плавающей запятой (точкой) — ПЗ (ПТ).

В форме представления с *фиксированной запятой* все числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть от дробной. Эта форма наиболее проста, естественна, но имеет небольшой диапазон представления чисел и поэтому чаще всего неприемлема при вычислениях.

В форме представления с *плавающей запятой* каждое число изображается в виде двух групп цифр: мантисса и порядок. При этом абсолютная величина мантиссы должна быть меньше 1, а порядок должен быть целым числом.

Форматы представления числовой информации в ЭВМ. В памяти ЭВМ числа с фиксированной точкой хранятся в трех форматах:

а) полуслово — это 16 бит, или 2 байта;

б) слово — это 32 бита, или 4 байта;

в) двойное слово — это 64 бита, или 8 байтов.

Отрицательные числа с ФТ записываются в разрядную сетку в дополнительных кодах, которые образуются прибавлением единицы к младшему разряду обратного кода. Обратный код получается заменой единиц на нули, а нулей — на единицы в прямом двоичном коде.

Типы данных. Структура информационных единиц, обрабатываемых на ЭВМ, включает следующие аспекты:

- *типы данных,* или совокупность соглашений о программно-аппаратурной форме представления и обработки, а также ввода, контроля и вывода элементарных данных;
- *структуры данных* — способы композиции простых данных в агрегаты и операции над ними;
- *форматы файлов* — представление информации на уровне взаимодействия операционной системы с прикладными программами.

Ранние языки программирования (ЯП), а точнее — системы программирования (СП) — Фортран, Алгол, будучи ориентированы исключительно на вычисления, не содержали развитых систем типов и структур данных.

В ЯП Алгол символьные величины и переменные вообще не предусматривались, в некоторых реализациях строки (символы в апострофах) могли встречаться только в операторах печати данных.

Типы числовых данных Алгола: INTEGER (целое число), REAL (действительное) — различаются диапазонами изменения, внутренними представлениями и применяемыми командами процессора ЭВМ (соответственно арифметика с фиксированной и плавающей

точкой). Нечисловые данные представлены типом BOOLEAN — логические, имеющие диапазон значений {TRUE, FALSE}.

Позже появившиеся ЯП COBOL, PL/1, Pascal вводят новые типы данных:

- символьные (цифры, буквы, знаки препинания и пр.);
- числовые символьные для вывода,
- числовые двоичные для вычислений.

Разновидности числовых данных здесь соответствуют внутреннему представлению и машинным (или эмулируемым) командам обработки. Кроме того, вводятся числа двойной длины (2 машинных слова), для обработки которых также необходимо наличие в процессоре (или эмуляция) команд обработки чисел двойной точности.

Появление систем управления базами данных и систем программирования для разработки ИС приводит к появлению новых типов данных:

- *дата и время*;
- *бинарные* (BLOB — Bynary Large Object) и *текстовые объекты*.

Понятие типа данных ассоциируется также с допустимыми значениями переменной и операциями над ними, например, данные типа время (ЧЧ:ММ:СС) или дата (ГГ/ММ/ДД) предполагают определенные диапазоны значений каждого из разрядов, а также машинные или эмулируемые операции (сложение/вычитание дат и/или моментов времени). Основной причиной «проблемы 2000 года» являлось не столько двухразрядная запись года в базах данных, сколько встроенные в огромное количество программ (часто не документированных) операции над данными типа DATE — ГГ/ММ/ДД.

2.1.3. Модель документа

Понятие *модель документа* охватывает аспекты создания, преобразования, хранения, поиска, передачи и отображения документов. Принято рассматривать структуру документа в двух аспектах: логическом (содержание) и физическом (макет).

Логическая структура определяет составные компоненты и их соотношения в понятиях, отвечающих взгляду на документы как смысловые структуры. Например, к основным смысловым компонентам относятся: авторские данные (имя автора, место работы), аннотация, оглавление, главы, разделы, параграфы, рисунки, сноски. На рис. 2.3 приведен пример документа «Пояснительная записка к ди-

пломному проекту». Здесь выделены такие базовые понятия структуры документа, как обязательность/необязательность элемента, уникальность или повторяемость, вхождение нижестоящих элементов в вышестоящие по принципу «И» (оба типа данных должны или могут входить в элемент) либо «ИЛИ» (только какой-либо один из типов данных может или должен входить в элемент).

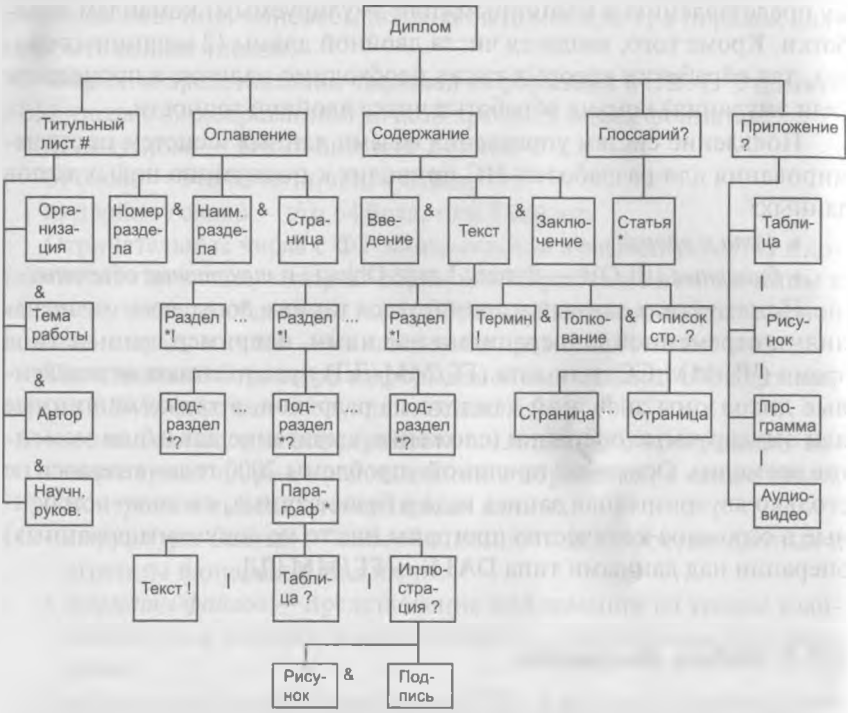


Рис. 2.3. Иерархическая структура документа «Пояснительная записка к дипломной работе»[1]

Макетная структура содержит описание документа в терминах физических единиц — страниц, полос, колонок, колонтитулов, рамок для рисунков, шрифтов, стилей и пр.

---

[1] # — уникальный элемент;* — повторяющийся элемент; ? — необязательный элемент; ! — обязательный элемент; & — вхождение типа «И», | — вхождение типа «ИЛИ».

Подходы к моделированию документов опираются на два стандарта — ISO 8613 (ODA — Office Document Architecture — архитектура управленческой документации) и ISO 8879 (SGML — Standard Generalized Markup Language — стандартный обобщенный язык разметки).

Документ в ODA представлен в виде профиля и собственно документа, организованных в форме древовидной структуры. Профиль содержит информацию о документе в целом и его прохождении; формальные признаки — дата составления, вид, регистрационный номер и т. д.

Собственно документ содержит текст и сведения о его структуре и стиле, а именно:

- структуру документа — заглавие, параграфы, оглавление и т. п. (логическая структура), а также абзацы, расположение текста, шрифты (физическая структура);
- архитектуру содержания — набор графических элементов, выделение определенных слов, строк и т. п.;
- коммуникативный формат — способы кодирования объектов, признаков и содержания документов.

2.1.4. Коммуникативные форматы

В 1965 — 1966 гг. в Библиотеке Конгресса был разработан проект, направленный на исследование возможности получения библиографического описания в машиночитаемой форме. Этот проект положил начало созданию семейства форматов MARC (Machine-Readable Catalogue), ориентированных на обмен всеми видами документов и решение разнообразных информационно-библиотечных задач, включая каталогизацию и применение в различных автоматизированных системах. Стандарты семейства коммуникативных форматов для обмена библиографическими и другими данными на машиночитаемых носителях определяют структуру записи и ее наполнение. При этом разные национальные и фирменные стандарты имеют практически одинаковую структуру (соответствующую требованиям ISO-2709) и достаточно разнообразные требования к наполнению, где основные различия касаются набора элементов данных и их представлений.

Отдельная запись в коммуникативном формате — это совокупность полей, описывающая одну или несколько информационных единиц (документов), рассматриваемых как единое целое. Но при

этом каждый экземпляр записи в одном файле может иметь свой состав полей переменной длины и свой вариант представления данных. Это достигается тем, что в структуре записи имеется три блока: заголовок, справочник, область элементов данных. Размещение полей данных определяется справочником переменной длины. В свою очередь структура справочника и состав идентифицирующих элементы данных компонентов определяется заголовком.

В частности, заголовок содержит *длину записи, базовый адрес области элементов данных* (равный общей длине маркера записи и справочника), *план справочника* (длина компонента «длина поля данных», длина компонента «позиция начального символа»), *идентификатор системы кодирования данных, длины индикатора*[1] *и идентификатора*, используемых для идентификации элементов данных данной записи.

Справочник состоит из переменного числа статей, каждая из которых идентифицирует соответствующее поле данных. Все статьи справочника имеют одинаковую структуру, задаваемую в плане справочника. Первые 3 позиции статьи справочника всегда занимает трехзначная *метка поля*.

Поля данных переменной длины следуют за справочником и содержат библиографические данные. Поле может состоять из одного или более элементов данных или подполей. Поле помимо метки идентифицируется индикатором, а подполе — идентификатором.

Такая структура формата может быть охарактеризована как «самоопределенная».

2.2. Языки разметки документов

В системах обработки текстов в документ включается дополнительная информация, называемая разметкой и выполняющая следующие функции:

- выделение логических элементов данного документа;
- задание функций обработки выделенных элементов (фрагментов).

[1] Индикатор поля — это элемент данных, расположенный (если длина его в маркере записи отлична от нуля) в начале поля и несущий дополнительную информацию о содержании поля, взаимосвязи между этим полем и другими полями в записи или об операциях, требуемых при определенной обработке данных.

В простейших текстовых процессорах существуют встроенные команды управления представлением (включения/выключения шрифтов и др.), аналогичные командам управления размещением информации на экране или при печати (так называемые *Escape-последовательности*[1]). Такой подход называется командной или процедурной разметкой (табл. 2.4).

Таблица 2.4. **Некоторые примеры команд разметки текстовых файлов**

Вид разметки	Принтер EPSON	Редактор Лексикон	Формат rtf (Rich Text Format), в т. ч. Word	Формат электронной почты (стандарт MIME)	html (Hypertext Markup Language)
Полужирный шрифт	ESC G... ESC H	chr(255)2 chr(255)0	{\b...}	<bold>...</bold>	...
Курсив (италик)	ESC 4... ESC 5	chr(255)1... chr(255)0	{\i ...}	<italic>...<italic>	...
Подчеркивание	ESC - 1... ESC - 0	_chr(255)... chr(0)	{ul\...}	<underline> </underline>	<u>...</u>
Индекс верхний	ESC S 0... ESC T	chr(255)5... chr(255)0	{\super...}	<superscript> </superscript>	
Индекс нижний	ESC S 1... ESC T	chr(255)4... chr(255)0	{\sub...}	<subscript> </subscript>	_{...}
Перевод страницы	chr(12)	.chr(12)	\page	пр	
Выравнивание	—	<Shift><F8>	\qc – по центру \ql — влево \qr — вправо	<Center> <FlushLeft> <FlushRight>	<aligh =center> =left> =right> =justifv>
Абзац	Табуляция (TAB, chr(9))	TAB	\par	Paragraph	<p>

Альтернативный способ разметки заключается в выделении фрагмента текста без указания способа обработки выделения. Затем другие команды назначают фрагментам способ обработки. Такая разметка называется *описательной* (дескриптивной). Она использует метки (*tags, теги*) начала и окончания элемента текста и указывает, как интерпретировать данный фрагмент.

[1] Escape-последовательность начинается с выделенного кода ESC, за которым следует обычно один байт из символьной части раскладки, значение которого интерпретируется как команда устройству отображения.

Изменяя набор процедур, соответствующий описательной разметке, можно изменить внешнее представление одного и того же документа. Развитие идей описательной разметки привело к определению разметки как формального языка.

2.2.1. Язык SGML

SGML разработан на базе программного продукта DCF GML фирмы IBM и представляет собой метод создания структурированных документов, а также языков для их разметки.

В языке SGML каждый документ имеет три части:

- декларации (объявления, определения) языка SGML, привязывающие к определенным значениям параметры обработки, а также имена синтаксиса;
- пролог, состоящий из деклараций о типе документа. Они определяют типы элементов, взаимосвязи между элементами и их атрибуты, а также условные обозначения, которые могут быть задействованы при разметке;
- данные, которые состоят из разметки документа и собственно информации.

Основные типы конструкций языка — описания:

- элементов `<!ELEMENT...>`;
- объектов `<!ENTITY...>`;
- атрибутов `<!ATTRIBUTE LIST...>`,

образующих структуру документа (документов), при этом элемент — основной компонент документа; объект — группа, род элементов; атрибут — характеристика элемента. Все прямоугольники с текстом, приведенные на рис. 2.3, являются элементами. Запишем одну из возможных конструкций, соответствующую выделенной на рис. 2.3 цепочке элементов (подраздел — параграф — текст...):

`<!ELEMENT SUBDIV (PAR*)>`	— подраздел состоит из параграфов (повторяющихся);
`<!ELEMENT PAR (TEXT | TABLE? | PICT?)>`	— параграф — из текста или таблицы/рисунка (не обязательны);
`<!ELEMENT PICT(IMAGE & CAPT)>`	— рисунок — из изображения и подписи.

Декларации и пролог на языке SGML задают структуру документа и, будучи отделены от размеченного текста, образуют описание типа документа (*DTD — Document Type Definition*).

2.2.2. HTML — язык разметки гипертекста

HTML[1] (Hypertext Markup language) в своей основе ориентирован на форматирование, в том числе выделение «активных» элементов текста (гипертекстовых ссылок), в которых участвуют его различные конструкции и элементы:

- описание структуры документа (HEAD, BODY, H1—H6, шрифты, списки и пр.);
- адресация ресурсов (BASE, LINK, HREF и пр.);
- создание гипертекстовых ссылок и управление навигацией в Internet (HREF и т. п.);
- реализация интерактивных интерфейсов с пользователем (ISINDEX, MENU, FORM и пр.).

HTML-документ включает два компонента:

- HEAD — содержит скрипты, инструкции по оформлению документа и другую метаинформацию о данном документе;
- BODY — содержит элементы, отвечающие за отображение документа и управление им, а также гипертекстовые ссылки.

Приведем некоторые элементы HTML, относящиеся к представлению документа.

1. Заголовки разделов документа.

H1 — жирный, очень крупный шрифт, текст центрирован. Между заголовком и последующим текстом вставляется одна или две пустые строки. При выводе на принтер заголовок печатается на новой странице;

H2 — жирный крупный шрифт, до и после заголовка помещаются одна или две пустые строки;

H3 — наклонный крупный шрифт, до и после заголовка помещаются одна или две пустые строки. Печатается с небольшим отступом;

H4 — жирный нормальный шрифт, до и после заголовка помещается пустая строка;

H5 — наклонный шрифт, как и для H4, пустая строка ставится перед заголовком;

H6 — жирный шрифт, перед заголовком ставится пустая строка.

2. Физические (макетные) стили.

TT — (телетайп) шрифт фиксированной ширины;

B — жирный или еще каким-либо образом выделенный шрифт;

---

[1] Собственно HTML определяется в терминах SGML. Например, HTML-документ как целое в DTD задается декларацией: <!ELEMENT HTML ((HEAD | BODY | %oldstyle)*, PLAINTEXT?)>.

I — наклонный шрифт (или видоизмененный еще каким-либо образом);

U — подчеркивание.

3. Логические стили.

STRONG — более четкое выделение, привлечение внимания (обычно применение более жирного шрифта);

CODE — фрагмент кода (формулы, выражения);

VAR — имя переменной (имена переменных в примерах, формулах);

DFN — определение к какому-либо термину — обычно жирный наклонный;

CITE — цитата, обычно наклонный шрифт (названия документов, выдержки из документов, цитируемые фразы и т. д.)

Рассмотрим пример документа с разметкой HTML, содержащий приведенные выше элементы управления стилем символов текста:

```
<HTML><TITLE> Примеры управления шрифтами </TITLE>
<H1> Заголовок 1 </H1>
<H2> Заголовок 2 </H2>
<b> Это текст Bold </b><p>
<i> Это текст Italic</i><p>
<u> Это подчеркнутый текст </u> <p>
<strike> Это перечеркнутый текст </strike>
<p>В обычный текст можно вставить <sub> подстрочный </sub>
текст, что позволяет написать выражение типа
P<sub>max</sub>=max{P<sub>1</sub>,P<sub>2</sub>}
<p>В обычный текст можно вставить <sup> надстрочный<,'sup>
текст, что позволяет написать обозначение изотопа в виде
Cs<sup>134</sup>
</HTML>
```

Пример отображения этого текста браузером MS Explorer представлен на рис 2.4.

Списки. В HTML предусмотрены следующие виды списков:

- UL — ненумерованный список (неупорядоченный);
- OL — нумерованный список (упорядоченный);
- DL — список определений.

Типичный неупорядоченный список:

```
<UL>
<LH>Title of WWW programmes (NCSA).
<LI> NCSA HTTPD;
<LI> NCSA MOSAIC
<UL>
<LH>Title of WWW programmes (CERN).
```

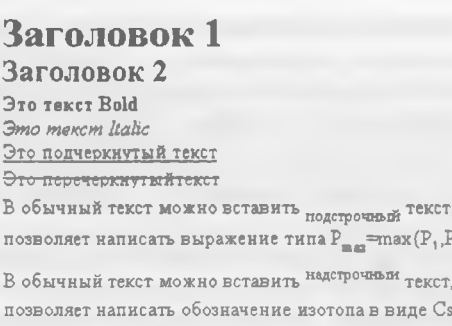


Рис. 2.4. Управление отображением стиля символов текста

```
<LI> CERN HTTPD;
<LI> AGORA - email robot;
<LI> HTTPD CERN;
<LI> WWW Line Browser;
<LI> Arena.
</UL>
<UL>
<LH> Title of WWW programmes (Netscape).
<LI> Netsite - server;
<LI> Netscape Navigator.
</UL>
```

Пример интерпретации данного списка приведен на рис. 2.5.

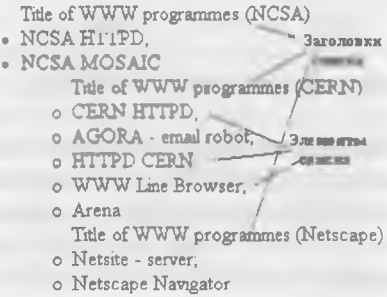


Рис. 2.5. Отображение ненумерованного списка

Таблицы. Для описания таблиц служит элемент TABLE, который является контейнером для других элементов описания таблицы. Наиболее часто он употребляется с атрибутом BORDER, определяющим разделительные линии граф таблицы, которые могут быть либо трехмерными, либо обычными.

Элемент TR (Table Row) задает описание строки таблицы. Обычно используется для выравнивания и оформления содержания граф строки. Способ выравнивания определяют: атрибут ALIGN — горизонтальное выравнивание, который принимает значения left, right, center, justify, decimal, и атрибут VALIGN — вертикальное выравнивание, который принимает значения top, bottom, middle, baseline.

Элементы TH (Table Header) и TD (Table Data) используются для описания граф таблицы. Кроме выравнивания TH и TD позволяют еще и объединять графы. Это делается с помощью атрибутов ROWSPAN (пропуск строки) и COLSPAN (пропуск столбца). Цифра в этих атрибутах определяет количество последовательно расположенных граф таблицы, объединенных в одну графу.

Математика. Для реализации математических выражений в языке определен элемент MATH, внутри которого содержатся следующие компоненты:

- ABOVE (запись символа над выражением);
- BELOW (запись символа под выражением)
- SQRT, ROOT (радикалы);
- TEXT (для записи текста);
- B, T, BT (выделение символов);
- OVER (черта) и пр.

Например, запись <ROOT>3<OF>1+x</ROOT> соответствует $\sqrt[3]{1+x}$.

2.2.3. Язык описания документов PostScript

Фактически PostScript представляет собой язык программирования, предназначенный для описания разного рода графических объектов и последующей печати созданных иллюстраций, верстки, простых документов точно в таком виде, как они видны на экране.

Концепция языка PostScript была создана в 1976 г., когда разрабатывался интерпретатор для большой трехмерной графической базы данных по Нью-Йоркской гавани.

Напомним, что ранние принтеры были устроены так, чтобы печатать символы текста, обычно поступающего на вход в коде ASCII. Было множество технологий для этой цели, но наиболее распространенным было то, что печатаемые символы были «намертво» проштампованы на клавиши пишущей машинки, отлиты в металле для линотипов или нане-

сены на негативы фотонаборных устройств, и поэтому их было физически трудно изменить.

Это изменилось до некоторой степени с распространением матричных печатающих устройств. Символы на этих системах могли быть «нарисованы» как совокупность точек, соответствующих определенным таблицам шрифтов в принтере. По мере усовершенствования матричные печатающие устройства стали включать несколько встроенных шрифтов, из которых пользователь мог выбирать, а некоторые модели давали пользователям возможность передать (загрузить) их собственные заказные шрифты в принтер.

Матричные печатающие устройства также дали возможность печатать растровую графику. Графические символы интерпретировались компьютером и посылались как ряд точек на принтер, используя «escape-последовательности». Эти языки управления менялись от принтера к принтеру, требуя разработки многочисленных драйверов.

Векторный вывод графических символов возлагался на другие устройства — плоттеры (графопостроители). Они также могли использовать общий командный язык — HPGL, но имели ограниченное использование для чего-нибудь другого, кроме вывода векторной графики. Кроме того, они были дорогими и медленными и, таким образом, не имели широкого распространения.

PostScript позволил комбинировать лучшие особенности как принтеров, так и плоттеров: высококачественную штриховую графику и единый язык управления, который мог использоваться на принтерах любых марок.

PostScript осуществляет растеризацию образа в процессе обработки данных («на лету»), поскольку все, даже текст, определено в терминах прямых линий и кубических кривых Безье, что позволяет осуществлять произвольное масштабирование, вращение изображения и другие преобразования. В процессе работы интерпретатор программы PostScript преобразует эти команды в точки изображения, формируя вывод. Поэтому интерпретаторы PostScript также иногда называют процессорами растровых изображений (PostScript Raster Image Processors, или RIP).

В процессе конвертации в PostScript программа, выполняющая печать, передает готовые данные программе-спуллеру, поставляемой вместе с операционной системой в виде ее расширения. Спуллер является не более чем накопителем данных, после того как печатающая программа закончила конвертацию и передачу спуллеру информации, укомплектованный временный файл печати (spool file) передается на драйвер принтера, который либо выводит его на печать, либо, по желанию пользователя, формирует принтерный файл (PostScript-файл).

Так как документ-программа не требует изменений в зависимости от адресата, он называется *независимым от устройства* (device-independent).

Такой файл, как правило, содержит следующие составляющие:

- исходный документ, описанный средствами PostScript;
- использованные в нем внедренные или импортированные по технологии OPI-файлы;
- файлы шрифтов;
- программу для принтера на языке PostScript.

С развитием компьютеров и принтеров, а также благодаря увеличению пропускной способности интерфейсов, шрифты стали загружать в большинстве случаев в файл, а не в принтер, что вызывает некоторое увеличение PS-файла, зато гарантирует идентичность отображения.

PostScript является полным языком программирования (в смысле Тьюринга). Как правило, PostScript-программы создаются не программистами, а другими программами. Конечно, есть возможность создать графические образы или выполнить какие-либо вычисления, кодируя вручную на ЯП PostScript. PostScript — интерпретируемый язык на основе стека, подобный Forth, использующий структуры данных, аналогичные встречающимся в Лиспе (Lisp) и пр. Большинство операторов (в других языках используется термин «функция») принимает значения параметров из стека и помещает результат выполнения в стек.

Синтаксис языка опирается на обратную польскую запись (Reverse Polish Notation — RPN), которая делает круглые скобки ненужными, но при которой чтение программы требует некоторых навыков, поскольку требуется помнить содержание стека. Рассмотрим ряд примеров.

С помощью оператора

```
3 4 add 5 1 sub mul
```

будет вычислено: (3 + 4) x (5 – 1).

Чтобы производить графические образы, PostScript использует обычную декартову систему координат, например, оператор

```
100 200 moveto 300 400 lineto stroke
```

перемещает «курсор» в точку с координатами (100, 200), а затем чертит прямую линию к точке (300, 400).

Оператор

```
50 70 moveto 100 200 50 80 100 100 curveto stroke
```

создает кубическую кривую Безье между точками (50, 70) и (100, 100), проходящую через контрольные точки (100, 200) и (50, 80).

Оператор

```
250 250 moveto (Programming Languages) show
```

поместит начало текста «Programming Languages» в точку с координатами (250, 250).

Шрифт, которым будет набран текст, может быть предварительно задан (например, командной строкой/Courier findfont 12 scalefont setfont).

2.2.4. Переносимый формат документов (PDF)

Переносимый формат документов — Portable Document Format — формат файла, созданный Adobe Systems в 1993 г. для использования в настольных издательских системах. Формат PDF позволяет представлять двумерные документы в форме, независимой от разрешающей способности устройств печати (или дисплеев). Каждый файл формата PDF содержит полное описание двумерного документа (с появлением Acrobat 3D — трехмерных документов), который включает текст, шрифты, изображения и двумерную векторную графику, которые образуют документ.

Формат PDF использует следующие технологии:

- подмножество языка программирования и описания страниц PostScript, чтобы генерировать размещение и графику;
- систему встраивания и замены шрифтов для обеспечения переносимости документов;
- структурированную систему хранения, позволяющую связывать эти элементы в отдельный файл, в том числе с использованием сжатия данных.

2.3. Технологии XML

Проблема интеграции информации из различных источников (в том числе и из Internet) состоит прежде всего в том, что во многих случаях данные слабо структурированы: нет схемы, которая была бы

задана заранее, а данные из разных источников могут иметь как различный набор атрибутов, так и различные типы. Более того, каждый из источников может иметь свою степень автономности и характеризоваться своими метаданными.

XML-технологии — современное развитие инструментария HTML и SGML. В начале февраля 1998 г. международная ассоциация W3C утвердила спецификацию «Extensible Markup Language(XML) 1.0». Сегодня появляются новые языки, созданные на основе XML. Возникают многочисленные Web-серверы, использующие и технологию XML для организации хранения информации.

2.3.1. Синтаксис XML

Так же как и в HTML, инструкции, заключенные в угловые скобки, называются тэгами и служат для разметки основного текста документа. В XML существуют открывающие, закрывающие и пустые тэги (в HTML понятие пустого тэга тоже существует, но специального его обозначения не требуется).

XML-документы имеют два раздела: *пролог* и *тело* документа.

Пролог, предваряющий любой XML-документ, включает *объявление XML* и *объявление типа документа*. Объявление XML заключается между парами символов <? и ?> и может включать указание программе-анализатору текущего стандарта, объявление кодировки и самостоятельности, а объявлению типа документа предшествует ключевое слово DOCTYPE, например, объявление XML-документа с внешним DTD-определением в файле `sample.dtd`:

```
<?xml version="1 1" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE sampledoc SYSTEM "sample.dtd">
```

Тело XML-документа состоит из элементов разметки и непосредственно содержимого документа — данных и представляет собой набор элементов и атрибутов, секций CDATA, директив анализатора, комментариев, спецсимволов, текстовых данных. К телу документа относится все, кроме пролога.

Помимо элементов и атрибутов, обеспечивающих структурированное представление текстовой информации, языки разметки могут оперировать и двоичными данными (специальные символы, графика и т. д.) — *сущностями*, которые представляют собой поименованные фрагменты данных.

Элементы данных. Элемент — это структурная единица XML-документа. В общем случае в качестве содержимого элементов могут выступать как просто какой-то текст, так и другие, вложенные элементы документа, секции CDATA, инструкции по обработке, комментарии, т. е. практически любые части XML-документа.

Любой непустой элемент должен состоять из начального, конечного тэгов и данных, между ними заключенных, например:

```
<flower> rose </flower>
<root>
   <child>
      <subchild>.....</subchild>
   </child>
</root>
```

Элемент в DTD определяется с помощью дескриптора `!ELEMENT`, в котором указываются название элемента и структура его содержимого.

Например, для элемента `<flower>` можно определить следующее правило:

```
<!ELEMENT flower (root+)>
```

Ключевое слово `ELEMENT` указывает, что данной инструкцией будет описываться элемент XML. Внутри этой инструкции задается название элемента (`flower`) и тип его содержимого, например, знак «+» указывает, что элемент `flower` включает по крайней мере один вложенный элемент `root`.

Атрибуты. Если при определении элемента необходимо задать какие-либо параметры, уточняющие его характеристики, то имеется возможность использовать `атрибуты элемента`. Атрибут — это пара `название = значение`, которую надо задавать при определении элемента в начальном тэге, например:

```
<color RGB="true">#ff08ff</color>
<color RGB="false">white</color>
```

Списки атрибутов элемента определяются с помощью ключевого слова `!ATTLIST`. Внутри него задаются названия атрибутов, типы их значений и дополнительные параметры. Например, для элемента `<article>` могут быть определены следующие атрибуты:

```
<!ATTLIST article
id ID #REQUIRED
```

```
about CDATA #IMPLIED
type (actual | review | teach ) 'actual' ''
```

В данном примере для элемента `article` определяются три атрибута: `id`, `about` и `type`, которые имеют типы `ID` (идентификатор), `CDATA` и список возможных значений соответственно. Всего существует шесть возможных типов значений атрибута:

- `CDATA` — содержимым документа могут быть любые символьные данные;
- `ID` — определяет уникальный идентификатор элемента в документе;
- `IDREF (IDREFS)` — указывает, что значением атрибута должно выступать название (или несколько таких названий, разделенных пробелами во втором случае) уникального идентификатора определенного в этом документе элемента;
- `ENTITY (ENTITIES)` — значение атрибута должно быть названием (или списком названий, если используется `ENTITIES`) компонента (макроопределения), определенного в документе;
- `NMTOKEN (NMTOKENS)` — содержимым элемента может быть только одно отдельное слово (т. е. этот параметр является ограниченным вариантом `CDATA`);
- список допустимых значений — определяется список значений, которые может иметь данный атрибут.

Также в определении атрибута можно использовать следующие параметры:

- `#REQUIRED` — определяет обязательный атрибут, который должен быть задан во всех элементах данного типа;
- `#IMPLIED` — атрибут не является обязательным;
- `#FIXED значение` — указывает, что атрибут должен иметь только указанное значение, однако само определение атрибута не является обязательным, но в процессе разбора его значение в любом случае будет передано программе-анализатору. `Значение` задает значение атрибута по умолчанию.

Сущности и специальные символы. Сущности (entity) представляют собой определения, содержимое которых может быть повторно использовано в документе. В языках программирования подобные элементы называются макроопределениями. Для того чтобы включить в документ символ, используемый для определения каких-либо конструкций языка (например, символ угловой

скобки) и не вызвать при этом ошибок в процессе разбора такого документа, нужно использовать его специальный символьный либо числовой идентификатор. Например, `<`, `>`, `&quot;` или `&#036;` (десятичная форма записи), `&#x1a` (шестнадцатеричная) и т. д.

Создаются DTD-сущности при помощи инструкции `!ENTITY`:

```
<!ENTITY hello ' Мы рады приветствовать Вас!' >
```

Программа-анализатор, просматривая в первую очередь содержимое области DTD-определений, обработает эту инструкцию и при дальнейшем разборе документа будет использовать содержимое DTD-сущности в том месте, где будет встречаться ее название. То есть теперь в документе мы можем использовать выражение `&hello;`, которое будет заменено на строчку «Мы рады приветствовать Вас!».

В общем случае внутри DTD можно задать следующие типы сущностей:

- внутренние — предназначены для определения строковой константы, с их помощью можно организовывать ссылки на часто изменяемую информацию, делая документ более читабельным. Внутренние компоненты включаются в документ при помощи знака амперсанта &;
- внешние — указывают на содержимое внешнего файла, причем этим содержимым могут быть как текстовые, так и двоичные данные. В первом случае в месте использования макроса будут вставлены текстовые строки, во втором — бинарные данные, которые анализатором не рассматриваются и используются внешними программами.

Комментариями является любая область данных, заключенная между последовательностями символов:

```
<!-- -->.
```

Комментарии пропускаются анализатором и поэтому при разборе структуры документа в качестве значащей информации не рассматриваются.

Директивы анализатора. Инструкции, предназначенные для анализаторов языка, описываются в XML-документе при помощи специальных тэгов `<?` и `?>`. Программа клиента использует эти инструкции для управления процессом разбора документа. Наиболее час-

то инструкции используются при определении типа документа (например, `<? Xml version="1.0"?>`) или создании пространства имен.

CDATA. Чтобы задать область документа, которую при разборе анализатор будет рассматривать как простой текст, игнорируя любые инструкции и специальные символы, но, в отличие от комментариев, иметь возможность использовать их в приложении, необходимо использовать тэги `<![CDATA[ и ]]>`. Внутри этого блока можно помещать любую информацию, которая может понадобиться программе-клиенту для выполнения каких-либо действий (в область CDATA можно помещать, например, инструкции JavaScript). Естественно, надо следить за тем, чтобы в области, ограниченной этими тэгами, не было последовательности символов]].

2.3.2. Семейство технологий XML

Помимо собственно языка разметки технологии XML включают набор средств, обеспечивающих возможности для определения типов данных и семантики ресурсов, интеграцию пространства имен, управление данными.

Спецификация Namespaces обеспечивает возможность применять специальные уточняющие обозначения к именам элементов и ссылкам, содержащимся в документах XML. Это позволяет избежать коллизий имен при использовании таких элементов, которые имеют одно и то же имя, но определены в разных словарях. Это также необходимо, если данные в документе получены из нескольких источников.

Пространства имен определены в рекомендации W3C. Для обеспечения их уникальности элементам и атрибутам присваиваются глобально уникальные имена с помощью ссылки URI.

Языки XSL и XSLT. Для представления документа XML в броузере может применяться спецификация каскадных таблиц стилей[1] (Cascading Stylesheet Specification — CSS), но она не позволяет вносить в документ структурные изменения. Поэтому W3C определил формальную рекомендацию по использованию языка расширяемых таблиц стилей (eXtensible Stylesheet Language — XSL), который соз-

[1] Стилем называется формализованное описание способа отображения информационных элементов.

дан не только для определения способа отображения в броузере XML-документа, но также для преобразования одного документа XML в другой.

Расширяемый язык таблиц стилей для поддержки преобразований (eXtensible Stylesheet Language for Transformations — XSLT) представляет собой подмножество языка XSL. Он одновременно является и языком разметки, и языком программирования, поскольку в нем предусмотрены механизмы преобразования структуры XML в любую другую структуру XML, в формат HTML или в любой другой текстовый формат.

RDF (Resource Description Framework) на сегодняшний день является наиболее перспективной и общеупотребительной моделью описания метаданных. Система создана международной организацией W3C при участии представителей различных заинтересованных организаций.

RDF является форматом представления сложных структурированных метаданных, при помощи которых можно описывать любой ресурс Web.

Объектная модель документа (DOM) определяет некоторый стандартный набор объектов для представления HTML- и XML-документов, а также методы для доступа к ним и выполнения операций над ними. В целом API-интерфейсы XML подразделяются на две категории: основанные на древовидном представлении и основанные на обработке событий.

Интерфейс объектной модели документа, предложенной WWW-консорциумом, представляет собой API-интерфейс для XML, основанный на древовидном представлении. Он описывает ряд независимых (от платформы и языка программирования) интерфейсов, позволяющих преобразовывать во внутреннюю форму любые формально правильные документы XML или HTML. Интерфейс DOM формирует древовидное представление XML-документа в оперативной памяти и предоставляет доступ к классам и методам, позволяющим выполнять такие структурные манипуляции, как добавление или удаление элементов, а также изменение порядка следования элементов.

XML-схемы (по сравнению с DTD) обладают более широкими возможностями для определения типов данных и интеграции пространства имен, реализуя тем самым *открытую модель данных*. Безусловным достоинством XML-схем является также то, что они позволя-

ют представлять правила формирования XML-документа средствами самого XML.

Основное назначение XML-схемы — обеспечить гибкий, хорошо структурированный способ описания[1] данных через определение их типов.

Достаточно обширный список простых типов базируется на общепринятых промышленных и международных стандартах.

Нестандартные типы образуются из предопределенных путем использования фасетов, задающих ограничения на содержание. Например, можно определить специальный тип, имеющий представление *integer*, но принимающий значения только в некотором диапазоне.

Каждый из элементов в схеме может иметь префикс. Префикс связан с именным пространством XML-схемы через объявление xmlns:xsd=http://URL.

Практически в XML-схему можно включить несколько подсхем, используя несколько элементов *include*, при этом включаемые подсхемы могут включать другие подсхемы.

Использование механизма схем фактически реализует компонентную структуру документа: использование внешних схем, помимо встроенного контроля содержания документов, обеспечивает хорошую возможность стандартизации типовых элементов и их повторное использование, что существенно при обмене и совместном использовании данных.

Спецификация XML-схем предоставляет более полный и строгий метод определения модели информационного наполнения документа XML, чем определения DTD. Но и она не обеспечивает поддержку необходимого уровня семантической совместимости. Например, если два приложения должны обмениваться информацией с помощью XML, то назначение и подразумеваемый смысл используемых при этом документов должны быть согласованы с той структурой данных, которая формируется при обмене, для чего необходимо создать модель предметной области с описанием данных, которые требуются для одного и другого приложения.

Поэтому может оказаться, что XML хорошо подходит для обмена данными только между приложениями, для которых уже известна

[1] В XML различаются определения и объявления. Определения создают новые типы элементов (простые и комплексные). Объявления задают имена и содержимое элементов и атрибутов (простых и комплексных), которые могут использоваться в документах, соответствующих данной схеме.

модель информационного наполнения, но плохо подходит для тех ситуаций, когда к обмену данными присоединяются новые приложения.

2.4. Текстовый процессор

2.4.1. Назначение

Средства, предназначенные для подготовки текстов, условно можно разделить на редакторы (подготовка писем и других простых документов без автоматического формирования строк и страниц) и процессоры (оформление документов с автоматическим формированием и оформлением страниц, с разными шрифтами, включающих графики, рисунки и др.). Значительной популярностью пользуется текстовый процессор MS Word for Windows.

Текстовый процессор реализует следующие функции:

- создание, открытие, закрытие, сохранение текстовых документов (опция Главного меню «Файл»);
- задание параметров страниц (вкладка меню «Разметка страницы» — рис. 2.6);
- набор текста (режим прописных букв, гарнитура, кегль и цвет шрифта, работа с выделенным фрагментом текста, межстрочный интервал, способы выравнивания, буфер обмена — вкладка меню «Главная»);
- форматирование абзаца — задание параметров абзаца, красная строка, межстрочный интервал, установка рамки и заливки абзаца (вкладка меню «Главная» → «Абзац» — рис. 2.7);

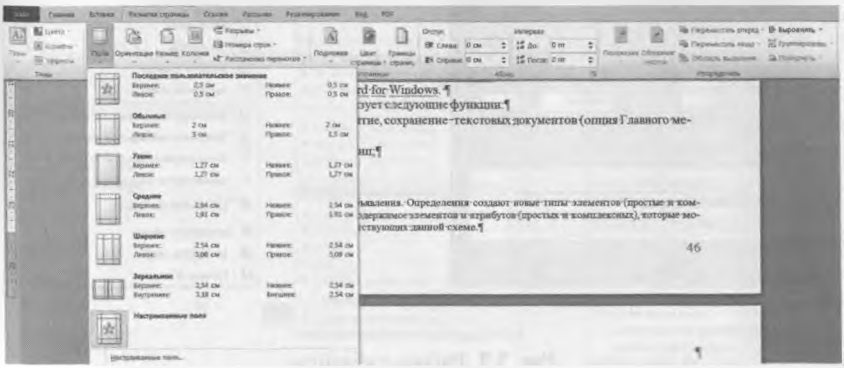
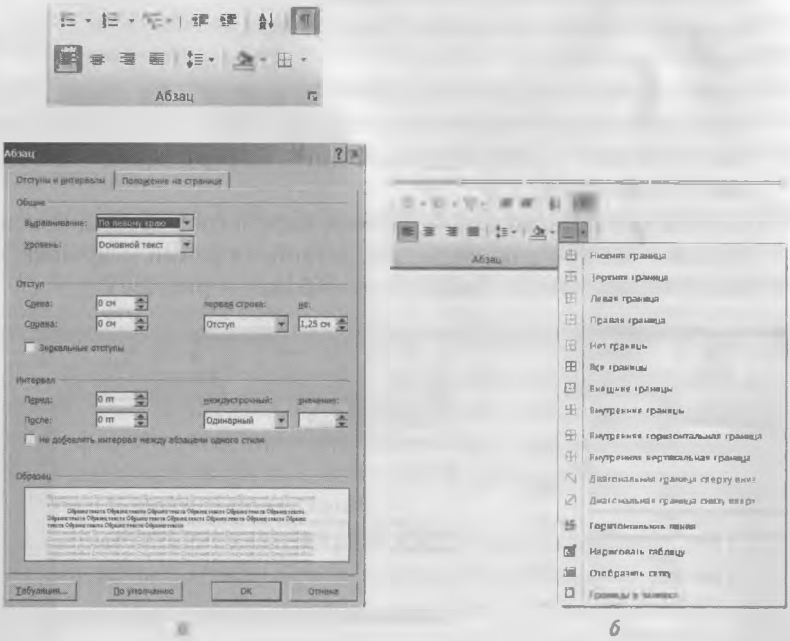


Рис. 2.6. Задание параметров страницы

- задание параметров шрифта (вкладка меню «Главная» → «Шрифт» — рис. 2.8);
- создание нумерованных и маркированных списков, настройка нумерованных списков (вкладка меню «Главная» → «Абзац» — рис. 2.9);
- ссылки, заголовки, оглавления (вкладка меню «Ссылки»);
- вставка и редактирование объектов — рисунков, клипов, MIDI-файлов, математических формул (вкладка меню «Вставка»);
- проверка правописания, расстановка переносов (вкладки меню «Рецензирование» → «Правописание», «Разметка страницы» → «Параметры страницы»);
- создание, заполнение и форматирование таблиц; рамки, заливка; изменение структуры таблицы (добавление и удаление строк и столбцов, объединение ячеек, изменение размеров ячеек); преобразование текста в таблицу и наоборот (вкладка меню «Вставка» → «Таблицы» — рис. 2.10);



а *б*

Рис. 2.7. Работа с абзацем:
а — задание параметров абзаца; *б* — установка рамки

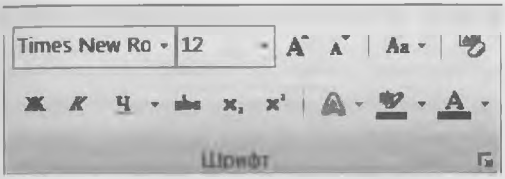


Рис. 2.8. Задание параметров шрифта

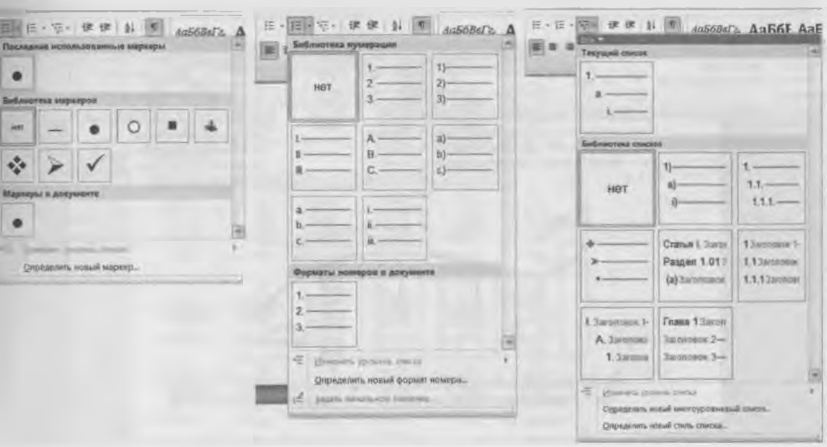


Рис. 2.9. Создание и настройка списков

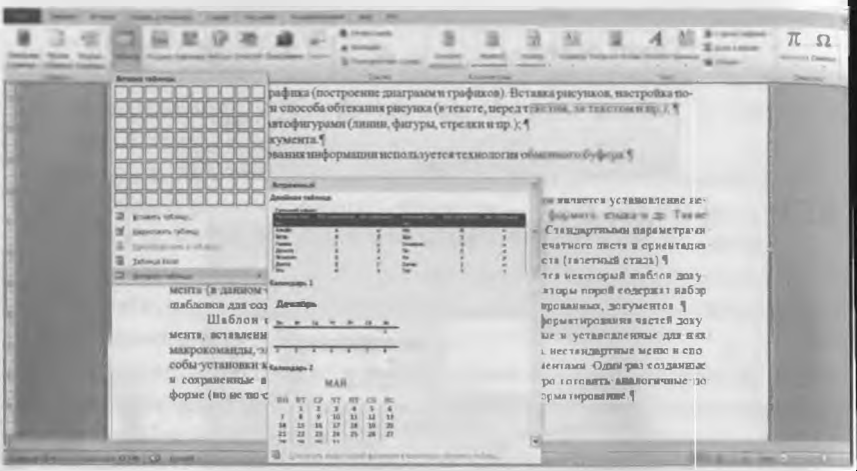


Рис. 2.10. Работа с таблицами

- деловая графика (построение диаграмм и графиков), работа с автофигурами (линии, фигуры, стрелки и пр., вставка рисунков (вкладка меню «Вставка» → «Иллюстрации»);
- настройка положения, размера и способа обтекания рисунка (вкладка меню «Формат», рис. 2.11);
- печать документа.

Для копирования информации используется технология обменного буфера.

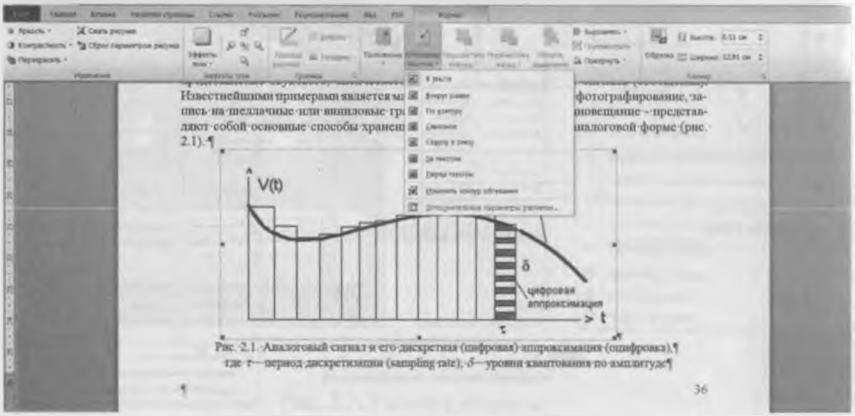


Рис. 2.11. Работа с рисунками

2.4.2. Оформление документа

Важным элементом работы с любым текстовым процессором является установление некоторых начальных параметров, например, параметров страницы, формата, языка и др. Такие процедуры называют оформлением структурных элементов текста. Стандартными параметрами оформления страниц документа являются: поля страниц; размер печатного листа и ориентация текста на бумаге; расположение колонтитулов; число колонок текста (газетный стиль).

Таким образом, при создании нового документа предлагается некоторый шаблон документа (в данном случае он называется «Normal»). Текстовые редакторы содержат набор шаблонов для создания различных типовых, а порой и стандартизированных документов.

Ш а б л о н содержит информацию о стилях форматирования частей документа, вставленных полях и т. д. В шаблонах хранятся выбранные и установленные для них макрокоманды, элементы глоссария, кнопки панели инструментов, нестандартные меню и способы установки клавиш сокращения, облегчающих работу с документами. Один раз созданные и сохраненные в памяти компьютера, шаблоны позволяют быстро готовить аналогичные по форме (но не по содержанию) документы без затрат времени на форматирование (рис. 2.12).

Пользователи могут создавать собственные шаблоны. Для этого необходимо выполнить определенную последовательность шагов. Рассмотрим их на примере создания бланка предприятия.

Рис. 2.12. Коллекция доступных шаблонов документов

Первоначально следует установить параметры страницы бланка (размер страницы, величины полей).

Затем в бланк встраивают постоянные реквизиты, соответствующие требованиям государственного стандарта или международных правил оформления документов. При этом используются возможности программы по вставке графических изображений эмблемы предприятия; заданию параметров для шрифтового исполнения различных частей шаблона документа; установке атрибутов оформления абзацев основной и заголовочной части документа и т. д. В шаблон можно включить глоссарий часто употребляемых слов и фраз деловой лексики для данного типа документов, язык документа (например, русский) и другие функции.

По окончании формирования шаблона его сохраняют как «шаблон документа», т. е. с расширением «dot» (document type, тип документа).

Созданная таким образом коллекция шаблонов документов может использоваться постоянно, в том числе всеми сотрудниками предприятия, что приводит к единообразию оформления документов предприятия и уменьшает время изготовления конкретного документа.

Одним из основных структурных элементов любого документа является абзац. При наборе текста новый абзац образуется после нажатия <Enter>. При этом курсор ввода переходит на новую строку и устанавливается в позицию левого отступа следующего абзаца. Позиция отступа зависит от параметров настройки конкретной системы текстовой обработки. К наиболее общим параметрам абзацного форматирования можно отнести:

- выравнивание границ строк;
- отступы для строк;
- межстрочные интервалы;
- обрамление и цвета фона текста;
- расположение текста абзаца на смежных страницах документа.

Если система подготовки текста используется для создания и оформления многостраничного документа, то применяется форматирование страниц или разделов. В тексте могут появиться новые структурные элементы: закладки, сноски, перекрестные ссылки, колонтитулы.

Под закладкой или меткой понимается определенный фрагмент текста документа, которому пользователь присваивает имя.

В многостраничном документе закладка может использоваться для быстрого перехода к месту документа, обозначенному закладкой, или для создания перекрестных ссылок в документе.

Перекрестная ссылка — указание, предлагающее читателю документа обратиться к другому фрагменту текста или рисунку, содержащемуся в тексте. Например: «*Вернитесь к разделу «Базовые функции редактирования текста» (страница ###)*».

Функции управления закладками и перекрестными ссылками размещены на вкладке меню «Вставка» → «Ссылки» (рис. 2.13).

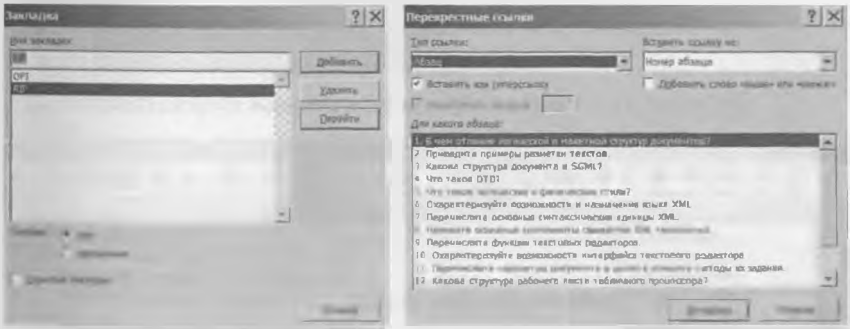


Рис. 2.13. Работа с закладками и перекрестными ссылками

Иногда документ содержит дополнения к основному тексту, подстрочные примечания. Подстрочные примечания оформляют с н о с к а м и. В состав подстрочного примечания входят два неразрывно связанных элемента: знак сноски и текст собственно примечания. Знак сноски располагают в основном тексте у того места, к которому относится примечание, и в начале самого примечания.

Управление сносками — на вкладке меню «Ссылки» (рис. 2.14).

Колонтитулом называют одинаковый для группы страниц текст и (или) графическое изображение, расположенное на полях печатной страницы вне основного текста документа. Различают верхний колонтитул — над текстом документа и нижний — под основным текстом. Порядковые номера страниц входят в колонтитул. Их называют колонцифрами (рис. 2.15).

Большинство текстовых процессоров поддерживает концепцию составного документа — контейнера, включающего различные объекты. Пользователь может вставлять в текст документа рисунки, таблицы, графические изображения, подготовленные в других про-

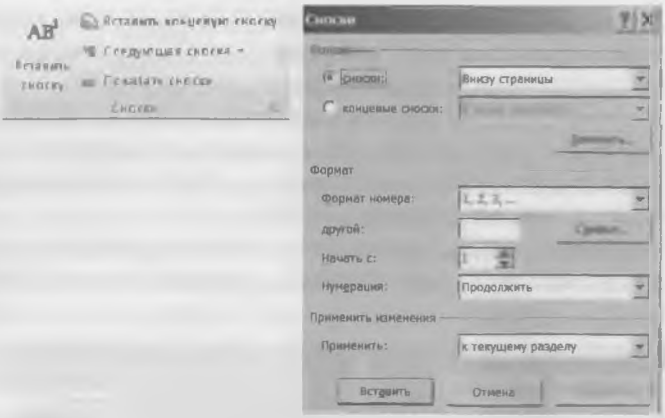


Рис. 2.14. Организация сносок

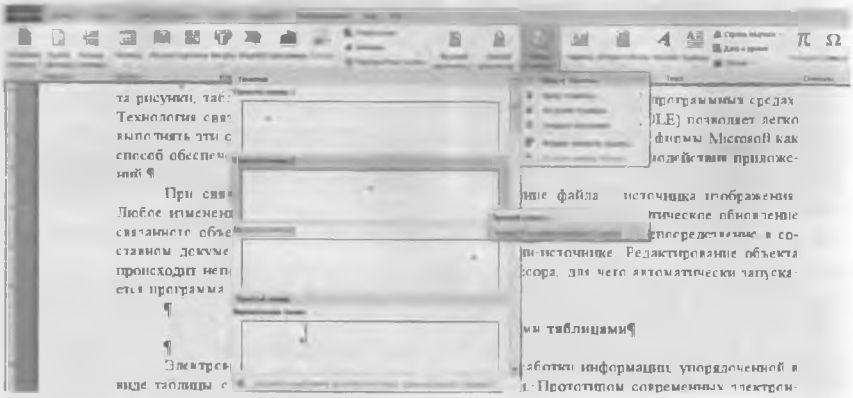


Рис. 2.15. Вставка колонцифры

граммных средах. Технология связи и внедрения объектов (Object Linking and Embedding — OLE) позволяет легко выполнять эти сложные задачи.

При связывании (Linking) отслеживается положение файла — источника изображения. Любое изменение данных этого файла с помощью OLE вызывает автоматическое обновление связанного объекта. При встраивании объекта (Embedding) он хранится непосредственно в составном документе вместе с информацией о приложении-источнике. Редактирование объекта происходит непосредственно из среды текстового процессора, для чего автоматически запускается программа, умеющая его редактировать.

Контрольные вопросы

1. В чем отличие логической и макетной структур документов?
2. Приведите примеры разметки текстов.
3. Какова структура документа в SGML?
4. Что такое DTD?
5. Что такое логические и физические стили?
6. Охарактеризуйте возможности и назначение языка XML.
7. Перечислите основные синтаксические единицы XML.
8. Назовите основные компоненты семейства XML-технологий.
9. Перечислите функции текстовых редакторов.
10. Охарактеризуйте возможности интерфейса текстового редактора.
11. Перечислите параметры документа в целом и опишите методы их задания.

Глава 3
МУЛЬТИМЕДИЙНЫЕ ТЕХНОЛОГИИ

Появление систем мультимедиа, безусловно, производит большие изменения в таких областях, как образование, компьютерный тренинг, во многих сферах профессиональной деятельности, науки, искусства, в компьютерных играх и т. д.

Мультимедиа представляет собой объединение в один интерактивный продукт информации, представленной различными способами — текст, неподвижные изображения (рисунки и фотографии), движущиеся изображения (мультипликация и видео) и звук (цифровой и MIDI).

3.1. Технологии обработки аудиоинформации

В то время как даже в нецифровых технологиях фото и видео в принципе могут быть найдены элементы дискретности (в первом случае зерна пленки и фоточувствительные ячейки иконоскопа или матрица ПЗС, во втором — то же и плюс разбиение изображения на строки), звуковой сигнал в своей основе является *чисто аналоговым*[1] (если не вдаваться в такие тонкости, как магнитные домены в чувствительном слое при записи на ленту).

Звуковая плата ПК содержит несколько аппаратных систем, связанных с формированием и сбором аудиоданных, две основных аудиоподсистемы, предназначенные для цифрового «аудиозахвата», синтеза и воспроизведения музыки. Исторически подсистема синтеза и воспроизведения музыки генерирует звуковые волны одним из двух способов:

- через внутренний синтезатор (например, ЧМ-синтезатор);
- проигрывая ранее оцифрованные звуковые фрагменты (sampled).

[1] Основы оцифровки аналогового сигнала представлены в главе 2.

Секция цифровой звукозаписи звуковой платы состоит из пары преобразователей (ЦАП и АЦП) и содержит программируемый генератор частоты выборки, синхронизирующий преобразователи и управляемый от ЦП.

Генератор звука, установленный на плате, использует процессор цифровых сигналов (Digital Signal Processor — DSP), который проигрывает требуемые музыкальные ноты, объединяя их считывание из различных областей звуковой таблицы с различными скоростями, чтобы получить требуемую высоту тона. Максимальное количество доступных нот определяется мощностью DSP-процессора и называется полифонией платы. DSP-процессоры используют сложные алгоритмы, чтобы создать эффекты наподобие реверберации, хорового звучания и запаздывания. Реверберация создает впечатление, что инструменты играют в больших концертных залах. Хор используется, чтобы создать впечатление, что несколько инструментов играют совместно, тогда как фактически есть только один. Добавление запаздывания к партии гитары, например, может дать эффект пространства и стереозвучания.

3.1.1. Методы синтеза звука

Частотная модуляция. Синтез с использованием частотной модуляции (ЧМ, FM-synthesis) основывается на последовательном и параллельном подключении генераторов простых сигналов и их взаимомодуляции. Схема соединения генераторов и параметры каждого сигнала (частота, амплитуда и закон их изменения во времени) определяет тембр звучания, а количество генераторов и степень тонкости управления ими определяют предельное количество синтезируемых тембров. Данный метод очень хорош с точки зрения дешевизны реализации, но при этом требует сложного программирования и тонкой настройки.

Каждый голос ЧМ-синтезатора требует минимум двух генераторов сигнала, обычно называемых «операторами». Различные конструкции ЧМ-синтезатора имеют различные степени управления параметрами оператора. Сложные системы ЧМ могут использовать 4 или 6 операторов на каждый голос.

Хотя системы ЧМ в аналоговом исполнении были реализованы в ранних клавиатурных синтезаторах, в дальнейшем синтез осуществлялся в цифровой форме. Методы синтеза ЧМ очень полезны для

того, чтобы создать выразительные новые звуки. Однако если цель синтезирующей системы состоит в том, чтобы воспроизвести звук некоторого классического инструмента, это лучше делать в цифровой форме на основе выборок сигналов, как при синтезе с использованием звуковых таблиц (WaveTable Synthesis).

Табличный синтез (WaveTable synthesis, или PCM-synthesis). Здесь используются выборки звуков реальных инструментов — образцов (samples), «кусочков», определенный набор которых позволяет создать звучание инструмента.

В то время как все звуковые платы ЧМ звучат аналогично, платы звуковых таблиц значительно отличаются по качеству, что определяется несколькими факторами:

- качество первоначальной записи;
- частота, на которой выборки были записаны;
- количество выборок, использованных для каждого инструмента;
- методы сжатия, использованные для сохранения выборки.

Большинство инструментальных выборок записаны в стандарте 16 бит и 44,1 кГц, но многие изготовители сжимают данные так, чтобы больше выборок можно было записать в ограниченный объем памяти, хотя сжатие часто приводит к потере качества.

Когда аудиокассета воспроизводится слишком быстро или слишком медленно, ее высота звучания меняется, и это также справедливо для цифровой звукозаписи. Проигрывание выборки на более высокой скорости, чем ее оригинал, приводит к более высокому воспроизводимому звуку, позволяя инструментам исполнять более чем несколько октав. Однако если некоторые тембры воспроизводятся быстро, они звучат слишком слабо и тонко; аналогично, когда выборка проигрывается слишком медленно, она звучит уныло и неестественно. Чтобы преодолеть эти эффекты, изготовители разбивают клавиатуру на несколько областей и применяют различные выборки звуков инструментов в каждой из них.

Каждый инструмент звучит с различным тембром в зависимости от стиля игры. Например, при мягкой игре на фортепьяно не слышен звук молотков, бьющих по струнам. При более интенсивной игре этот звук становится более очевидным.

Для каждого инструмента должны быть записаны много выборок и их разновидностей, чтобы синтезатор точно воспроизвел соответствующий диапазон звука, а это неизбежно требует большего количества памяти. Точное воспроизведение фортепьяно соло, например, тре-

бует от 6 до 10 Мбайт данных, вот почему нет никакого сравнения между синтезируемым и реальным звуком.

Аддитивный, или суммирующий, синтез (additive synthesis). Данный метод прекрасно иллюстрируют первые модели от Hammond, которые были основаны на принципе построения звучания реальных органов. В его основе лежит метод создания сложных гармонически насыщенных звуков из простых изменяющихся синусоидальных волн, различных по амплитуде и/или частоте.

Вычитающий синтез (subtractive synthesis). Данный метод противоположен предыдущему. В качестве исходного берется тембрально богатый, насыщенный гармониками звук, а потом в результате сложной фильтрации из него формируется определенный тембр с характерной тоновой окраской. Вычитающий синтез активно использовался в аналоговых синтезаторах, а сейчас применяется в программных моделях для получения «аналогового эффекта».

Непосредственное формирование (Direct Draw). В ряде синтезаторов используются осцилляторы, генерирующие звуковые волны со стандартными формами (синусоида, прямоугольная, пилообразная и т. п.). В варианте Direct Draw пользователь может самостоятельно рисовать в любом звуковом редакторе любые формы и после использовать их в качестве звукового фрагмента.

Гранулированный синтез (Granular synthesis). Является частным случаем табличного синтеза. Звук формируется из коротких фрагментов звуковой волны. В результате взаимодействия частоты их повторения и частотных составляющих сэмплированной звуковой формы получается тембрально сложный монотонный звук, который впоследствии можно обрабатывать методами вычитающего синтезирования.

Сэмплинг (Sample playback). Данный метод базируется на использовании образцов звуков отдельных инструментов и воспроизведении их в режиме обычного проигрывателя. Небольшие звуковые фрагменты, из которых складывается звучание инструмента, загружаются в память и затем воспроизводятся.

Линейно-арифметический синтез (Linear/Arithmetic (L/A) synthesis). За основу концепции L/A synthesis было взято смешивание небольшого фрагмента сэмпла «живого» инструмента с синтезированной волновой формой. Этот метод позволяет дать натуральную звуковую окраску, близкую к реальному звучанию.

Синтез физического моделирования (Physical modeling synthesis). За основу данного метода берется сложная математическая модель, которая полностью описывает формирование звука в инструменте, хотя

до полноценного математического повторения реальных физических процессов еще далеко.

Синтез по математической функции (Mathematical function synthesis). Также частный случай физического моделирования, с помощью которого можно вкладывать математические функции, объединять их в функциональные блоки, а из них создавать математические алгоритмические модели. Вернее сказать, что этот метод является одним из простейших разделов физического моделирования.

Спектральный синтез (Spectral synthesis). Это даже не метод, а скорее способ создания сложных гармонических звуков на основе спектрограмм (графическое представление зависимости частоты от амплитуды).

3.1.2. Цифровой интерфейс музыкальных инструментов (MIDI)

MIDI (Musical Instrument Digital Interface) появился в начале 1980-х гг. и был разработан для того, чтобы обеспечить стандартный интерфейс между пультами управления музыкой (наподобие клавиатур) и звуковыми генераторами (типа синтезаторов и «роботов-барабанщиков»).

На уровне электросигналов MIDI представляет полудуплексную токовую петлю, которая пропускает последовательный поток данных по 8 битов на скорости передачи 31,25 килобод.

На уровне передачи информации MIDI представляет собой что-то вроде языка для описания музыкальных тактов и эффектов в реальном масштабе времени. Он обеспечивает соединение более чем по 16 каналам, позволяя подключить до 16 инструментов MIDI к одному интерфейсу.

Интерфейс MIDI передает не звук, а команды, которые выполняет устройство-приемник. Например, если на клавиатуре нажата определенная клавиша, то передается команда `Note On` (включить ноту), которая заставляет принимающее устройство проиграть некоторую музыкальную ноту.

Команда состоит из трех элементов:

- байт состояния (`Status Byte`);
- номер ноты (`Note Number`);
- значение скорости нажатия клавиши (`Velocity Value`).

Байт состояния содержит информацию о типе команды (в этом случае — «включить ноту»), а также, на какой канал она должна быть послана (1—16).

Номер ноты описывает клавишу, которая была нажата (скажем, «ре» большой октавы).

Значение скорости указывает силу, с которой эта клавиша была нажата. Принимающий инструмент будет исполнять эту ноту, пока не придет команда Note Off (отключить ноту), которая содержит аналогичные данные.

В зависимости от того, какой именно звук проигрывается, синтезаторы по-разному обрабатывают данные Velocity Value. Звук фортепьяно, например, становится громче, если клавиша нажата более сильно, а также изменяются тональные свойства. Профессиональные синтезаторы часто вводят дополнительные тембры, чтобы подражать звуку молоточков, ударяющих по струнам.

Число голосов (MIDI-каналов), или полифония звуковой платы, определяет максимальное количество элементарных звуков, которые плата может воспроизвести одновременно. Это число иногда указывают в названии звуковой карты, например: SB 16, AWE 64, SB PCI 64, SB PCI 128 и т. д.

Существует несколько разновидностей стандарта MIDI — 6M, GS и т. д. Практически все современные звуковые адаптеры совместимы со стандартом GM (General MIDI — единый или общий MIDI).

MIDI секвенсоры. По существу, секвенсор (sequencer) представляет собой цифровой магнитофон, который записывает и воспроизводит команды MIDI, а не аудиосигналы. Первые секвенсоры имели небольшую память и были способны к запоминанию только от 1 до 2 тыс. музыкальных тактов. Использование секвенсоров позволяет удобно редактировать музыкальные фразы и синхронизировать их с фильмом.

Сэмплер — синтезатор, у которого для хранения образцов звучания (сэмплов) вместо постоянной памяти (ROM) используется оперативная память большого объема (RAM). Пользователь перед каждым сеансом работы загружает в память уже готовые звуки или записывает новые звуки точно так же, как на обычный магнитофон. Впоследствии все эти сэмплы воспроизводятся с разной высотой под управлением клавиатуры или секвенсора.

В сэмплере каждый звук создается в нескольких источниках, сигналы которых смешиваются между собой. Каждый такой источник принято называть *леером* (от англ. layer — слой). Главным элементом любого леера является генератор — именно в нем образуется звук при

воспроизведении сэмпла. Иногда генератор сэмплера называют осциллятором. Сэмплы находятся в оперативной памяти устройства и извлекаются оттуда при поступлении соответствующей команды от программы управления.

Генератор воспроизводит сэмпл с разной высотой, в зависимости от поступающей в него команды MIDI `Note` (MIDI нота). Причем сэмпл может воспроизводиться как линейно, т. е. от начала до конца, так и зацикливаться. В последнем случае инструмент звучит ровно столько, сколько времени удерживается в нажатом состоянии клавиша на MIDI-клавиатуре. Помимо изменения высоты тона генератор изменяет уровень воспроизводимого сэмпла в зависимости от сообщения `Velocity` (Скорость нажатия клавиши).

Амплитудой колебаний LFO можно управлять с помощью генератора огибающей (Envelope Generator), создающего произвольную огибающую. Этот метод называется амплитудной модуляцией (AM — Amplitude Modulation). Но в любом сэмплере с помощью амплитудной модуляции можно управлять не только параметрами генератора низкой частоты, но и параметрами воспроизведения сэмпла. Например, если указано «время линейной атаки» 1 с, то после нажатия клавиши громкость сэмпла будет линейно возрастать от минимальной громкости к максимальной в течение 1 секунды. Если указывается время затухания (Release) 0,5 с, то после отпускания клавиши сэмпл будет звучать указанное время, причем его громкость будет линейно уменьшаться. Естественно, можно «нарисовать» и более сложные огибающие.

К сэмплу, который воспроизводится генератором с разной высотой и уровнем в зависимости от поступающих с клавиатуры команд `MIDI Note` (MIDI нота) и `Velocity` (Скорость нажатия клавиш), можно применить два вида модуляции: частотную и амплитудную. В первом случае будет периодически меняться высота воспроизводимого сэмпла относительно взятой на клавиатуре ноты, а во втором — его относительный уровень во время звучания. Если же мы применяем амплитудную модуляцию к генератору низкой частоты, то можно будет управлять также изменениями амплитуды колебаний высоты в течение времени.

Эквалайзер. Для управления тембром звука используются эквалайзеры — программно-аппаратные средства, способные понижать или повышать уровень разных частотных полос. При этом понижается или повышается относительный уровень разных гармоник сигнала, в результате чего мы в акустических системах слышим изменение тембра звука.

Известно два основных типа эквалайзеров — графические и параметрические. Первые отличаются наличием фиксированного количества полос: их обычно бывает 15 (можно менять уровень каждой $^2/_3$ звукового диапазона) или 30 (можно менять уровень каждой $^1/_3$ октавы звукового диапазона). На любой из полос уровень сигнала может опускаться или подниматься на 10—15 дБ. Параметрические эквалайзеры, в отличие от графических, могут настраиваться на любую частотную полосу любой ширины и поднимать/опускать ее уровень.

3.1.3. Единый стандарт MIDI (General MIDI)

Введение стандарта MIDI позволило создавать аранжировки, используя любые инструменты, имевшиеся в наличии MIDI. Но когда созданные файлы проигрывались на другом синтезаторе, не было никакой гарантии, что звучание будет тем же самым, потому что различные изготовители могли назначить инструментам различные номера программ, так что фортепьяно, записанное на одном синтезаторе, может прозвучать как труба на другом и пр.

В сентябре 1991 г. Ассоциация изготовителей MIDI (MMA) и Японский комитет стандартов MIDI (JMSC) положили начало новому этапу в технологии MIDI, приняв стандарт «Общая Система MIDI, уровень 1» (General MIDI System Level 1 — GM или GM1). Спецификация разработана, чтобы обеспечить совместимость функционирования инструментов, и налагает на звукогенерирующие устройства (клавиатура, звуковой модуль, звуковая плата, программные продукты) ряд требований:

- должно быть доступно одновременно минимум 24 канала («голоса») для звуков мелодии и ударных инструментов или 16 каналов для мелодии плюс 8 для ударных;
- должны поддерживаться все 16 каналов MIDI, каждый из которых способен воспроизвести различное число голосов (полифония) или различные инструменты (звук, аккорд, тембр);
- может выполняться одновременно минимум 16 различных тембров, воспроизводящих различные инструменты. Поддерживается как минимум 128 предварительно настроенных инструментов (номера MIDI-программ), соответствующих Инструментальной карте GM1 (GM1 Instrument Patch Map), и 47 звуков ударных, которые соответствуют Карте ударных GM1 (GM1 Percussion Key Map). Мелодический набор

состоит из 16 групп инструментов по 8 в каждой группе (фортепиано, органы, струнные, духовые, гитары, и т. п.).

За всеми инструментами были закреплены конкретные номера, поэтому мелодия, записанная в GM, будет одинаково звучать на разных GM-синтезаторах.

3.1.4. Форматы записи-воспроизведения аудиосигналов

Формат MP3

MP3 — сокращение от MPEG Layer3. Это один из основных цифровых форматов хранения аудио, утвержденный как часть стандартов сжатого видео и аудио MPEG1 и MPEG2. Данная схема является наиболее сложной схемой семейства MPEG Layer 1/2/3. Она требует наибольших затрат машинного времени для кодирования по сравнению с двумя другими и обеспечивает более высокое качество кодирования. Используется главным образом для передачи аудио в реальном времени по сетевым каналам и для кодирования Audio CD.

Высокая степень компактности MP3 при сохранении качества звучания достигается с помощью дополнительного квантования по установленной схеме, позволяющей минимизировать потери качества. Это достигается за счет учета особенностей человеческого слуха, в частности, эффекта маскирования слабого сигнала одного диапазона частот более мощным сигналом соседнего диапазона или мощным сигналом предыдущего фрейма, вызывающего временное понижение чувствительности уха к сигналу текущего фрейма. Принимается во внимание также неспособность большинства людей различать сигналы, по мощности лежащие ниже определенного уровня, разного для разных частотных диапазонов. Этот метод называется *адаптивным кодированием* и позволяет экономить на наименее значимых с точки зрения восприятия человеком деталях звучания.

Степень сжатия и, соответственно, объем дополнительного квантования определяются не форматом, а самим пользователем при задании параметров кодирования. Ширина потока, или битрейт (bitrate), может изменяться от наибольшего для MP3 (320 кбит/с) до 96 кбит/с и даже ниже. Термин «битрейт» обозначает общую ширину потока, независимо от того, монофонический или стереофонический сигнал он содержит.

При испытаниях опытные эксперты, специализирующиеся на оценке качества звучания, не смогли различить звучание оригиналь-

ного трека на CD и закодированного в MP3 с коэффициентом сжатия 6:1, т. е. с битрейтом в 256 кбит/с.

Более низкие битрейты, несмотря на их популярность, не дают возможности обеспечить надлежащее качество кодирования. Объективно и 256 кбит/с не дает возможности осуществить полностью обратимое кодирование, то же самое можно сказать и про наивысший битрейт — 320 кбит/с, но отличия от CD Audio, по которому кодируется тестовый MP3, сравнимы с отличиями самого CD Audio от исходного высококачественного сигнала, из которого он был получен путем оцифровки.

Файл формата MP3 может также содержать информацию о файле непосредственно в заголовке: имя исполнителя, графику (альбом диска), URL для дальнейшей информации, текст песни, и т. д.

Процесс кодирования. Перед кодированием исходный сигнал разбивается на участки, называемые фреймами, каждый из которых кодируется отдельно и помещается в конечный файл независимо от других. Последовательность воспроизведения определяется порядком расположения фреймов. Каждый фрейм может кодироваться с разными параметрами. Информация о них содержится в заголовке фрейма.

Кодирование начинается с того, что исходный сигнал с помощью фильтров разделяется на несколько составляющих, представляющих отдельные частотные диапазоны, сумма которых эквивалентна исходному сигналу. Для каждого диапазона определяется величина маскирующего эффекта, создаваемого сигналами соседних диапазонов и сигналом предыдущего фрейма. Если она превышает мощность сигнала интересующего диапазона или мощность сигнала в нем оказывается ниже определенного опытным путем порога слышимости, то для данного фрейма данный диапазон сигнала не кодируется. Для оставшихся данных каждого диапазона определяется, сколькими битами на сэмпл можно пожертвовать, чтобы потери от дополнительного квантования были ниже величины маскирующего эффекта. После завершения работы психоакустической модели формируется итоговый поток, который дополнительно кодируется по Хаффману.

Кодирование стереосигнала может осуществляться четырьмя методами:

- Dual Channel — каждый канал получает ровно половину потока и кодируется отдельно, как моносигнал. Используется главным образом в случаях, когда разные каналы содержат принципиально разный сигнал — скажем, текст на различных языках;

- Stereo — каждый канал кодируется отдельно, но кодер может принять решение отдать одному каналу больше места, чем другому. Это может быть полезно в том случае, когда после исключения части сигнала, лежащей ниже порога слышимости или полностью маскируемой, оказалось, что код не полностью заполняет выделенный для данного канала объем, и кодер имеет возможность использовать это место для кодирования другого канала. Этим, например, устраняется кодирование «тишины» в одном канале, когда в другом есть сигнал. Данный режим используется на битрейтах выше 192 кбит/с. Применим и на более низких битрейтах порядка 128—160 кбит/с;
- Joint Stereo (MS Stereo) — стереосигнал раскладывается на средний между каналами и разностный. При этом второй кодируется с меньшим битрейтом. Это позволяет несколько увеличить качество кодирования в обычной ситуации, когда каналы по фазе совпадают. Но приводит и к резкому его ухудшению, если кодируются сигналы, не совпадающие по фазе;
- Joint Stereo (MS/IS Stereo) — вводит еще один метод упрощения стереосигнала, повышающий качество кодирования на особо низких битрейтах. Состоит в том, что для некоторых частотных диапазонов оставляется уже даже не разностный сигнал, а только отношение мощностей сигнала в разных каналах. Очевидно, для кодирования этой информации употребляется еще меньший битрейт. MS Stereo — частный случай MS/IS Stereo, когда переменная, отвечающая за кодируемый таким образом диапазон, принимает нулевое значение.

Скорости передачи. На низких битрейтах всегда срезаются мелкие, сравнительно тихие детали, наличие или отсутствие которых нередко серьезно меняет эмоциональную окраску композиции, придает или лишает ее таких эффектов, как ощущение «кристальной чистоты» звука (в той мере, в которой она присутствует в CD Audio).

Различие между качеством звука на битрейтах 128 кбит/с и 256 кбит/с — 320 кбит/с принципиально. Первый к качеству уровня CD никакого отношения не имеет, в отличие от двух последних.

Другие форматы

WAV. Формат WAV является метаформатом для данных любого типа. Имеет стандартный заголовок и описания областей данных, которых может быть несколько, способ же кодирования аудиосигнала

может быть каким угодно. Вполне могут содержаться данные, не имеющие отношения к аудио.

Каждый метод кодирования, указываемый в заголовке, имеет собственный идентификатор, в соответствии с которым Windows и определяет, установлен ли кодек для работы с данным файлом, и если установлен — подключает его.

Кодеки, индивидуальные для каждого подформата, регистрируются в системе при их установке, после чего становится возможным использовать WAV-файлы, содержащие аудиоданные в форматах, поддерживаемых данными кодеками.

WMA. Отличительной чертой WMA является очень быстрое кодирование файлов, а также наилучшее качество звука при очень сильном сжатии. Формат WMA подходит для кодирования самых разных записей, однако, начиная с 8 кГц, он начинает искажать частоты, а звук выше 20 кГц полностью обрезается.

OGG Vorbis. Механизм компрессии использует схожие с MP3 психоакустические модели, но по гораздо более точным методикам. Кроме того, вся запись разбивается на несколько участков, и каждый из них кодируется с разным сжатием (это аналогично Variable Bit Rate в MP3).

В файле OGG может содержаться до 255 каналов, т. е. можно кодировать многоканальные потоки вроде Dolby Digital. Кроме того, в OGG-файлы можно встраивать графические изображения и тексты, которые могут возникать по ходу воспроизведения.

MP3Pro. Формат MP3Pro в отличие от стандартного MP3 содержит два потока, один из которых обычный Layer 3-поток, а второй содержит информацию, на основе которой декодер восстанавливает самые верхние частоты. Поэтому файл, сжатый с использованием MP3Pro, может быть воспроизведен и обычным проигрывателем, но с частотой дискретизации 22 кГц, так как плеер воспримет только первый поток.

DVD-аудио. С появлением DVD производители CD начали создавать стандарты более высококачественного воспроизведения. Среди них — аудиокомпакт-диск высшего качества (SACD, или Super Audio CD), диск цифровой звукозаписи (DAD, или Digital Audio Disc).

Эти стандарты предполагают диски с разрешающей способностью (уровни квантования) 24 бита и частотой выборки в 96 кГц, в противоположность обычному CD с форматом 16 бит и 44,1 кГц. Кроме того, формат SACD обладает обратной совместимостью с существующими проигрывателями.

DVD-аудио поддерживает широкое разнообразие аудиоформатов при изменении уровней спецификации, например, те же самые многоканальные аудиоформаты, которые используют в DVD-видео. Поэтому и DVD-видео и DVD-аудио могут обеспечить высокое качество звучания многоканальной записи в аудиоформатах Dolby Digital и DTS. Однако основное преимущество спецификации DVD-аудио сравнительно с DVD-видео и компакт-диском заключается в значительном увеличении качества при записи в аудиоформате PCM (Pulse Code Modulation — импульсно-кодовая модуляция).

DVD-аудио обеспечивают значительно более высокое качество PCM, чем возможно на компакт-диске или DVD-видео. DVD-аудио PCM может быть записан с диапазоном частот, который более чем в четыре раза шире, чем для CD, что придает живость и выразительность. DVD-аудио PCM также имеет намного больший динамический диапазон, чем это возможно на компакт-диске.

3.1.5. Программные средства записи-воспроизведения звука

Большинство MP3-файлов производится из аудиокомпакт-дисков. Это двухступенчатый процесс, первая стадия включает преобразование дорожек из формата цифровой звукозаписи CD-DA (CD-Digital audio) к формату WAV. Есть программы, которые могут произвести MP3 непосредственно из аудио CD, но они достигают этого, все же выполняя извлечение аудио из компакт-диска как начальный шаг процесса. Программа читает дорожки аудиокомпакт-диска в цифровой форме и записывает их на жесткий диск как WAV-файлы. Четырехминутная дорожка (трек, фонограмма) занимает около 40—50 Мбайт формата WAV (расширение `.wav`), так что преобразование полного компакт-диска требует большого пространства на жестком диске.

Вторая стадия в процессе заключается в конвертировании `.wav`-файла в формат MP3. Этот шаг использует специализированное программное обеспечение, и программы, которые исполняют эту задачу, известны как кодеры MP3. MP3 файлы создают, используя разнообразие норм сжатия, разрешая пользователям выбрать оптимальное соединение количества и качества.

Системы воспроизведения звукового окружения. Средства воспроизведения звукового окружения начинались со стереозаписей и УКВ ЧМ-радио. Широко использовались магнитофоны и FM-стереотюнеры с высококачественным двухканальным звуком. Сначала использовались просто раздельные звуковые дорожки, затем технология Hi-Fi. Лазерные диски с самого начала выпускались с двухканальным стереозвуком высокого качества. Вскоре и большинство стандартов вещательного телевидения были адаптированы для передачи видео с двухканальным звуковым сопровождением.

Первыми на рынке появились простые декодеры Dolby Surround, которые позволяли на домашней аппаратуре выделить и прослушать третий пространственный канал — surround channel. Впоследствии был разработан более интеллектуальный декодер Dolby Surround Pro Logic, который выделял и центральный канал — center channel.

Кодер Dolby Surround не предназначен для передачи четырех независимых сигналов звука, каждый из которых надо прослушивать раздельно (например, звука одной ТВ программы на разных языках). В этом случае развязка между двумя любыми каналами должна была бы быть максимальной, а амплитуды и фазы сигналов могли бы быть совершенно не связаны между собой. Напротив, задача Dolby Surround — передать четыре канала звука (soundtrack), которые будут прослушиваться одновременно, и при этом воссоздавать в сознании слушателя пространственную звуковую картину (soundfield). Эта картина составляется из нескольких звуковых образов (sound images) — звуков, которые слушатель воспринимает связанными, например, со зрительными образами на экране. Звуковой образ характеризуется не только содержанием и мощностью звука, но и направлением в пространстве.

На входе кодера Dolby Surround присутствуют сигналы четырех каналов — L, C, R и S, а на выходах — два канала L (left total) и R (right total). Слово «total» (общий) означает, что каналы содержат не только «свой» сигнал (левый и правый), но и кодированные сигналы других каналов — C и S.

Кодирование реализуется простыми аналоговыми методами. Сигнал, кодированный в Dolby Surround, не содержит каких-либо управляющих сигналов или инструкций для декодера. По своим электрическим характеристикам он ничем не отличается от обычного двухканального сигнала стерео, и опознать кодированный сигнал простыми аппаратными методами (например, с помощью осциллографа или анализатора спектра) невозможно.

3.2. Технологии статических изображений

С 1980-х гг. бурно развивается технология обработки на компьютере графической информации. Компьютерная графика широко используется в компьютерном моделировании в научных исследованиях, компьютерных тренажерах, компьютерной анимации, деловой графике, играх и т. д.

3.2.1. Способы формирования изображения

Существуют два основных способа формирования изображения. Первый — путем нанесения на поверхность рисунка совокупности точек разного цвета, плотности, яркости (как это и происходит в цветной или черно-белой полиграфии), второй — путем вычерчивания и заштриховывания (графика или гравюра).

Оба этих подхода[1] сохранились и в компьютерную эру, только точечное изображение получило наименование растрового[2], а рисованное — векторного. Кроме того, компьютеризация сама предложила ряд новых подходов к графике, например фрактальный[3].

Рассмотрим основные способы и процессы создания и обработки изображений, которые в значительной степени определяются способом представления информации.

Битовое (точечное) представление. Двоичное изображение, для представления и хранения которого в цифровом виде используется битовая карта, где на каждый элемент изображения (пиксель) отводится 1 бит информации.

На рис. 3.1 приведен пример (двоичный, шестнадцатеричный, графический виды) бинарного изображения, записанного байтами, где 1 бит представляет 1 пиксель [Бинарное изображение. http://ru.wikipedia.org/wiki].

---

[1] В принципе можно выделить три способа отображения: точечное, линейное (штриховое), векторное. Причем линейное также можно рассматривать как частный случай векторного (когда задается начало и длина вектора, а направление фиксировано).

[2] Наименование «растровое» здесь скорее отражает не форму представления информации, а способ отображения — движение луча ЭЛТ при прорисовке точек на экране.

[3] Фрактал — это объект, отдельные элементы которого наследуют свойства родительских структур. Фракталы позволяют детально описывать целые классы изображений с расходованием относительно малого количества памяти, однако к изображениям вне этих классов фракталы плохо применимы.

Растровое представление. Нашло наиболее эффективное воплощение в полиграфических технологиях. В памяти фотонаборных машин с ЭЛТ (например, электронная фотонаборная машина Digiset 40T20) осуществлялась запись изображения с помощью микрорастра: знак представлен в памяти вертикальными точечно-растровыми линиями, высота которых ограничена верхним контуром знака и некоторой начальной (базовой) линией микрорастра (рис. 3.2). При этом двоичная информация о знаке задает длины белых и черных отрезков в каждой вертикальной линии микрорастра [*Самарин*].

Рис. 3.1. Пример бинарного изображения

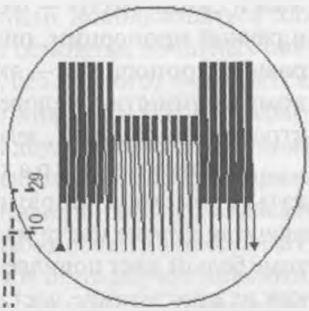


Рис. 3.2. Микрорастр буквы H (Digiset 40T)

Для воспроизведения знака большего размера управляющее устройство фотонаборной машины удлиняет вертикальный ход луча и одновременно (для получения необходимой ширины) увеличивает расстояние между линиями развертки.

Векторное представление. Основывается на применении простых геометрических фигур: точек, линий, многоугольников и т. п. Для создания рисунка объектам присваиваются параметры, это может быть толщина линий или форма заполняемых цветом фигур и т. п.. Изображение, как правило, должно храниться в виде набора координат, векторов, математических функций. Изображение в векторном формате предназначено для отображения специализированными устройствами — плоттерами, но может отображаться и при помощи растрового устройства, с использованием программных или аппаратных преобразователей.

3.2.2. Схемы цветообразования

Предметы, в том числе и их расцветку, человек видит потому, что они излучают свет, или потому, что они его отражают.

Соответственно эти диаметрально противоположные способы генерации цвета являются основной причиной искажения цветов на

экране и при печати. Для этих способов используются две противоположные системы описания цвета в компьютере: *аддитивная* и *субтрактивная*.

Аддитивные и субтрактивные цвета

Аддитивный цвет (от англ. add — добавлять, складывать) получается при соединении лучей света разных цветов. Отсутствие всех цветов представляет собой черный цвет, а присутствие всех цветов — белый. Система аддитивных цветов работает с излучаемым светом, например, от монитора компьютера.

В этой системе используются три основных цвета: красный, зеленый и синий (RGB — red, green, blue). Если их смешать друг с другом в равной пропорции, они образуют белый цвет, а при смешивании в разных пропорциях — любой другой. Система RGB адекватна цветовому восприятию человеческого глаза, рецепторы которого тоже настроены на красный, зеленый и синий цвета.

В системе субтрактивных цветов (от англ. subtract — вычитать) происходит обратный процесс: вы получаете какой-либо цвет, вычитая другие цвета из общего луча отраженного света. В этой системе белый цвет появляется в результате отсутствия всех цветов, тогда как их присутствие дает черный цвет. Система субтрактивных цветов работает с отраженным светом, например, от листа бумаги. Белая бумага отражает все цвета, окрашенная — некоторые поглощает, а остальные отражает.

В системе субтрактивных цветов основными являются голубой, пурпурный и желтый цвета (CMY), противоположные красному, зеленому и синему. Когда эти цвета смешиваются на белой бумаге в равной пропорции, получается черный цвет. Вернее, предполагается, что должен получиться черный цвет. В действительности типографские краски поглощают свет не полностью, и поэтому комбинация трех основных цветов выглядит темно-коричневой. Чтобы исправить возникающую неточность, для представления тонов черного цвета принтеры добавляют немного черной краски. Систему цветов, основанную на таком процессе четырехцветной печати, принято обозначать аббревиатурой CMYK (**c**yan, **m**agenta, **y**ellow, black).

Цветовые модели

Цветовая модель RGB. Монитор компьютера создает цвет непосредственно излучением света и использует, таким образом, систему

цветов RGB. Поверхность монитора состоит из мельчайших точек (пикселей) красного, зеленого и синего цветов, форма точек варьируется в зависимости от типа электронно-лучевой трубки (ЭЛТ). Пушка ЭЛТ подает сигнал различной мощности на экранные пиксели. Каждая точка имеет один из трех цветов, при попадании на нее луча из пушки она окрашивается в определенный оттенок своего цвета в зависимости от силы сигнала. Поскольку точки маленькие, уже с небольшого расстояния они визуально смешиваются друг с другом и перестают быть различимы. Комбинируя различные значения основных цветов, можно создать любой оттенок из более 16 млн цветов, доступных в RGB.

Цветовая модель CMYK. Система цветов CMYK была широко известна задолго до того, как компьютеры стали использоваться для создания графических изображений. Триада основных печатных цветов — голубой, пурпурный и желтый (CMY, без черного) — является, по сути, наследником трех основных цветов живописи (синего, красного и желтого). Изменение оттенка первых двух связано с отличным от художественных химическим составом печатных красок, но принцип смешения тот же самый. И художественные, и печатные краски, несмотря на провозглашаемую самодостаточность, не могут дать очень многих оттенков. Поэтому художники используют дополнительные краски на основе чистых пигментов, а печатники добавляют, как минимум, черную краску.

Все представления изображений, предназначенные для вывода в типографии, должны быть конвертированы в CMYK. Этот процесс называется цветоделением.

Цветовые модели HSB и HSL. Системы цветов RGB и CMYK являются следствиями ограничений, накладываемых аппаратным обеспечением (мониторами и сканерами в случае с RGB и типографскими красками в случае с CMYK). Более логичным способом описания цвета является представление его в виде *тона*, *насыщенности* и *яркости* — система HSB. Она же известна как система HSL (*тон, насыщенность, освещенность*).

Тон представляет собой конкретный оттенок цвета на цветовом круге, отличный от других: красный, зеленый, голубой и т. п. Насыщенность цвета характеризует его относительную интенсивность (или чистоту). Уменьшая насыщенность, например красного, мы делаем его более пастельным, приближаем к серому. Яркость (или освещенность) цвета показывает величину затемнения или осветления исходного оттенка.

HSB имеет перед другими системами важное преимущество: она больше соответствует природе цвета, хорошо согласуется с моделью восприятия цвета человеком. Многие оттенки можно быстро и удобно получить в HSB, конвертировав затем в RGB или CMYK.

Цветовая модель Grayscale. Цветовая модель Grayscale («градации серого», полутоновая) представляет собой ту же индексированную палитру, где вместо цвета пикселям назначена одна из 256 градаций серого.

3.2.3. Основные параметры и устройства формирования изображения

Оптическое разрешение

Оптическое разрешение измеряется в пикселях на дюйм (ppi — pixels per inch), иногда dpi — точки на дюйм, однако понятие точка означает элемент, не имеющий конкретной формы. Например, сканеры и растровые графические файлы оперируют пикселями, имеющими форму квадрата.

Сканеры. Оптическое разрешение показывает, сколько пикселей сканер может считать на квадратный дюйм. Его значение записывается так: 300×300, 300×600, 600×1200 и т. п. Первое число говорит о количестве считывающих информацию датчиков, именно на него стоит обращать внимание, хотя часто производители и продавцы любят указывать в качестве разрешения что-нибудь вроде 4500 dpi. Это интерполированное разрешение, которое является свойством не сканера, а поддерживающей его программы.

Интерполяция — способ увеличения (уменьшения) размера изображения (разрешение) файла посредством программы. При уменьшении данные отбрасываются, при увеличении — программа их вычисляет. Таким образом, сильно увеличенные картинки выглядят размытыми или зубчатыми (в зависимости от способа интерполяции).

Известны три основных способа интерполяции:

- Nearest Neighbor — для добавляемого пикселя берется значение соседнего с ним;
- Bilinear — выбирается среднее цветовое значение пикселей с каждой стороны от создаваемого;
- Bicubic — усредняется значение группы не только непосредственно граничащих, но и всех соседних пикселей. Какой именно диапазон пикселей выбирается для усреднения и по какому ал-

горитму это усреднение происходит — этим отличаются способы бикубической интерполяции в разных программах.

Цифровые камеры. Качество цифровой камеры зависит от нескольких факторов, включая оптическое качество линзы, матрицу съемки изображения, алгоритмы сжатия и другие компоненты. Однако самый важный детерминант качества изображения — разрешающая способность матрицы ПЗС: чем больше элементов, тем выше разрешающая способность и, таким образом, больше подробностей может быть зафиксировано.

Однако даже датчики 16 мегапикселей все еще не перекрывают возможностей обычной фотопленки. Поскольку высококачественные линзы объективов обеспечивают разрешение по крайней мере 200 точек на 1 мм, негативная пленка стандарта 100ASA с размером кадра 24×36 мм обеспечит разрешение 24×200×36×200 = 34,56 млн пикселей, что все еще недостижимо для цифровых камер.

Разрядная глубина

Разрядная (битовая, цветовая) глубина характеризует количество информации, содержащейся в одном пикселе выходного образа. Битовую глубину изображения часто называют цветовой разрешающей способностью. Она измеряется в битах на пиксель (bit per pixel, bpp). Так, если речь идет об иллюстрации, имеющей в каждом пикселе по 8 бит цветовой информации, то ее цветовая разрешающая способность будет 8 bpp, что дает $2^8 = 256$ доступных для 8-битового изображения цветов.

Самый простой сканер (черно-белый) использует для представления каждого пикселя 1 бит. Чтобы воспроизвести полутона между черным и белым, сканер должен иметь хотя бы 4 бита для 16 (2^4) полутонов на каждый пиксель.

Современные цветовые сканеры поддерживают не менее 24 бит, что означает использование 8 бит для каждого из первичных цветов (красный, синий, зеленый). Устройство на 24 бита может теоретически фиксировать более чем 16 млн различных цветов, хотя практически это число намного меньше. Это почти фотографическое качество, и упоминается поэтому обычно как «полноцветное» сканирование («true colour» scanning). Однобитовые изображения, называемые Bitmap, или, иногда, Lineart, используются и сегодня там, где не требуются цвето-тоновые переходы. Равный по размеру Bitmap-файл в 24 раза меньше, чем файл RGB, и кроме того, очень хорошо сжимается.

Динамический диапазон

Динамический диапазон по своей сути подобен разрядной глубине, которая описывает цветовой диапазон сканера, и определяется как функционированием АЦП сканера, так и чистотой света, качеством цветных фильтров и уровнем помех в системе.

Динамический диапазон измеряется в шкале от 0,0 (абсолютно белый) до 4,0 (абсолютно черный), и единственное число, данное для конкретного сканера, говорит, сколько оттенков модуль может различить. Большинство цветных планшетных сканеров с трудом воспринимает тонкие различия между темными и светлыми цветами на обоих концах диапазона и имеет динамический диапазон около 2,4. Наиболее близко к пределу динамического диапазона позволяют подойти барабанные сканеры, часто обеспечивающие значения от 3,0 до 3,8.

Теоретически сканер на 24 бита предлагает диапазон 8 бит (256 уровней) для каждого первичного цвета. Различие между парой из 256 уровней в этом случае обычно не воспринимается человеческим глазом, но, к сожалению, наименьшие из значащих битов теряются в шуме, в то время как любые тональные исправления после сканирования еще более сужают диапазон. Именно поэтому лучше всего предварительно устанавливать любые исправления яркости и цвета на уровне драйвера сканера перед заключительным сканированием. Более дорогие сканеры с глубиной в 30 или 36 битов имеют намного более широкий диапазон, предлагая более детализированные оттенки, и разрешают пользователю делать тональные исправления, заканчивающиеся 24-битовым изображением. Сканер на 30 битов принимает 10 битов данных на каждый цвет, в то время как сканеры на 36 битов — по 12 битов. Драйвер сканера позволяет пользователю выбрать, какие именно 24 бита из исходных 30 или 36 битов сохранить, а какие — нет. Эта настройка делается путем изменения «кривой цветовой гаммы» (Gamma Curve) и доступна при обращении к Настройке тонов (Tonal Adjustment control) драйвера TWAIN.

Режимы сканирования

Среди разнообразия методов представления изображений в ЭВМ наиболее распространенными являются:

- штриховая графика (line art);
- полутоновое изображение (greyscale);
- цветное изображение (colour).

Штриховая графика — наиболее простой формат. Так как сохраняется только черно-белая информация (в компьютере представлен черный цвет как «1» и белый как «0»), требуется только 1 бит данных, чтобы сохранить каждую точку сканированного изображения. Штриховая графика наиболее подходит при сканировании чертежей или текста.

Полутоновое изображение. В то время как компьютеры могут сохранять и выдавать изображения в полутонах, большинство принтеров не способно печатать различные оттенки серых цветов. Они применяют метод, названный *обработкой полутонов*, используя точечный растр, имитирующий полутоновую информацию.

Изображения в оттенках серого — наиболее простой метод сохранения графики в компьютере. Человек может различить не более 255 различных оттенков серого, что требует единственного байта данных со значением от 0 до 255. Данный тип изображения составляет эквивалент черно-белой фотографии.

Полноцветные изображения — наиболее объемные и самые сложные, сохраняемые и обрабатываемые в ПК, используют 24 бита (по 8 на каждый из основных цветов), чтобы представить полный цветовой спектр.

3.2.4. Программные средства обработки изображений

Драйвер TWAIN. Стандарт TWAIN (Toolkit Without AnInteresting Name) обеспечивает взаимодействие сканеров практически с любым прикладным ПО — пакетами обработки изображений (наподобие Adobe PhotoShop), настольными издательскими системами или программами распознавания символов. Этот стандарт совместно разработан Hewlett-Packard, Kodak, Aldus, Logitech и Caere и определяет, каким образом устройства получения изображений (сканеры, цифровые камеры и др.) передают данные прикладным программам. Стандарт TWAIN позволяет приложениям работать с устройствами получения изображений, «не зная» что-либо об устройстве непосредственно.

Цветовая калибровка. Одна из особенностей использования настольного сканера — отсканированное изображение может выглядеть по-разному на экране и в отпечатанной форме, и все это будет отличаться и от оригинала. Решение этой проблемы — система цветовой калибровки (или установка соответствия цветов). Сложности человеческого восприятия цветов сделали калибровку цветов

большой проблемой, вследствие чего есть несколько различных подходов, как разработанных, так и перспективных.

Одна из самых полных систем — система управления цветом, разработанная Kodak (Colour Management System — CMS), которая использует различные цветовые профили, соответствующие каждому устройству — сканеру, монитору, принтеру в системе, чтобы передавать и стандартизировать цвета.

Другие системы были разработаны изготовителями сканеров и прикладными программистами. Эти системы также базируются на цветовых профилях различных устройств, которые будут использоваться для сканирования, редактирования и вывода заключительного изображения. В таких системах используется *исправление на основе вывода*, при этом сканируется и выводится стандартно калиброванное эталонное изображение, и затем вносятся изменения в цветовые профили, чтобы стандартизировать цвета.

Фоторедактирование (ретуширование). Когда сканер осуществляет фиксацию цветного изображения, это часто только начало технологического процесса. Будучи однажды оцифрованной, фотография может быть представлена в разных видах и комбинироваться с другой информацией в растровом редакторе или пакете раскрашивания.

В подготовке печатного издания очень редко используются «сырые» изображения — черты моделей «очищаются»: сглаживание морщин, окраска глаз, «причесывание» волос и пр. Обычно фоторетушер пытается сделать одну из двух вещей — это или замена некоторых элементов изображения (например, изменение цвета чьих-то волос), или сотворение чего-то нереального и фантастического. В любом случае, вмешательство ретушера не должно обнаруживаться визуально.

3.2.5. Форматы графических файлов

Сжатие данных (data compression) позволяет резко уменьшить объем памяти, необходимой для хранения данных, сократить время их передачи. Сжатие данных может осуществляться как программным, так и аппаратным или комбинированным методом.

Сжатие текстов основано на выборе более компактного расположения байтов, кодирующих символы или их последовательности. Определенные результаты дает статистическое кодирование, в котором наиболее часто встречающиеся символы имеют коды наименьшей

длины. Здесь также используется счетчик повторений пробелов. Что же касается звука и изображений, то объем представляющей их информации зависит от выбранного шага квантования и числа разрядов аналого-цифрового преобразования. В принципе, здесь используются те же методы сжатия, что и при обработке текстов. Но если сжатие текстов происходит без потери информации, то сжатие звука и изображения почти всегда приводит к ее некоторой потере.

Размер файла, в котором сохраняется изображение, существенно зависит от формата файла, а это важная характеристика технологии, поскольку высокие разрешающие способности, поддерживающиеся многими современными сканерами, могут привести к созданию файлов размером более 30 Мбайт для страницы формата А4.

Методы сжатия графики

Метод сжатия RLE (Run Length Enconding, кодирование длины серий). При сжатии этим методом последовательность повторяющихся величин (например, набор бит для представления пикселя) заменяется парой — повторяющейся величиной и числом ее повторений.

Программа сжатия файла может сначала записывать количество видеопикселей, а затем их цвет, или наоборот. Поэтому возможна такая ситуация, когда программа, считывающая файл, ожидает появления данных в ином порядке, чем программа, сохраняющая этот файл на диске.

Сжатие методом RLE наиболее эффективно для изображений, которые содержат большие области однотонной закраски, так как в них есть длинные последовательности одинаковых видеопикселей.

Метод сжатия LZW (назван так по первым буквам его разработчиков Lempel, Ziv, Welch) основан на поиске повторяющихся узоров в изображении. Сильно насыщенные узорами рисунки могут сжиматься до 0,1 их первоначального размера.

Метод сжатия JPEG. Формат разработан объединенной группой экспертов по фотографии (Joint Photographic Experts Group). Сжатие по JPEG сильно уменьшает размер файла с растровым рисунком (возможен коэффициент сжатия 100 : 1). Высокий коэффициент сжатия достигается за счет сжатия с потерями. Метод JPEG использует тот факт, что в то время как человеческий глаз чувствителен к изменению яркости, изменения цвета он замечает хуже. Поэтому при сжатии этим методом запоминается больше информации о разнице между яркостями пикселей и меньше — о разнице между их цветами.

Уровень сжатия (степень потери данных) может изменяться, но даже при задании максимального качества JPEG теряет некоторые подробности.

Растровые форматы

BMP (BitMaP — точечный рисунок) — основной формат растровой графики в ОС Windows.

В файлах BMP информация о цвете каждого пикселя кодируется 1, 4, 8, 16 или 24 битами (бит/пиксель). Числом бит/пиксель, называемым также цветовой глубиной, определяется максимальное число цветов в изображении.

Файл разбит на четыре основных раздела — заголовок файла растровой графики, информационный заголовок растрового массива, таблица цветов и собственно данные растрового массива. В информационном заголовке растрового массива содержатся сведения об изображении, хранящемся в файле (например, высоте и ширине в пикселях), в том числе адрес, с которого начинается область данных растрового массива. В таблице цветов представлены значения основных цветов RGB (красный, зеленый, синий) для используемых в изображении цветов.

Формат собственно данных растрового массива в файле BMP зависит от числа бит, используемых для кодирования данных о цвете каждого пикселя. При 256-цветном изображении каждый пиксель в той части файла, где содержатся собственно данные растрового массива, описывается одним байтом (8 бит). Это описание пикселя не представляет значений цветов RGB, а служит указателем для входа в таблицу цветов файла. Значения пикселей хранятся в порядке их расположения слева направо, начиная (как правило) с нижней строки изображения. Таким образом, в 256-цветном BMP-файле первый байт данных растрового массива представляет собой индекс для цвета пикселя, находящегося в нижнем левом углу изображения; второй байт представляет индекс для цвета соседнего справа пикселя и т. д. Файлы BMP с глубиной 16 и 24 бит/пиксель не имеют таблиц цветов; в этих файлах значения пикселей растрового массива непосредственно характеризуют значения цветов RGB.

PCX — первый стандартный формат файлов для растровой графики в компьютерах систем IBM PC. Файлы PCX включают три части — 128-байтовый заголовок PCX, данные растрового массива, таблицу цветов (не обязательная часть). Заголовок содержит несколько полей

данных, в том числе о размере изображения и количестве бит для кодирования цветовой информации каждого пикселя. Информация растрового массива сжимается с использованием метода RLE; факультативная таблица цветов в конце файла содержит 256 значений цветов RGB, определяющих цвета изображения. Кодирование цвета каждого пикселя в современных изображениях PCX может производиться с глубиной 1,4,8 или 24 бит.

TIFF (Tagged Image File Format — формат файлов изображения, снабженный тегами). Если PCX — один из самых простых для декодирования форматов растровой графики, то TIFF — один из самых сложных. Каждый файл начинается 8-байтовым заголовком файла изображения (IFH), важнейший элемент которого — каталог файла изображения (Image File Directory — IFD) — служит указателем к структуре данных. IFD представляет собой таблицу для идентификации одной или нескольких порций данных переменной длины, называемых тегами, хранящих информацию об изображении. В спецификации формата файлов TIFF определено более 70 различных типов тегов. Например, тег, хранящий информацию о ширине изображения в пикселях, или о его высоте, или таблицу цветов (при необходимости), или сами данные растрового массива. Формат файла легко расширяется, поскольку для придания файлу дополнительных свойств достаточно определить дополнительные типы тегов.

Данные растрового массива в файле TIFF могут сжиматься с использованием нескольких методов.

GIF (Graphics Interchange Format) — формат обмена графическими данными) разработан компанией CompuServe. Структура файла зависит от версии GIF-спецификации (распространены две версии — GIF87a и GIF89a). Независимо от номера версии файл GIF начинается с 13-байтового заголовка, содержащего сигнатуру, которая идентифицирует этот файл в качестве GIF-файла, номер версии GIF и другую информацию. Если файл хранит только одно изображение, вслед за заголовком обычно располагается общая таблица цветов, определяющая палитру изображения. Если в файле хранится несколько изображений, то вместо общей таблицы цветов каждое изображение сопровождается локальной таблицей цветов.

Основные достоинства GIF заключаются в широком распространении этого формата и в его компактности. Но ему присущи два достаточно серьезных недостатка. Один из них состоит в том, что в изображениях, хранящихся в виде GIF-файла, не может быть использовано более 256 цветов. Второй, возможно, еще более серьезный, заключает-

ся в том, что разработчики программ, использующие в них форматы GIF, должны иметь лицензионное соглашение с CompuServe и вносить плату за каждый экземпляр программы.

PNG (Portable Network Graphic — переносимый сетевой формат) был разработан для замены GIF. PNG унаследовал многие возможности GIF и, кроме того, позволяет хранить изображения с истинными цветами. Еще более важно, что он сжимает информацию растрового массива в соответствии с вариантом пользующегося высокой репутацией алгоритма сжатия LZ77 (предшественника LZW), которым можно пользоваться бесплатно.

JPEG разработан компанией C-Cube Microsystems как эффективный метод хранения изображений с большой глубиной цвета, например, получаемых при сканировании фотографий с многочисленными едва уловимыми оттенками цвета. Используется алгоритм JPEG-сжатия с потерями информации.

Векторные форматы

Файлы векторного формата содержат описания рисунков в системе команд для построения простейших графических объектов (линий, окружностей, прямоугольников, дуг и т. д.). Кроме того, в этих файлах хранится некоторая дополнительная информация. Различные векторные форматы отличаются набором команд и способом их кодирования.

WMF (Windows Metafile) — формат, доступный большинству приложений Windows, так или иначе связанных с векторной графикой, служит для передачи векторов через буфер обмена (Clipboard). Однако может искажать цвет, не сохранять ряд параметров, которые могут быть присвоены объектам в различных векторных редакторах, не воспринимается программами Macintosh.

EPS (Encapsulated PostScript) — упрощенный PostScript, может использоваться большинством настольных издательских систем и векторных программ, некоторыми растровыми программами. Однако не может содержать в одном файле более одной страницы, не сохраняет ряд установок для принтера. Как и в файлы печати PostScript, в EPS записывают конечный вариант работы, хотя такие программы, как Adobe Illustrator, Photoshop и Macromedia FreeHand, могут использовать его как рабочий.

DXF (Drawing Interchange Format) используется всеми программами САПР, многими векторными редакторами, некоторыми издательскими системами.

CGM (Computer Graphics Metafile) используется в программах редактирования векторных рисунков, САПР и издательских системах.

SVG (Scalable Vector Graphics) — расширение языка XML, предназначенное для того, чтобы описать двумерную векторную графику как статическую, так и анимированную. SVG допускает три типа графических объектов: 1) векторные графические формы (например, контуры, состоящие из прямых и кривых линий и областей, ограниченных ими); 2) растровая графика, представляющая оцифрованные образы; 3) текст.

3.3. Цифровое видео

В телевизионной системе PAL[1] (Phase-Alternation-Line, чередование строк) каждый законченный кадр заполняется построчно, сверху донизу. В Европе используется переменный электрический ток с частотой 50 Гц, и система PAL связана с этим — здесь выполняется 50 проходов экрана каждую секунду. Требуется два прохода, чтобы нарисовать полный кадр, так что частота кадров равна 25 кадров/с. Нечетные строки выводятся при первом проходе, четные — на втором. Этот метод называется чересстрочная развертка (interlaced), в противоположность чему изображение на компьютерном мониторе, создаваемое за один проход, известен как без чередования строк (progressive).

Компьютеры, наоборот, имеют дело с информацией в цифровой форме. Чтобы хранить визуальную информацию в цифровой форме, аналоговый видеосигнал должен быть переведен в цифровой эквивалент с использованием аналого-цифрового преобразователя, иначе — осуществить оцифровку, или видеозахват. Так как компьютеры имеют дело с цифровой графической информацией, никакая другая специальная обработка данных не требуется, чтобы в дальнейшем выводить это цифровое видео на компьютерный монитор. Однако чтобы отобразить цифровое видео на обычном телевизо-

[1] Известны три стандарта кодирования сигнала телевидения:
система PAL (использует большинство стран Европы);
система SECAM (используют Франция, Россия и некоторые восточноевропейские страны). Отличается от системы PAL только в тонкостях, однако этого достаточно, чтобы они были несовместимыми;
система NTSC (используют США и Япония).

ре, нужен обратный конвертер — цифро-аналоговый (DAC — или ЦАП), который должен преобразовать двоичную информацию в аналоговый сигнал.

3.3.1. Цифровые видеокамеры

Основная составляющая цифровой видеокамеры — светочувствительная матрица (прибор с зарядовой связью — ПЗС) собирает и обрабатывает свет, приходящий от объектива, и преобразует его в электрический сигнал. В то время как видеокамеры среднего качества оборудованы единственной ПЗС, модели более высокого класса используют три матрицы. В этом случае объектив содержит призму, которая расщепляет поступающий свет на три основных цвета, причем каждый поступает на отдельную матрицу. Результатом являются высококачественная цветопередача и качество изображения, заметно лучшие, чем для моделей с единственной ПЗС, хотя и при существенной дополнительной стоимости.

Число пикселей, которые составляют матрицу, может изменяться от одной модели к другой, однако большее число пикселей не обязательно означает лучшее качество изображения.

Цифровые камеры обеспечивают цифровую или оптическую стабилизацию изображения, чтобы уменьшить колебание, которое неизбежно сопровождает ручную съемку. Цифровая стабилизация изображения (Digital image stabilisation — DIS) очень эффективна, но имеет тенденцию уменьшать разрешение картины, поскольку активно используется для записи образа меньший процент датчиков (остальные заняты стабилизационной обработкой). Оптическая стабилизация изображения (Optical image stabilisation — OIS) использует призму, которая компенсирует колебания регулировкой пути светового луча, проходящего через систему линз камеры. Оба метода достигают примерно одной и той же степени видимой стабильности, но OIS, возможно, лучше, так как не уменьшает разрешение.

3.3.2. Форматы цифрового видео

VCD. Формат VideoCD был создан, чтобы обеспечить диалоговую среду, которая была бы недорога для копирования, поддерживала полный экран и видео полного движения и функционировала бы в

широком диапазоне различных платформ ПЭВМ, телевидения, игровых приставок и мультимедийного оборудования.

В середине 1993 г. Philips, Sony, Matsushita и JVC согласовали спецификации VideoCD, позже получившие известность как «Белая Книга». Стандарт использует определения, описанные в стандартах «красной» (CD-DA) и «желтой» (CD-ROM) книг и вводит дополнительную гибкость, чтобы учесть защиту авторского права, вставки библиографической информации, абстрактных данных, компьютерных программ, обеспечить диалоговый контроль в течение воспроизведения.

Запись первой дорожки диска (Track 1), которая содержит файловую структуру ISO 9660 и информационную область, сделана в соответствии с CD-ROM XA Mode 2. Файловая структура может также включать расширения Joliet, чтобы поддерживать длинные имена файлов Windows.

VCD 1.1 поддерживает технологию выбираемых треков, а начиная с версии VCD 2.0 (1995 г.) поддерживается полная интерактивность через дистанционное управление. VCD 2.0 позволяет организовать до 98 треков, каждый из которых может быть индексирован в 99 сценах. Каждый трек может содержать и воспроизводить сцены, которые могут быть видео, звуковыми или фотоизображениями. В основном этот формат можно трактовать как Audio CD с дополнением видео- или фото-фрагментами и средствами навигации по содержанию.

SVCD. Выпущенный в 1998 г. консорциумом, который включал Philips, Sony, Matsushita и JVC, формат «VCD высшего качества» (SuperVCD) — впоследствии стандартизированный как ISO IEC 62107 — является естественным развитием стандарта VCD. Основное различие в том, что для видеопотока используется кодирование MPEG-2 (вместо MPEG-1), которое обеспечивает более высокое разрешение и скорость и переменную скорость видеопотока, а также поддерживаются субтитры. Как следствие, SVCD способен к показу в два раза более четких видеоизображений, чем его предшественник, за счет уменьшенной вместимости (35 и 80 мин на диск в зависимости от средней используемой битовой скорости).

Видеопоток SVCD может содержать до четырех независимых каналов субтитров для различных языков, которые накладываются на видеоизображения в процессе воспроизведения и могут подключаться или удаляться по желанию. Так как субтитры сохранены как битовая графика, они не привязаны к какому-то специфическому набору символов. SVCD поддерживает также гиперсвязи (типа HTML), по-

зволяет подключать фотографии, автоматическое проигрывание слайдов и музыкальных фрагментов, поддерживает многоуровневые иерархические меню и оглавления (индексацию).

XVCD и **XSVCD** (eXtended VCD и eXtended SVCD) являются неофициальными вариантами стандартов VCD и SVCD, предназначенными для достижения улучшенного качества изображения, например, увеличивая битовую скорость (битрейт) в соответствии с более высокой способностью передачи данных современными накопителями CD-ROM.

DivX. Формат DivX базируется на видеотехнологии MPEG-4 с дополнением звукового потока MP3. Поскольку сжатый в формате DivX кинофильм составляет от 10 до 20 % от размера оригинала DVD (обычно 5 Гбайт), 80—90-минутное DVD-кино занимает приблизительно 650 Мбайт в разрешении 640×480.

DV. Технически формат DV использует дискретное косинус-преобразование, осуществляемое в три стадии. Первая стадия использует DCT-сжатие, удаляющее информацию, которая не может быть замечена человеческим глазом. При этом в каждом пикселе отделяют цветовую и яркостную информацию, что сокращает данные на одну треть. Затем сигнал RGB преобразуется в YUV — Y для яркости, а U и V для цвета, по формуле YUV 4:2:2. Затем цифровой видеокодек оптимизирует формулу к YUV 4:2:0, связывая цветовую информацию от смежных пикселей в блоки 4×4. Далее система аппаратного сжатия, размещенная на камере, сжимает видео с использование алгоритма, подобного M-JPEG.

Система DV отличается способностью записи различных частей каждого кадра с различной степенью сжатия. Так, синее небо в фоне изображения может быть сжато, скажем, к 25:1, в то время как лес на переднем плане, который нуждается в большем количестве деталей, только до 7:1. Этим способом цифровое видео может оптимизировать видеоструктуру потока кадров, в то время как M-JPEG должен иметь установленную норму сжатия для видео в целом и не может разумно регулировать сжатие каждого изображения. Кроме того, также используется метод, известный как адаптивное межстрочное сжатие, которое заключается в том, что перекрывающиеся строки кадра (как в PAL, например) соединяются в одну, если различие между ними невелико. В теории это означает, что сцены с меньшим количеством движения обрабатываются лучше, чем быстрые сцены.

Mini-DV (мини-цифровое видео). Главное преимущество формата MiniDV состоит в том, что лента, являющаяся 1/12 от размера стан-

дартной пленки VHS, позволяет сделать запись 1 часа в формате SP или до 90 мин более низкого качества выхода в «долгоиграющем режиме» (long play, LP) при горизонтальном разрешении до 500 линий.

Digital8. Введенный в начале 1999 г., формат видеокамеры Sony Digital8 может рассматриваться как шаг между 8 мм или Hi-8 и MiniDV. Запись здесь производится почти в том же самом качестве, как для MiniDV, но на ленты 8 мм и Hi-8, которые имеют размер $^1/_4$ размера VHS и вместимость до 1 часа.

MICROMV. В 2001 г. Sony объявила ряд цифровых видеокамер MICROMV, использующих формат сжатия MPEG-2 при записи сигналов качества DV на ленты, размер которых составляет 70 % от кассет MiniDV. При скорости в 12 Мбит/с ультракомпактный формат MICROMV имеет битовую скорость вполовину меньшую, чем для miniDV, что делает редактирование видео на ПЭВМ намного менее ресурсопоглощающей задачей.

Региональное кодирование

Поскольку обычно выход фильма на экраны не является одновременным (фильм может выйти на видео в США, когда только выходит на экраны в Европе), киностудии хотят контролировать выпуск видеокопий в различных странах. Поэтому потребовалось, чтобы стандарт DVD включал коды, которые могут предотвратить воспроизведение некоторых дисков в определенных географических областях (регионах). Каждый видеопроигрыватель получает код для региона, в котором он продан. Это означает, что диски, купленные в одной стране, не могут считываться на плеерах, купленных в другой стране.

Региональные коды являются дополнительными для изготовителя диска, и отсутствие кода означает отсутствие региональных ограничений. Это не система кодирования, а только информационный байт, обозначающий восемь различных регионов, который проверяется при проигрывании диска.

Видеоредактирование

Известны два типа видеоредактирования. Первый заключается в редактировании при переписывании одной ленты на другую и называется *линейным редактированием*. Второй требует, чтобы редактируемые видеопоследовательности были вначале помещены на жесткий диск, затем отредактированы и возвращены на пленку. Этот метод известен как *нелинейное редактирование* (НЛР, NLE). Для нелинейного

редактирования видеопередачи карты захвата переводят видео в цифровую форму, и при этом функция редактирования выполняется полностью на ПЭВМ, почти так же, как редактируется документ в текстовом редакторе.

Стандарт MPEG-2

Рассмотрим в качестве примера стандарт MPEG-2, который состоит из трех основных частей: системной, видео и звуковой.

Системная часть описывает форматы кодирования для мультиплексирования звуковой, видео- и другой информации, рассматривает вопросы комбинирования одного или более потоков данных в один или множество потоков, пригодных для хранения или передачи.

Системное кодирование в соответствии с синтаксическими и семантическими правилами, налагаемыми данным стандартом, обеспечивает необходимую и достаточную информацию, чтобы синхронизировать декодирование без переполнения или «недополнения» буферов декодера при различных условиях приема или восстановления потоков.

Таким образом, системный уровень выполняет пять основных функций:

- синхронизацию нескольких сжатых потоков при воспроизведении;
- объединение нескольких сжатых потоков в единый поток;
- инициализацию для начала воспроизведения;
- обслуживание буфера;
- определение временной шкалы.

Видеочасть стандарта описывает кодированный битовый поток для высококачественного цифрового видео. MPEG-2 является совместимым расширением MPEG-1, он поддерживает чересстрочный видеоформат и содержит средства для поддержки телевидения высокой четкости.

Стандарт MPEG-2 определяется в терминах расширяемых профилей, каждый из которых, являясь частным случаем стандарта, имеет черты, необходимые всем классам приложений. Иерархические масштабируемые профили могут поддерживать такие приложения, как совместимое наземное многопрограммное ТВ (ТВЧ), пакетные сетевые видеосистемы, обратную совместимость с другими стандартами (MPEG-1 и H.261) и приложениями, использующими многоуровневое кодирование.

Звуковая часть стандарта MPEG-2 определяет кодирование многоканального звука. MPEG-2 поддерживает до пяти полных широкополосных каналов плюс дополнительный низкочастотный канал и (или) до семи многоязычных комментаторских каналов.

Стандарт MPEG-2 не регламентирует методы сжатия видеосигнала, а только определяет, как должен выглядеть битовый поток кодированного видеосигнала, поэтому конкретные алгоритмы являются коммерческой тайной фирм — производителей оборудования. Однако существуют общие принципы, и процесс сжатия цифрового видеосигнала может быть разбит на ряд последовательных операций:

- преобразование аналогового сигнала в цифровую форму;
- предварительная обработка;
- дискретное косинусное преобразование;
- квантование;
- кодирование.

После аналого-цифрового преобразователя (АЦП) производится предварительная обработка сигнала, которая включает в себя следующие преобразования.

1. Удаление избыточной информации. Например, если фон изображения состоит из идентичных символов (пикселей), то совершенно не обязательно их все передавать. Достаточно описать один пиксель и послать его с сообщением о том, как часто и где он повторяется в изображении.

2. Если исходное изображение передается в виде чересстрочных полей, то они преобразуются в кадры с прогрессивной разверткой.

3. Сигналы цветности (RGB) преобразуются в цветоразностные сигналы U и V и сигнал яркости Y.

4. Изображение достраивается до кратного 16 количества пикселей по строкам и столбцам, чтобы обеспечить разбиение изображения на целое число макроблоков.

5. Производится преобразование из формата цветности 4:4:4 в формат 4:2:2 (горизонтальная передискретизация цветоразностных компонентов) или 4:2:0 (горизонтальная и вертикальная передискретизация цветоразностных компонентов).

Квантование. Изображение разбивается на последовательность макроблоков, каждый из которых состоит из шести блоков по 8×8 пикселей: четыре образуют матрицу 16×16 и несут информацию о яркости; по одному — определяют цветоразностные компоненты U и V, которые соответствуют области изображения, покрываемой матрицей 16×16 пикселей.

Производится разбиение потока кадров изображения по типам для них находятся векторы движения, которые необходимы для повышения предсказуемости величин элементов изображения. Векторы движения обеспечивают компенсацию перемещений в прошедших и последующих кадрах. Компенсация движения применяется при предсказании текущего кадра на основе предыдущих и интерполяционного предсказания на основе прошедших и последующих изображений. Векторы движения определяются для каждой зоны изображения с размерами 16×16 пикселей, т. е. для макроблоков. В большинстве случаев видеопоследовательности содержат избыточность в двух направлениях — временном и пространственном. Главное статистическое свойство, на котором основано сжатие, — межэлементная корреляция, включающая предположение о коррелированности последовательных кадров видеоданных. Таким образом, значения отдельных пикселей изображения могут быть предсказаны либо по значениям ближайших пикселей внутри одного кадра (внутрикадровое кодирование), либо по значениям пикселей, расположенных в ближайших кадрах (межкадровое кодирование и компенсация перемещения).

Кодирование. В некоторых случаях, например, при смене видеосцены в видеопоследовательности, временная корреляция между ближайшими кадрами очень низка. В таких случаях решающую роль в достижении эффективного сжатия видеоинформации играет внутрикадровая корреляция, т. е. пространственная корреляция пикселей изображения. Однако если корреляция между последовательными кадрами видеоданных высока, то в случае, когда два последовательных кадра имеют схожее или одинаковое содержание, желательно применение межкадровой корреляции пикселей с временным предсказанием. На практике для достижения высокого коэффициента сжатия видеоинформации используется комбинация из двух подходов.

Стандарт MPEG-2 определяет три типа кадров, для каждого из которых предусмотрен свой вид кодирования:

- опорные кадры, так называемые I-кадры (Intra Frames), которые являются основными и кодируются без обращения к другим кадрам, т. е. с использованием информации только этого кадра. Вид кодирования — внутрикадровый, обеспечивающий умеренное сжатие. Все остальные кадры анализируются процессором, который сравнивает их с опорными, а также между собой;

- P-кадры (Predicted) — закодированные относительно предыдущих I- или P-кадров. Кодирование P-кадров выполняют с использованием алгоритмов компенсации движения и предсказанием «вперед» по предшествующим I- и P-кадрам. Они сжаты в три раза сильнее, чем I-кадры, и служат опорными для последующих P- и B-кадров. Компенсация движения, применяемая к макроблокам P-кадров, вырабатывает два вида информации: векторы движения (разница между базовыми и кодированными макроблоками) и значения ошибок (разница между предсказанными величинами и действительными результатами). Если макроблок в P-кадре не может быть описан с использованием компенсации движения, что случается при появлении некоторого неизвестного объекта, то он кодируется тем же способом, что и макроблок в I-кадре;
- B-кадры (Bidirectionally Predicted) — закодированные относительно предыдущих и последующих кадров, т. е. с двунаправленным предсказанием и компенсацией движения. B-кадры имеют наибольшее сжатие.

Таким образом, в стандарте MPEG-2 используются три вида кодирования: внутрикадровое, межкадровое «вперед» с компенсацией движения, межкадровое двунаправленное, также с компенсацией движения.

Полученные кадры объединяются в группы последовательных кадров (GOP — group of pictures). Каждая последовательность начинается с I-кадра и состоит из переменного числа P- и B-кадров.

В начале сцены должен стоять I-кадр, в конце — P-кадр. Увеличивать долю B-кадров можно только в рамках одной сцены, иначе возникнут большие ошибки предсказания и компенсации движения. Поскольку типичная длительность группы кадров (во временном представлении — примерно 0,5 с) значительно меньше характерного расстояния между границами сцен, то в большинстве случаев жесткое задание структуры группы кадров не приводит к существенным визуальным ошибкам из-за того, что смена сцен попадает внутрь группы кадров.

Описательный мультимедиа-стандарт MPEG-7

Спецификация разработана на основе использования методов и достижений интеллектуальных информационных систем в мультимедийных приложениях. Попытки решения данной задачи известны

уже давно — ситуационное моделирование (Ю.И. Клыков, 1974 г.), RX-коды (1969 г.), проект PIPS (Pattern information processing system) программная среда и язык распознавания и генерации сцен NALIG - Native language interprator of graphics (Япония, 1980 г.) и др.

MPEG-7 формально называется «Мультимедиа-интерфейс для описания содержимого» (Multimedia Content Description Interface), он имеет целью стандартизовать описание мультимедийного материала поддерживающего некоторый уровень интерпретации смысла информации, которая может быть передана для обработки ЭВМ. Стандарт MPEG-7 не ориентирован на какое-то конкретное приложение, он стандартизует некоторые элементы, которые рассчитаны на поддержку как можно более широкого круга приложений. Следовательно, средства MPEG-7 позволят формировать описания (т. е. наборы схем описания и соответствующих дескрипторов по желанию пользователя) материала, который может содержать:

- информацию, описывающую процессы создания и производства материала (указатель, заголовок, короткометражный игровой фильм);
- информацию, относящуюся к использованию материала (указатели авторского права, история использования, расписание вещания);
- информацию о характеристиках записи материала (формат записи, кодирование);
- структурную информацию о пространственных, временных или пространственно-временных компонентах материала (разрезы сцены, сегментация областей, отслеживание перемещения областей);
- информацию о характеристиках материала нижнего уровня (цвета, текстуры, тембры звука; описание мелодии);
- концептуальную информацию о реальном содержании материала (объекты и события, взаимодействие объектов);
- информацию о том, как эффективно просматривать материал (конспекты, вариации, пространственные и частотные субдиапазоны и пр.);
- информацию о собрании объектов;
- информацию о взаимодействии пользователя с материалом (предпочтения пользователя, история использования).

MPEG-7 сконструирован так, чтобы учесть все подходы, учитывающие требования основных стандартов, таких как SMPTE Metadata Dictionary, Dublin Core, EBU P/Meta и TV Anytime. Эти стандарты

ориентированы на специфические приложения и области применения, в то время как MPEG-7 пытается быть как можно более универсальным. MPEG-7 использует также схему XML в качестве языка выбора текстуального представления описания материала. Главными элементами стандарта MPEG-7 являются:

- дескрипторы (D) — представление характеристик, которые определяют синтаксис и семантику каждой из характеристик;
- схемы описания (DS — description scheme), которые специфицируют структуру и семантику взаимодействия между компонентами. Эти компоненты могут быть дескрипторами и схемами описания;
- язык описания определений DDL (description definition language), позволяющий создавать новые схемы описания и, возможно, дескрипторы и обеспечивающий расширение и модификацию существующих схем описания;
- системные средства, которые служат для поддержки мультиплексирования описаний, синхронизации описаний и материала, механизмов передачи, кодовых представлений (как текстуальных, так и двоичных форматов) для эффективной записи и передачи, управления и защиты интеллектуальной собственности в описаниях MPEG-7.

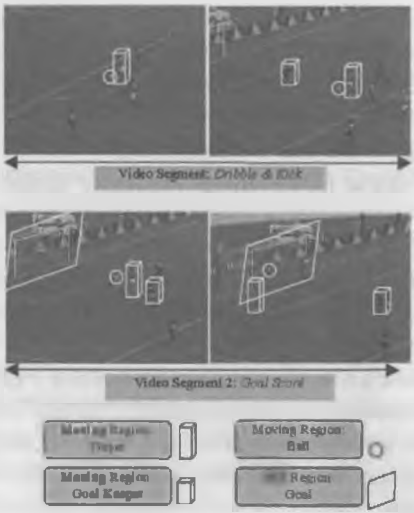
В принципе, любой тип аудиовизуального материала может быть получен с помощью запроса на любой вид материала: видео, музыка, звук и т. д.

Рассмотрим пример описания визуального материала (рис. 3.3, *а*) графовыми представлениями (рис. 3.3, *б*). Этот пример демонстрирует момент футбольного матча. Определены два видеосегмента, одна стационарная область и три движущиеся области.

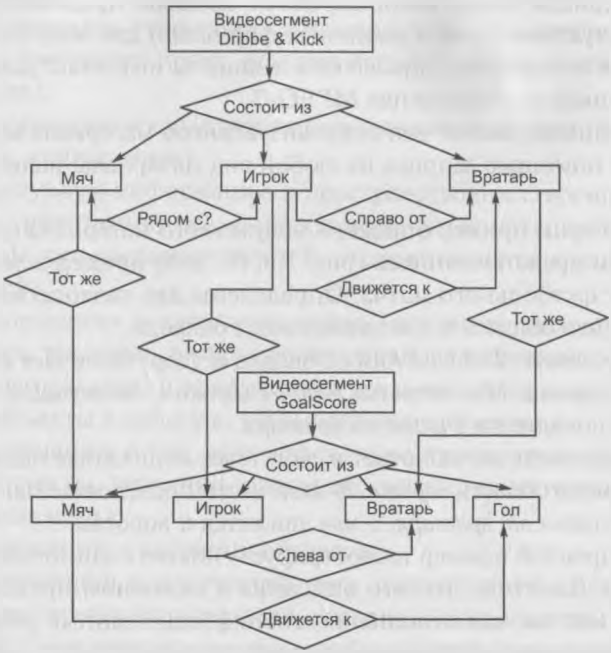
Видеосегмент *Dribble&Kick* (Обводка и удар) включает в себя *мяч*, *вратаря* и *игрока*. *Мяч* остается *рядом с игроком*, *движущимся* к *вратарю*. *Игрок* появляется *справа от вратаря*.

Видеосегмент *гол* включает в себя те же подвижные области плюс стационарную область *ворота*. В этой части последовательности *игрок* находится *слева от вратаря*, а *мяч* движется к воротам.

Этот простой пример иллюстрирует гибкость данного вида представления. Заметим, что это описание в основном представляется структурным, так как отношения, специфицированные ребрами графа, являются чисто физическими, а узлы представляют объекты, которые описываются данными о создании, информацией об использовании и медиаданными, а также дескрипторами низкого уровня,



а



б

Рис. 3.3. Пример видеосегмента и областей ситуации (*а*); соответствующий граф (*б*

такими как цвет, форма, движение. В семантически явном виде доступна только информация из текстовой аннотации (где могут быть специфицированы ключевые слова м я ч, и г р о к или в р а т а р ь).

3.4. Трехмерная компьютерная графика

Трехмерная компьютерная графика (3D computer graphics) отличается от двумерной тем, что трехмерное представление геометрических данных сохраняется в компьютере с целью их обработки для построения и выдачи двумерных изображений, которые могут либо просматриваться в реальном масштабе времени, либо запоминаться для последующего использования.

Трехмерное моделирование — процесс подготовки геометрических данных для трехмерной компьютерной графики, использует многие из алгоритмов, что и в случае двумерной. В программном обеспечении машинной графики часто исчезает грань между двух- и трехмерным, поскольку двумерные приложения могут использовать трехмерные алгоритмы (чтобы, например, описать эффекты освещения) и наоборот.

Как правило, процесс построения трехмерной компьютерной графики может быть представлен в виде последовательности трех шагов:

- создание содержания (трехмерное моделирование, текстурирование, анимация);
- конфигурирование сцены;
- рендеринг (представление)

Во многих случаях между этими фазами нет строгого различия, моделирование может оказаться частью процесса создания сцены.

3.4.1. Моделирование трехмерных сцен

Стадия моделирования состоит в формировании индивидуальных объектов, которые затем размещаются на сцене. Известен ряд методов моделирования, в том числе:

- стереометрия твердых тел;
- использование B-сплайнов;
- аппроксимация многоугольниками.

Моделирование процессов может также включать редактирование поверхности объекта или его материальных свойств (например, цвет яркость, шероховатость или блеск, характер отражения света, прозрачность или непрозрачность, коэффициент преломления и пр.), добавляя текстуры поверхности, карты рельефа или другие особенности.

При моделировании могут также применяться операции, связанные с подготовкой трехмерной модели для анимации (хотя в моделировании сложных процессов это может быть отдельной стадией, известной как «оснащение»). Объекты могут быть оснащены основой, или «костяком», — центральной структурой объекта, которая определяет форму и допустимые движения этого объекта. Это помогает в процессе анимации, поскольку движение основы автоматически определяет состояние соответствующих частей модели.

3.4.2. Конфигурирование сцены

Формирование сцены предполагает размещение в пространстве виртуальных объектов, средств освещения, съемочных камер и других объектов, которые будут в последующем использоваться для создания неподвижных или анимированных изображений. Если речь идет об анимации, на этой фазе часто используется метод ключевых кадров (keyframing), который облегчает моделирование сложного движения в сцене. Ключевые кадры задают при анимации некоторые обязательные промежуточные положения объектов в сцене, перемещения/изменения между которыми (смещение, вращение, масштабирование) реализуются путем интерполяционных вычислений.

Важным аспектом установки сцены является освещение (подсветка). Так же как и в реальных съемках сцен, освещение — существенный фактор эстетического и визуального качества результата.

Рендеринг (Rendering) — окончательная компиляция изображения. На этапе рендеринга осуществляется построение растрового изображения (пикселей).

Процесс рендеринга предполагает использование различных 3D-методов:

- текстурирование, отображение текстур (texture mapping) — технология детализации 3D-изображения, которая лучше всего может быть представлена как обтягивание некоего трехмерного каркаса окрашенной бумагой (конечно, двумерной). Это ресурсоемкий процесс, который должен быть выполнен не только

для каждого пикселя изображения, но и для каждого элемента текстуры (текселя, texel);

- сжатое текстурирование (mip mapping, mip-отображение) — форма сокращения объема данных, при которой создается большее количество текселей, без выполнения эквивалентного необходимого числа вычислений. Если сжатие составляет 1:4, то считывание одного текселя эквивалентно передаче четырех текселей первоначальной структуры. Если использованы надлежащие фильтры, качество изображения может даже повыситься, поскольку при этом сглаживаются зубчатые грани;
- билинейная фильтрация (bi-linear filtering) — считывание четверок текселей, усреднение их характеристик и использование представленного результата как единственного текселя. В результате выравнивается фактура близлежащих участков, изображение сглаживается и уменьшается пикселизация (blocky, pixelated appearance). Билинейная фильтрация является в настоящее время стандартом для большинства графических карт;
- Z-буферизация (Z-buffering) — метод вычисления пикселей, которые следует загрузить в буфер экрана (память, хранящая данные, которые должны быть немедленно выведены). Проблема состоит в том, что акселератор не имеет возможностей «узнать», должен ли рассчитываемый пиксель быть показан немедленно или же позже. Z-буферизация вычисляет и приписывает каждому пикселю некоторый вес «Z». Чем меньше значение Z, тем раньше данный пиксель должен быть выведен на экран;
- сглаживание (anti-aliasing) — технология снижения «шумов», присутствующих в изображении. Например, если объект находится в движении, необходим большой информационный поток, отражающий изменение положения, цвета, размера и т. д. Иногда процессор не успевает обработать всю информацию, и тогда некоторые места заполняются бессмысленным шумом. Сглаживание наряду с mip-отображением удаляет этот шум;
- закраска/штриховка Гуро (Gouraud shading) применяет тени к поверхности объектов, заставляет их выглядеть более объемно. Алгоритм определяет цвета смежных многоугольников и вычисляет гладкий переход между ними, что гарантирует отсутствие резких цветовых переходов в окраске объекта;
- отображение выпуклостей/неровностей (bump mapping) создает иллюзию объемных углублений на плоской поверхности (шершавые стены, бурное море и пр.).

3.4.3. Программные средства трехмерной графики

Мультимедийные технические средства (а особенно компьютерная графика) — наиболее быстро развивающаяся область ИТ, где с высокими темпами постоянно возникают новые версии интерфейсных карт, устройства и принципы.

Программный интерфейс приложения (API, application programming interface) играет роль посредника между прикладной программой и аппаратными средствами ЭВМ (интерфейсная карта и внешнее устройство), на которых она выполняется. Разработчик программного обеспечения пишет обращение к устройству на некотором стандартизированном языке, а не в кодах аппаратных средств ЭВМ. Затем драйвер, написанный изготовителем периферийного устройства или его карты, переводит этот стандартный код в формат, понятный специфической модели периферийных устройств (рис. 3.4).

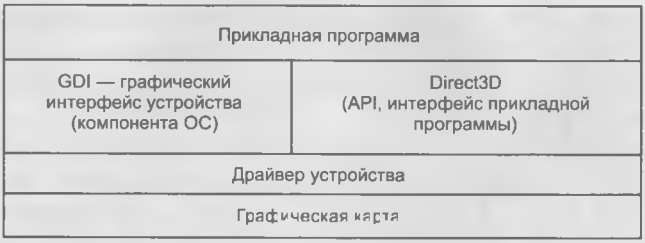


Рис. 3.4. Взаимодействие драйверов с прикладной программой посредством API и GDI

API-интерфейсы обеспечивают доступ к устройствам, таким как микросхемы ускорения трехмерной графики и звуковые платы. Эти интерфейсы управляют функциями нижнего уровня, в том числе ускорением двумерной графики, поддержкой устройств ввода (таких, как джойстик, клавиатура и мышь), а также микшированием и выводом звука.

Графический интерфейс устройства (Graphics Device Interface — GDI) является стандартом Microsoft Windows, который описывает, как следует представлять графические объекты для передачи их на устройства вывода типа мониторов или принтеров.

GDI поддерживает такие задачи, как рисование линий, представление шрифтов и обработка палитр. Он не занимается непосредственно формированием окон, меню и т. д., эти задачи оставлены для

подсистемы пользователя (user32.dll), которая является надстройкой над GDI.

Существенная способность GDI (кроме более прямых методов обращения к аппаратным средствам) — это масштабирование и абстрагирование от конечных устройств.

Используя GDI, можно достаточно просто осуществлять вывод изображений на различные устройства (мониторы, принтера) и ожидать надлежащего результата в каждом случае. Эта способность обеспечивает все приложения WYSIWYG для Microsoft Windows.

API для трехмерной графики

Основными направлениями в обработке трехмерной графики в последние годы являются OpenGL и Direct3D.

OpenGL (Open Graphics Library — открытая графическая библиотека) — межъязыковая и межплатформенная спецификация API для трех- и двумерных приложений компьютерной графики. В своей основе OpenGL — это спецификация, т. е. некоторый документ, который определяет набор функций и содержит точное описание действий, которые они должны выполнять. На основе этой спецификации производители аппаратных средств ЭВМ создают конкретные программные реализации — библиотеки, соответствующие функциям, объявленным в OpenGL-спецификации, используя видеоакселераторы там, где возможно. Обычно оборудование подвергается сертификационным тестам, чтобы квалифицировать его соответствие OpenGL.

Основная функция OpenGL заключается в считывании графических примитивов (точек, линий и многоугольников) и преобразовании их в пиксели. Это происходит в графическом конвейере (рис. 3.5), известном как «машина OpenGL» (OpenGL state machine).

Рис. 3.5. Краткое описание процесса в графическом конвейере

Большинство команд OpenGL или направляет примитивы на конвейер, или задает, как конвейер должен их обрабатывать. OpenGL — процедурный программный интерфейс приложения низкого уровня, который требует от программиста точного описания шагов рендеринга сцен, а также хорошего знания графического конвейера.

DirectX. Впервые предложенный в 1995 г., DirectX представлял собой объединенный набор инструментов программирования, предназначенных для того, чтобы помочь разработчикам создавать мультимедийные приложения для платформы Windows. Охватывая почти все аспекты мультимедийных технологий, например, выпуск DirectX 8.0 включает следующие компоненты:

- DirectX Graphics, который в свою очередь состоит из двух API: DirectDraw — для обработки двумерных растровых изображений; и Direct3D (D3D) — обработчик 3D-графики;
- DirectInput, обрабатывает данные, поступающие от клавиатуры, мыши, джойстиков или других игровых контроллеров;
- DirectPlay — для поддержки сетевых игр;
- DirectSound — проигрывание и запись звука;
- DirectSound3D (DS3D) — для воспроизведения 3D-звучания;
- DirectMusic — проигрывание звукозаписей, подготовленных в DirectMusic Producer;
- DirectSetup — установка и настройка компонентов DirectX;
- DirectX Media, который включает DirectAnimation, DirectShow, DirectX Video Acceleration, Direct3D Retained Mode и DirectX Transform для анимации, воспроизведения мультимедиа, 3Б-интерактивных приложений;
- DirectX Media Objects — поддержка кодирования/декодирования в реальном масштабе времени и создание спецэффектов.

Direct3D включает два компонента API — Immediate Mode (немедленная обработка) и Retained Mode (отложенная обработка).

Первый обеспечивает использование всех 3D-функций видеокарт (подсветка, текстурирование, трансформация объектов и пр.), в то время как второй — более сложные графические технологии, такие как иерархия сцен или анимация.

Графический конвейер Direct3D включает следующие этапы:

- входная сборка (input assembler) поставляет данные на конвейер;
- раскраска вершин многогранников (vertex shader) выполняет операции над вершинами, такие как трансформация, покрытие текстурой, подсветка;

- раскраска геометрических примитивов (geometry shader) — операции над примитивами (треугольники, вершины, линии), иногда — над связанными с ними примитивами. На этой стадии каждый примитив передается дальше или уничтожается, или на его основе создается один или несколько новых примитивов;
- выходной поток (stream output) — запись в память результатов предыдущих стадий. На основе этих данных могут быть организованы итерационные циклы обработки данных на конвейере;
- растеризация (rasterizer) — трансформация примитивов в пиксели, удаление невидимых;
- раскраска пикселей (pixel shader) и другие операции над ними;
- окончательная сборка (output merger), объединение различных типов выходных данных и построение кадра-результата.

Контрольные вопросы

1. Каковы характеристики аналогово-цифрового и цифро-аналогового преобразований аудиоданных?
2. Перечислите методы синтеза звука.
3. Какие характеристики имеют аудиоадаптеры?
4. Что такое ЧМ и WaveTable?
5. Перечислите возможности карты SoundBlaster.
6. Что такое LivelDrive?
7. Охарактеризуйте MIDI-интерфейс.
8. Перечислите основные характеристики форматов аудиосигнала.
9. Какие основные функции реализует программное обеспечение обработки аудиосигналов?
10. Охарактеризуйте методы оптической интерполяции.
11. Перечислите основные характеристики цифровых видеокамер (ЦВК).
12. Что такое схемы цветообразования?
13. Охарактеризуйте форматы графических файлов.
14. Что такое видеозахват?
15. Что такое цветоразностные компоненты?
16. Перечислите форматы записи цифрового видео.
17. В чем заключается сущность M-JPEG сжатия видеоданных?

18. Перечислите основные особенности алгоритмов MPEG-1—4.
19. Что такое GOP?
20. Что такое профили MPEG?
21. В чем сущность стандарта MPEG-7?
22. Перечислите основные фазы работы с трехмерной графикой.
23. Что такое рендеринг?
24. Охарактеризуйте API OpenGL.
25. Что такое DirectX и Direct3D?
26. В чем состоит различие между растровыми изображениями и векторными рисунками?

Глава 4
ИНФОРМАЦИОННЫЕ КРОСС-ТЕХНОЛОГИИ

К данному классу отнесены технологии пользователя, ориентированные на следующие виды преобразования форм представления информации:

- распознавание символов;
- звук—текст;
- текст—звук;
- автоматический перевод.

4.1. Оптическое распознавание символов (OCR)

Отсканированная страница текста представлена в ПК в виде состоящего из пикселей изображения, которое не может быть обработано текстовым редактором. Чтобы группы пикселей превратились в доступные для редактирования символы и слова, изображение должно пройти процесс, известный как оптическое распознавание символов (optical character recognition — OCR).

Первые шаги в области оптического распознавания символов были предприняты в конце 50-х гг. XX в. Принципы распознавания, заложенные в то время, используются в большинстве систем OCR: сравнить изображение с имеющимися эталонами и выбрать наиболее подходящий.

В середине 1970-х гг. была предложена следующая технология ввода информации в ЭВМ:

1) исходный документ печатается на бланке с помощью пишущей машинки, оборудованной *стилизованным* шрифтом (каждый символ комбинируется из ограниченного числа вертикальных, горизонтальных, наклонных черточек, подобно тому, как это делаем мы и сейчас, нанося на почтовый конверт цифры индекса);

2) полученный «машинный документ» считывается оптоэлектрическим устройством (собственно OCR), которое кодирует каждый символ и определяет его позицию на листе;

3) информация переносится в память ЭВМ, образуя электронный образ документа.

Очевидно, что по сравнению с перфолентами (перфокартами) OCR-документ лучше хотя бы тем, что он может быть прочитан и проверен человеком и представляет собой «твердую копию» соответствующего введенного документа. Было разработано несколько модификаций подобных шрифтов, разной степени «удобочитаемости» (OCR A, OCR B и пр., рис. 4.1).

а *б*

Рис. 4.1. Стилизованные шрифты:
а — OCR A; *б* — OCR B

Считывающее устройство при этом представляет собой *специализированный* (считывание стилизованных символов) *интеллектуальный* (их распознавание) сканер.

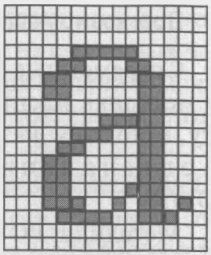
OCR-технология в таком виде просуществовала недолго. В настоящее время исходный документ считывается универсальным сканером, осуществляющим создание растрового образа, а функции распознавания полностью возлагаются на программные продукты.

4.1.1. Основные методы оптического распознавания

Один из самых ранних методов оптического распознавания символов базировался на сопоставлении матриц или на сравнении с образцами букв. Программы оптического распознавания символов, использующие метод сопоставления с образцом, имеют точечные рисунки для каждого символа каждого размера и шрифта (рис. 4.2, *а*). Сравнивая базу данных точечных рисунков с рисунками отсканированных символов, программа пытается их распознавать. Эта методика успешно работала только с непропорциональными шрифтами (подобно Courier) и текстами, где символы хо-

рошо отделены друг от друга. Сложные документы с различными шрифтами оказывались уже вне возможностей таких программ.

Следующим шагом в развитии оптического распознавания символов было выделение признаков. Эта методика основана на идентификации универсальных особенностей символов, что позволяет сделать распознавание их независимым от шрифтов. Если все символы идентифицировать с помощью правил, по которым элементы букв (например, окружности и линии) присоединяются друг к другу, то отдельные символы могли бы быть описаны независимо от их шрифта. Например: символ «а» может быть представлен как состоящий из окружности в центре снизу, прямой линии справа и дуги окружности сверху в центре (рис. 4.2, *б*). Если отсканированный символ имеет эти особенности, он может быть правильно идентифицирован программой оптического распознавания как символ «а».



а

б

Рис. 4.2. Различные подходы к распознаванию символов:
а — сравнение с образцом; *б* — выделение признаков

Выделение признаков было шагом вперед по сравнению с сопоставлением матриц, но практические результаты оказались весьма чувствительными к качеству печати. Дополнительные пометки на странице или пятна на бумаге существенно снижали точность обработки. Устранение такого «шума» само по себе стало целой областью исследований, пытающейся определить, какие биты печати не являются частью отдельных символов. Если шум идентифицирован, достоверные символьные фрагменты могут тогда быть объединены в наиболее вероятные формы символа.

Некоторые программы сначала используют сопоставление с образцом и/или метод выделения признаков для того, чтобы распознать столько символов, сколько возможно, а затем уточняют результат, используя грамматическую проверку правильности написания для восстановления нераспознанных символов. Например, если программа оптического распознавания символов не способна распознать символ «е» в слове «th~ir», программа проверки грамматики может решить, что отсутствующий символ — «е».

Прогнозирующее оптическое распознавание слов

Современные технологии оптического распознавания основаны на методике идентификации не отдельных символов, а целых слов. Эту методику называют прогнозирующим оптическим распознаванием слов (Predictive Optical Word Recognition — POWR).

Методика POWR анализирует различные способы, которыми точки изображения могут быть собраны в символы слова. Каждой возможной интерпретации приписывается некоторая вероятность, что позволяет использовать для принятия решения нейронные сети и прогнозирующие методы моделирования, заимствованные из теории распознавания образов. При этом используются «эксперты» — алгоритмы, разработанные специалистами в различных областях распознавания символов. Один «эксперт» анализирует начертания шрифта, другой — словарную информацию, третий — степень ухудшения качества вследствие «зашумленности» и пр. В процессе привлечения «экспертов» генерируется начальный набор общих гипотез, и каждой гипотезе приписывается вероятность. Исследование продолжается, пока не будет сформирован окончательный ответ. На каждой стадии исследования привлекается новый набор «экспертов» с учетом близости их «областей знаний» к специфической ситуации.

Таким образом, методика POWR способна идентифицировать слова способом, близко напоминающим человеческое визуальное распознавание. Практически методика значительно улучшает точность распознавания слов во всех типах документа. Все возможные интерпретации слова оценивают, комбинируя все источники — от информации пикселя нижнего уровня до контекстных особенностей высокого уровня, в результате чего выбирается самая вероятная интерпретация.

Принципы IPA (целостности, целенаправленности, адаптивности)

Классическая система оптического распознавания, выделив на отсканированном изображении объекты, могущие оказаться буквами, вычисляет для каждого определенный набор параметров. Затем полученные значения поочередно сравниваются с эталонами — наборами тех же параметров, рассчитанных для известных символов. В результате сравнения система примет решение, каким символом следует считать обнаруженный объект. Естественно, в процессе подобного сравнения неизбежно допускается некоторое количество ошибок. Для сокращения числа ошибок используются следующие принципы.

Принцип целостности (integrity). Согласно этому принципу объект рассматривается как целое, состоящее из связанных частей. Связь частей выражается в пространственных отношениях между ними, и сами части получают толкование только в составе предполагаемого целого, т. е. в рамках гипотезы об объекте.

Принцип целенаправленности (purposefulness). Любая интерпретация данных преследует определенную цель. Согласно этому принципу распознавание представляет собой процесс выдвижения гипотез о целом объекте и целенаправленной их проверки.

Принцип адаптивности (adaptability). Подразумевает способность системы к самообучению. Полученная при распознавании информация упорядочивается, сохраняется и используется впоследствии при решении аналогичных задач. Преимущество самообучающихся систем заключается в способности «спрямлять» путь логических рассуждений, опираясь на ранее накопленные знания.

Таким образом, на этапе распознавания символов изображение, согласно принципу целостности, будет интерпретировано как некий объект, только если на нем присутствуют все структурные части этого объекта и эти части находятся в соответствующих отношениях. Выдвигается ряд гипотез относительно того, на что похоже обнаруженное изображение, затем каждая гипотеза целенаправленно проверяется. Причем проверять, верна ли выдвинутая гипотеза, система будет, опираясь на накопленные ранее сведения о возможных начертаниях символа в распознаваемом документе (принцип адаптивности).

Многоуровневый анализ документа (MDA)

Подлежащий распознаванию документ часто выглядит сложнее, чем белая страница со строками черного текста. Иллюстрации, таблицы, колонтитулы, фоновые изображения, применяемые для оформления, усложняют структуру страницы. Чтобы корректно воспроизводить в электронном виде такие документы, все современные OCR-программы начинают распознавание с анализа структуры. Как правило, при этом выделяют несколько иерархически организованных логических уровней. Объект наивысшего уровня только один — собственно страница, на следующей ступени иерархии располагаются таблица и текстовый блок, и т. д.

Любой объект может быть представлен как набор объектов более низкого уровня: буквы образуют слово, слова — строки и т. д. Поэтому анализ всегда начинается в направлении сверху вниз. Программа

делит страницу на объекты, их, в свою очередь, — на объекты низших уровней, и так далее, вплоть до символов. Когда символы выделены и распознаны, начинается обратный процесс — «сборка» объектов высших уровней, который завершается формированием целой страницы. Такая процедура называется многоуровневым анализом документа, или MDA (multilevel document analysis).

Очевидно, что программа, допустившая ошибку при распознавании объекта высокого уровня (например, перепутавшая абзац текста с иллюстрацией), почти не имеет шансов корректно завершить процедуру — итоговый электронный документ будет искажен.

Распознаватели символов (классификаторы)

Выделенные в процессе MDA изображения символов поступают на рассмотрение механизмов распознавания букв, называемых классификаторами.

Растровый классификатор. Классификатор сравнивает символ с набором эталонов, поочередно накладывая изображения друг на друга (рис. 4.3, *а*) . Эталонами в данном случае выступают специально подготовленные изображения; каждое из них объединяет в себе очертания множества вариантов написания того или иного символа. Гипотезы выдвигаются в зависимости от того, с какими эталонами точнее совпало изображение буквы. Сами эталоны строятся методом наложения друг на друга большого количества одних и тех же букв в разных вариантах начертания. Растровый классификатор работает быстро, однако высокой точности не обеспечивает.

Признаковый классификатор. Аналогичен растровому: выдвигает гипотезы исходя из степени совпадения параметров символа с эталонными значениями. Оперирует определенными числовыми признаками, такими, например, как длина периметра, количество черных точек в разных областях или вдоль различных направлений и т. п. (рис. 4.3, *в*). В определенных условиях способен работать почти так же быстро, как растровый. Точность работы признакового классификатора во многом зависит от качества признаков, выбранных для каждого символа. Под качеством признаков в данном случае понимается их способность максимально точно, но без избыточной информации, охарактеризовать начертание буквы.

Контурный классификатор. Представляет собой разновидность признакового классификатора. От вышеописанного отличается тем, что признаки вычисляются не по полному изображению символа, а

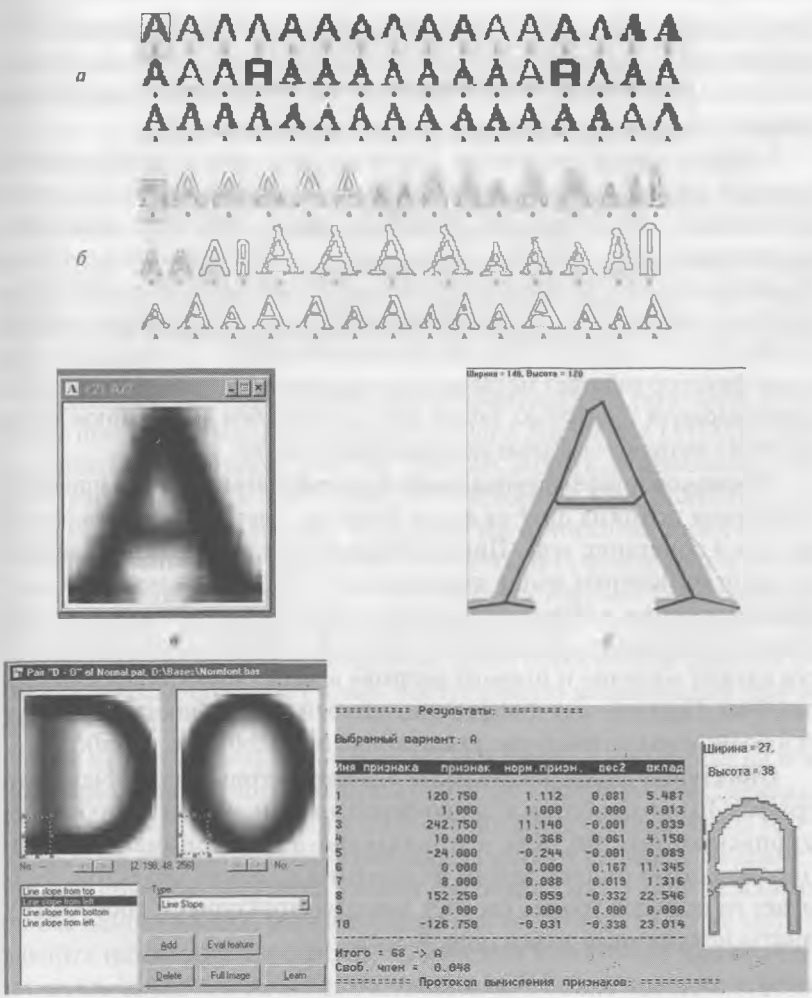


Рис. 4.3. Распознавание символов:

а — растровые эталоны буквы «A»; *б* — контурные эталоны буквы «A»; *в* — изображение буквы для признакового классификатора, определяющего определенные признаки (например, количество серого в какой-либо точке буквы); *г* — пример обучения структурного классификатора (заметен «скелет» буквы «A»); *д* — пример работы признаково-дифференциального классификатора. Чтобы верно выбрать одну из похожих букв («D» и «O»), классификатор вычисляет признак (наклон линии в ключевой зоне); *е* — пример работы структурно-дифференциального классификатора — чтобы выбрать одну из похожих букв (сочетание «fl» и «A»), сравнивается структура букв, обращается особое внимание на внешний профиль

по его контуру (рис. 4.3, *б*). Этот быстродействующий классификатор предназначен для распознавания текста, набранного декоративными шрифтами (например, стилизованного под готический, старорусский стиль и т. п.).

Структурный классификатор. Первоначально был создан и использовался для распознавания рукописного текста, однако в последнее время применяется и для обработки печатных документов. Этот классификатор проводит структурный анализ символа, раскладывая последний на элементарные составляющие (отрезки, дуги, окружности, точки) и формируя точную схему анализируемого знака (рис. 4.3, *г*). Затем полученная схема (структурное описание буквы) сравнивается с эталоном. Этот классификатор работает медленнее растрового и признакового, но отличается высокой точностью. Более того, он способен «мысленно» восстанавливать непропечатанные или залитые символы.

Признаково-дифференциальный классификатор. Предназначен для различения похожих друг на друга объектов, таких, например, как буква «m» и сочетание «rn». Принципиальное отличие этого классификатора от описанных выше заключается в том, что он не анализирует все изображение, а обращается только к тем частям объекта, где может находиться признак правильного ответа. В случае с «m» и «rn» признаком служит наличие и ширина разрыва в месте касания предполагаемых букв. Признаково-дифференциальный классификатор используется во многих системах распознавания символов (рис. 4.3, *д*).

Структурно-дифференциальный классификатор. Аналогичен структурному. Был разработан и первоначально применялся для обработки рукописных текстов. Как и признаково-дифференциальный, этот классификатор решает задачи различения похожих объектов, но работает на порядок точнее (за счет анализа структуры) и способен «узнавать» искаженные знаки (рис. 4.3, *е*).

4.1.2. Технологии Finereader

Типичным представителем семейства программ оптического распознавания символов является система ABBYY FineReader, технологический процесс которой включает следующие шаги:

- сканирование исходного документа (страницы);
- разметку областей (ручную или автоматическую), требующих различных видов обработки (страницы разворота книги, таблицы, рисунки, колонки текста и пр.);

- распознавание — создание и вывод на экран текстового файла (с вставленными рисунками и таблицами, если это необходимо);
- контроль правильности (ручной, автоматический, полуавтоматический);
- вывод информации в выходной файл в заданном формате (.DOC или .RTF для Word, .XLS для Excell и пр.).

Данные, полученные на каждом этапе (изображение, текстовый файл), сохраняются в виде *пакета* (страницы с номером), что позволяет в любой момент вернуться и повторить разметку, распознавание и пр.

Распознавание любого документа производится поэтапно, с помощью процедуры многоуровневого анализа документа (MDA). Деление страницы на объекты низших уровней, вплоть до отдельных символов, распознавание этих символов и «сборку» электронного документа FineReader проводит, опираясь на принципы целостности, целенаправленности и адаптивности (IPA). В соответствии с этим в первую очередь выдвигаются гипотезы относительно типов обнаруженных объектов, затем они целенаправленно проверяются. При этом программа учитывает найденные ранее особенности данного документа, а также сохраняет вновь поступающую информацию (обучается).

При использовании алгоритма MDA в системе FineReader на всех этапах многоуровневого анализа существует возможность обратной связи — результаты анализа на одном из нижних уровней всегда могут повлиять на действия с объектами более высоких уровней. Например, если все объекты текущего уровня распознаны, но при детальном анализе одного из них, определенного как текстовый блок, не удается выделить ни абзацы, ни строки, происходит повторный анализ. При повторном анализе предыдущего уровня могут быть внесены коррективы (текст, наложенный на фоновое изображение), и после дополнительной обработки распознавание будет продолжено без ошибок. Наличие обратной связи в процедуре MDA дает возможность резко понизить вероятность ошибок, связанных с неверным распознаванием объектов более высоких уровней.

Распознавание от уровня «страница» до уровня «слово»

На первом этапе распознавания система структурирует страницу, выделяет на ней текстовые блоки. Современные документы часто со-

держат всевозможные элементы дизайна: иллюстрации, колонтитулы, цветной фон или фоновые изображения и т. д. Основная задача на данном этапе состоит в том, чтобы отделить текст от иллюстраций и «подложенных» текстур.

Все современные системы распознавания начинают процесс с создания черно-белого изображения документа. При этом подлежащее анализу изображение чаще всего цветное или полутоновое (т. е. состоящее из разных оттенков серого цвета, подобно картинке на экране черно-белого телевизора). Любая OCR-система прежде всего преобразует такое изображение в монохромное, состоящее только из черных и белых точек. Процесс преобразования называется *бинаризацией*, он всегда предшествует детальной обработке распознаваемой страницы.

Блок текста, состоящий из строк, должен иметь характерную линейчатую структуру. Разделив этот блок на строки, можно приступать к выделению слов. Однако на практике столь простые варианты встречаются нечасто. В любом документе, где строки текста наложены на цветной фон, при бинаризации вокруг каждого символа обнаружатся десятки и сотни «лишних» точек, оставшихся от фона. Работая с таким «загрязненным» текстом, большинство OCR-программ не сможет уверенно распознавать символы, поскольку лишние точки будут искажать очертания букв и даже границы строк, приводя к ошибкам.

Для повышения качества выделения строк FineReader использует процедуры интеллектуальной *фильтрации фоновых текстур* и *адаптивной бинаризации*. Первая позволяет уверенно отделять строки текста от сколь угодно сложного фона, вторая — гибко выбирать оптимальные для данного участка параметры бинаризации. Естественно, к этим процедурам система прибегает не всегда, а лишь в тех случаях, когда предварительный анализ указывает на подобную необходимость.

В каждом конкретном случае FineReader выбирает подходящий «инструмент», опираясь на информацию, накопленную в процессе анализа документа.

Распознавание от уровня «слово» до уровня «символ»

Разделив строку на отдельные слова, FineReader приступает к обработке символов. Разделение слов на символы и собственно распознавание букв, как и все остальные механизмы многоуровневого ана-

лиза документа, реализованы в виде составных частей единой процедуры. Это позволяет в полной мере использовать преимущества принципов IPA.

При распознавании символов в системе ABBYY FineReader применяются следующие типы классификаторов: растровый, контурный, признаковый, структурный, признаково-дифференциальный и структурно-дифференциальный. Каждый классификатор опирается в процессе распознавания на те или иные эталоны, либо изначально заложенные разработчиками, либо выработанные в ходе анализа документа.

В самых общих чертах процесс обработки символа выглядит так: растровый и признаковый классификаторы анализируют изображение и выдвигают несколько гипотез относительно того, какая буква им представлена. Каждой гипотезе присваивается определенная оценка (так называемый *вес гипотезы*), и список гипотез сортируется по весу (по степени уверенности).

Затем, в соответствии с принципами IPA, FineReader приступает к целенаправленной проверке имеющихся гипотез с помощью дифференциального признакового классификатора. В тех случаях, когда требуется различить два похожих символа (например, «I» и «l»), к анализу подключается структурный или структурно-дифференциальный классификатор. В результате построения и анализа полной схемы распознаваемого знака изменяются веса гипотез.

Однако окончательное решение относительно обрабатываемого символа на данном этапе не принимается. По окончании работы всех задействованных классификаторов список гипотез ранжируется по степени достоверности.

Структурирование гипотез

При распознавании букв FineReader оперирует множеством гипотез, учитывающих возможные варианты деления строки на слова, слова на буквы и т. д. Для быстрого и точного принятия решений система объединяет гипотезы в многоуровневые структуры — модели. Существуют следующие типы моделей слова: *словарное слово, несловарное слово* (для каждого из поддерживаемых языков распознавания построены соответствующие разновидности), *e-mail* или *URL*, *цифры с префиксом* или *суффиксом*, *регулярное выражение* и т. д. В результате структурирования количество подлежащих проверке гипотез сильно сокращается.

Рассмотрим процесс структурирования на примере слова «turn» (рис. 4.4). Предположим, при разделении слова на символы было выдвинуто две гипотезы: первая соответствует прочтению «tum», вторая — «turn». Классификаторы, обработав символы, в свою очередь предложили для каждой буквы обоих слов некоторый ряд гипотез, отсортированных по весу. Следующий шаг кажется очевидным — теперь надо выбрать гипотезы с максимальным весом. Однако далеко не всегда наиболее вероятная гипотеза в итоге оказывается истинной. Лучший способ принять правильное решение — перейти на уровень «слово» и выяснить, какой из вариантов больше остальных похож на правильный.

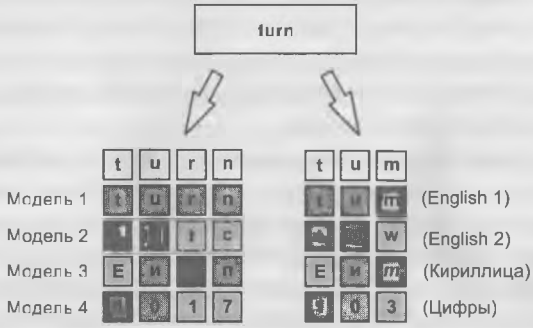


Рис. 4.4. Гипотезы о разделении слов на буквы

В рассматриваемом примере произойдет следующее: контекстная проверка покажет, что весь текст состоит из английских слов, и вес моделей «слово — английский язык» значительно увеличится, а моделей «слово — кириллица» соответственно уменьшится. Модель «цифры» также останется позади в силу крайне малого суммарного веса составляющих гипотез. Затем словарная проверка подтвердит, что в словаре английского языка слова «tum» нет, а «turn» — есть. Следовательно, гипотеза относительно слова «turn» приобретет еще больший вес, что позволит ей в дальнейшем оказаться «победителем». Заметим, что «авторитет» словаря значительно выше, нежели у любого классификатора, поэтому в данном примере даже при полностью слившихся буквах r и n итоговое решение будет принято правильно.

Тем не менее, словарная проверка в системе FineReader не является «последней инстанцией»: она не определяет правильность гипотезы, как это бывает в других системах, а лишь изменяет вес выдвинутых предположений.

4.2. Системы распознавания речи

Процесс распознавания речи (STT — speech-to-text) в последние годы сделал гигантский скачок вперед. В наибольшей мере ее стимулирует существование специфических областей компьютеризации, где голосовые команды являются наиболее приемлемым или даже единственно возможным решением. К ним относятся телефонный доступ к автоматическим справочным системам, управление удаленным компьютером или мобильным портативным устройством, осуществляемое во время движения.

Компьютерное распознавание речи существует с 1993 г., однако работы в этом направлении начались гораздо раньше, в частности, в 1962 г. фирма IBM представила первое коммерческое устройство речевого вывода — модель 7772.

4.2.1. Принципы распознавания речи

Системы распознавания речи обычно состоят из двух компонент, которые могут быть выделены в блоки или в подпрограммы, — акустической и лингвистической. Лингвистическая часть может включать в себя фонетическую, фонологическую, морфологическую, синтаксическую и семантическую модели языка. Акустическая модель отвечает за представление речевого сигнала. Лингвистическая модель интерпретирует информацию, получаемую от акустической модели, и отвечает за представление результата распознавания потребителю.

Акустическая модель. Существуют два подхода к построению акустической модели: изобретательский и бионический. Оба подхода имеют свои достоинства и недостатки. Первый базируется на результатах поиска механизма функционирования акустической модели. При втором моделируется работа естественных систем.

Лингвистическая модель. Лингвистический блок подразделяется на следующие слои (уровни); фонетический, фонологический, морфологический, лексический, синтаксический, семантический. Все уровни содержат априорную информацию о структуре естественного языка. Поскольку естественный язык несет сильно структурированную информацию, для каждого естественного языка может потребоваться своя уникальная лингвистическая модель (отсюда трудности русификации сложных систем распознавания речи зарубежной разработки).

В соответствии с данной моделью на первом (фонетическом) уровне производится преобразование входного (для лингвистического блока) представления речи в последовательность фонем как наименьших единиц языка. Считается, что в реальном речевом сигнале можно обнаружить лишь аллофоны — варианты фонем, зависящие от звукового окружения.

На следующем (фонологическом) уровне накладываются ограничения на комбинаторику фонем (аллофонов) — не все сочетания фонем (аллофонов) встречаются, а те, что встречаются, имеют различную вероятность появления, зависящую еще и от окружения. Для описания этой ситуации используется математический аппарат цепей Маркова.

Далее, на морфологическом уровне оперируют со слогоподобными единицами речи более высокого уровня, чем фонема. Иногда они называются морфемами. Они накладывают ограничение уже на структуру слова, подчиняясь закономерностям моделируемого естественного языка.

Лексический уровень охватывает слова и словоформы того или иного естественного языка, также внося важную априорную информацию о том, какие слова возможны для данного естественного языка. Семантика устанавливает соотношения между объектами действительности " словами, их обозначающими. Она является высшим уровнем языка. При помощи семантических отношений интеллект человека производит как бы сжатие речевого сообщения в систему образов, понятий, представляющих суть речевого сообщения.

Рис. 4.5. Комплексная схема речевых технологий

В соответствии с этим решение задачи речевых технологий можно представить в виде схемы рис. 4.5.

В основе лежит выделение фонем из потока слитной речи в режиме реального времени, их кодирование и последующее восстановление.

Чтобы создать систему распознавания, необходимо «привязать» сегментацию к конкретному языку с помощью двух словарей — «звукового», сопоставляющего реальным звукам речи определенные фонемы, т. е. смыслоразличительные единицы (на слух мы, как правило, воспринимаем именно фонемы родного языка, не замечая различий между их вариантами), и «фонетико-орфографического», который будет переводить фонемную запись в письменную.

4.2.2. Практическая реализация

Многие научные центры, в том числе и в нашей стране, брались за решение этой проблемы (фундаментальные исследования теории языка, которые велись в 70-х гг. в СССР, легли в основу многих современных продуктов), но первый серьезный прорыв в области речевых технологий удалось сделать только в 1986 г. в Defense Advanced Research Project Agency (DARPA) — Агентстве перспективных исследований Министерства обороны США.

Успех связан с тем, что было уменьшено число фонетических структур, предлагаемых распознающему устройству. Для реализации этой задачи применили так называемую «скрытую марковскую модель». Имея последовательность символов, сгенерированную марковской моделью, можно однозначно восстановить породившую ее последовательность состояний, но лишь только при том условии, что каждый символ соответствует одному состоянию.

В процессе цифровой обработки речевой сигнал подвергается сначала логарифмическому, а затем обратному преобразованию Фурье, в результате чего отыскиваются первые коэффициенты, несущие наиболее существенную информацию об огибающей спектральной характеристике сигнала. Собственно, современные развитые коммерческие программы распознавания речи и отличаются именно способом реализации механизма выбора из встроенной (или созданной пользователем) базы данных наиболее вероятного набора фонем (минимально значимых элементов, из которых состоит слово).

На первом этапе компьютер записывает звук речи в виде цифровой аудиопоследовательности и делит ее на фрагменты длительно-

стью несколько миллисекунд. Программа сравнивает эти аудиофрагменты с записанными в память речевыми образцами. Качество базы данных образцов является наиболее важным условием для безошибочного распознавания речи. Она содержит фрагменты речи различных людей с разными особенностями произношения, такими, как снижение звука, диалект, выделение слогов и произношение. Эта часть системы распознавания речи называется системой, не зависящей от говорящего.

Систему, не зависящую от говорящего, дополняет система распознавания говорящего. В основе последней лежит понятие фонемы — наименьшей акустической единицы языка. В процессе тренировки программное обеспечение распознает наиболее важные признаки произношения пользователем фонем и записывает полученные данные в виде профиля говорящего. Очень важно, чтобы в дальнейшем во время диктовки пользователь по возможности точно выдерживал мелодию речи и произношение.

В системе распознавания говорящего при определении «сомнительных слов» используется тот факт, что после определенного слова могут следовать (и имеют при этом смысл) лишь немногие конкретные слова.

4.2.3. Классификация систем распознавания речи

Классификация по назначению:

- командные системы;
- системы диктовки текста.

По потребительским качествам:

- диктороориентированные (тренируемые на конкретного диктора);
- дикторонезависимые;
- распознающие отдельные слова;
- распознающие слитную речь.

По механизмам функционирования:

- простейшие (корреляционные) детекторы;
- экспертные системы с различным способом формирования и обработки базы знаний;
- вероятностно-сетевые модели принятия решения, в том числе нейронные сети.

Разумеется, относительно проще реализовать программу, способную распознавать только ограниченный, совсем небольшой набор

управляющих команд и символов. Это, например, могут быть цифры от 0 до 9, слова «да», «нет», односложные команды типа «открыть», «закрыть», «выйти» и т. п. Такие программы появились первыми и уже давно применяются в компьютерной телефонии для голосового набора телефонного номера или выбора пункта меню. Если в словарь добавить названия букв алфавита, то, в принципе, по буквам можно продиктовать и любое слово или название — например, при заказе билета таким путем можно ввести станцию назначения.

В подобных системах распознавание происходит без предварительной настройки под конкретного пользователя, т. е. они независимы от диктора (speaker-independent).

Программы для диктовки текстов (еще одно очевидное применение функции распознавания речи) первоначально могли понимать только так называемую «раздельную» речь, в которой после каждого произнесенного слова требовалось сделать небольшую паузу. Такая манера говорить неестественна — в процессе обычного человеческого разговора интенсивность звука практически никогда не падает до нуля. Распознавать диктовку текстов общей тематики, выполняемую в манере слитной речи, коммерческие программы научились только в 1997 г. Разумеется, что словарь подобных пакетов обслуживает так называемую общую тематику и охватывает лишь небольшую часть всей лексики. Значительная часть пользователей этим словарем не ограничивается и подключает еще специализированные (технические, медицинские, юридические и др.) словари. На качество распознавания влияет даже манера ведения разговора — непринужденную беседу с относительно небольшим количеством используемых лексических единиц запротоколировать гораздо сложнее, чем размеренный диктант. Проблема заключается, в основном, в вариативности и наличии большого количества различных смысловых оттенков у самых простых конструкций. Тяжелее всего распознаются короткие слова. Серьезнейшая проблема — одно-двухбуквенные слова. Заставить компьютер различать английские «a» и «an» можно, только обращаясь к контексту всей фразы.

4.3. Системы генерации речи

Задача, обратная распознаванию, — синтез речи (Text-to-Speech — TTS). Известно, что синтезированная речь воспринимается человеком хуже, чем живая, причем это особенно заметно при передаче по

каналу телефонной связи, т. е. как раз в тех условиях, в которых было бы наиболее заманчиво ее использовать. Тем не менее, эксперты отмечают улучшение звучания синтезированной английской речи.

Для характеристики качества речи обычно используют такие понятия, как естественность звучания, фонетическая разборчивость, комфортность восприятия и время привыкания.

Естественность звучания характеризует то, насколько близок синтезированный звук к человеческой речи. Пока еще не существует синтезатора, прослушав который, человек не мог бы указать, что это неестественный звук.

Фонетическая разборчивость характеризует то, насколько слушателю легко или трудно разобрать фонемы, произносимые синтезатором. Здесь надо понимать, что неестественная с металлическим звуком «речь робота» может обладать высокой фонетической разборчивостью, т. е. слушатель с легкостью может различить фонемы (слоги) произносимых слов. В то же время в естественной речи разборчивость может быть невысокой (представьте себе бубнящего человека — речь на сто процентов естественная, а ничего не понять). Так происходит потому, что для придания естественности звучания синтезируемая речь проходит дополнительную фильтрацию, в результате чего получает дополнительные обертона (их богатство во многом и определяет близость синтезированной речи человеческой).

Комфортность восприятия и время привыкания показывают субъективную оценку слушателем качества синтезируемой речи. Несмотря на свою субъективность, с точки зрения пользователя это самые главные критерии, по которым оценивается работа синтезатора. Долгое прослушивание синтезированной речи не должно вызывать чрезмерного утомления, а время привыкания должно быть достаточно коротким, чтобы обеспечить легкий переход от одного синтезатора к другому.

4.3.1. История проблемы

В 1779 г. российский профессор Кристиан Краценштейн (иногда упоминается в источниках как Кристиан Готтлиб) построил акустическую модель, позволяющую создавать гласные звуки, используя различные геометрические формы резонаторов, как это показано на рис. 4.6.

При этом, возможно, использовался аддитивный синтез (см. гл. 3), как в обычных органах (напомним, что один из регистров

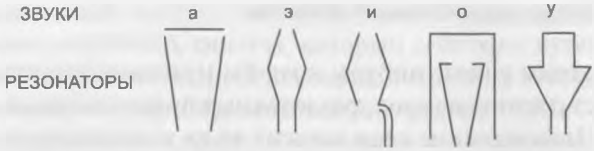


Рис. 4.6. Акустическая модель К. Краценштерна

органа так и называется — vox humanum — голос человеческий)[1]. В 1791 г. Вольфганг фон Кемпелен (Volfgang von Kempelen) представил акустико-механическую говорящую машину, которая воспроизводила определенные звуки и их комбинации. Шипящие и свистящие выдувались с помощью специального меха с ручным управлением. Затем это изобретение было улучшено ученым Чарльзом Уитстоуном (Charles Wheatstone) и уже могло воспроизводить гласные и большинство согласных звуков. В 1846 г. Джезеф Фабер представил свой говорящий орган, в котором была реализована попытка синтезирования не только речи, но и пения. В конце XVIII в. знаменитый ученый Александр Белл (Alexander Graham Bell) создал собственную «говорящую» механическую модель, очень схожую с конструкцией Уитстоуна. Начиная с 1920 г. наступила эра электрических инструментов, при этом основным видом синтеза оставался а д д и т и в н ы й.

Ключевой датой в развитии вокодеров является 1939 г. Именно в этом году ученый-изобретатель Хомер Дадли (Homer.W. Dudley) из Bell Laboratories представил устройство Parallel Bandpass Vocoder. Самая ранняя модель называлась The Voder The Machine That Talks (VODER — машина, которая говорит).

Voder, представленный в 1939 г., управлялся человеком-оператором. Вот как описывает свои впечатления Ванневар Буш (Vannevar Bush) в работе As We May Think: «На мировой выставке 1939 г. было показано устройство, называемое Voder. Девушка-оператор нажимала на его клавиши, и Voder воспроизводил звук, похожий на речь. Это происходило без использования человеческих голосов, нажатие на клавиши просто вызывало комбинации нескольких вибраций, созданных электронным способом, которые воспроизводились с помощью громкоговорителя».

Отметим, что алгоритмические модели синтезаторов речи с того времени практически не изменилась. При этом эти системы развивались практически параллельно с аналоговыми синтезаторами.

---

[1] Есть также и регистр, именуемый vox caelesta — глас небесный.

4.3.2. Методы озвучивания текста

Рассмотрим какой-нибудь, хотя бы минимально осмысленный текст. Текст состоит из слов, разделенных пробелами и знаками препинания. Произнесение слов зависит от их расположения в предложении, а интонация фразы — от знаков препинания и довольно часто от типа применяемой грамматической конструкции — в ряде случаев при произнесении текста слышится явная пауза, хотя какие-либо знаки препинания отсутствуют. Произнесение зависит и от смысла слова — сравните, например, выбор одного из семантических вариантов «за 'мок» или «замо 'к» для одного и того же слова «замок».

Основная классификация стратегий, применяемых при озвучивании речи — это:

- построение действующей модели речепроизводящей системы человека;
- моделирование акустического сигнала.

Первый подход известен под названием артикуляторного синтеза. Второй подход представляется на сегодняшний день более простым, поэтому он гораздо лучше изучен и практически более успешен. Внутри него выделяется два основных направления — формантный синтез по правилам и компилятивный синтез.

Формантные синтезаторы используют возбуждающий сигнал, который проходит через цифровой фильтр, построенный на нескольких резонаторах, похожих на резонансы голосового тракта. Разделение возбуждающего сигнала и передаточной функции голосового тракта составляет основу классической акустической теории речеобразования. Компилятивный синтез осуществляется путем склейки нужных единиц компиляции из имеющегося набора.

На этом принципе построен ряд систем, использующих разные типы единиц и различные методы составления инвентаря. В таких системах необходимо применять обработку сигнала для приведения частоты основного тона, энергии и длительности единиц к тем, которыми должна характеризоваться синтезируемая речь. Кроме того, требуется, чтобы алгоритм обработки сигнала сглаживал разрывы в формантной (и спектральной в целом) структуре на границах сегментов.

В системах компилятивного синтеза применяются два типа алгоритмов обработки сигнала: LP (Linear Prediction — линейное предсказание) и PSOLA (Pitch Synchronous Overlap and Add). LP-синтез осно-

ван в значительной степени на акустической теории речеобразования, в отличие от PSOLA-синтеза, который действует путем простого разбиения звуковой волны, составляющей единицу компиляции, на временные окна и их преобразования. Алгоритмы PSOLA позволяют добиваться хорошего сохранения естественности звучания при модификации исходной звуковой волны.

4.3.3. Обобщенная функциональная структура речевого синтезатора

Структура идеализированной системы автоматического синтеза речи состоит из нескольких блоков:

- определение языка текста;
- нормализация текста;
- лингвистический анализ (синтаксический, морфемный и т. д.);
- формирование просодических характеристик;
- фонемный транскриптор;
- формирование управляющей информации;
- получение звукового сигнала

Такая обобщенная схема содержит компоненты, которые можно обнаружить во многих системах. Разработчики конкретных систем уделяют много внимания отдельным блокам и реализуют их по-разному, в соответствии с практическими требованиями.

Модуль лингвистической обработки. Прежде всего, текст, подлежащий прочтению, поступает в модуль лингвистической обработки. В нем производится определение языка (в многоязычной системе синтеза), а также отфильтровываются не подлежащие произнесению символы. В некоторых случаях используются процедуры проверки орфографических и пунктуационных ошибок. Затем происходит нормализация текста, т. е. осуществляется разделение введенного текста на слова и остальные последовательности символов. К символам относятся, в частности, знаки препинания и символы начала абзаца. Все знаки пунктуации очень информативны.

Для озвучивания цифр разрабатываются специальные блоки. Если читать цифры как цифры, а не как числа, которые должны быть правильно оформлены грамматически, то преобразование цифр в последовательности слов является относительно легкой задачей. Однако цифры, имеющие разное значение и функцию, произносятся по-разному. Например, для многих языков можно говорить о суще-

ствовании отдельной произносительной подсистемы телефонных номеров.

Лингвистический анализ. После процедуры нормализации каждому слову текста (каждой словоформе) необходимо приписать сведения о его произношении, т. е. превратить в цепочку фонем, или, иначе говоря, создать его фонемную транскрипцию. Во многих языках, в том числе и в русском, существуют достаточно регулярные правила чтения — правила соответствия между буквами и фонемами (звуками), которые, однако, могут требовать предварительной расстановки словесных ударений. В то же время в английском языке правила чтения очень нерегулярны, и задача данного блока для английского синтеза тем самым усложняется. В любом случае при определении произношения имен собственных, заимствований, новых слов, сокращений и аббревиатур возникают серьезные проблемы. Просто хранить транскрипцию для всех слов языка не представляется возможным из-за большого объема словаря и контекстных изменений произношения одного и того же слова во фразе.

Кроме того, следует корректно рассматривать случаи графической омонимии: одна и та же последовательность буквенных символов в различных контекстах порой представляет два различных слова и читается по-разному (например, выше приведенный пример слова «замок»).

Для языков с достаточно регулярными правилами чтения одним из продуктивных подходов к переводу слов в фонемы является система контекстных правил, переводящих каждую букву/буквосочетание в ту или иную фонему, т. е. автоматический фонемный транскриптор. Однако чем больше в языке исключений из правил чтения, тем хуже работает этот метод. Стандартный способ улучшения произношения системы состоит в занесении нескольких тысяч наиболее употребительных исключений в словарь. Альтернативное подходу «слово-буква-фонема» решение предполагает морфемный анализ слова и перевод в фонемы морфов (т. е. значимых частей слова: приставок, корней, суффиксов и окончаний). Однако в связи с разными пограничными явлениями на стыках морфов разложение на эти элементы представляет собой значительные трудности. В то же время для языков с богатой морфологией, например, для русского, словарь морфов был бы компактнее. Морфемный анализ удобен еще и потому, что с его помощью можно определять принадлежность слов к частям речи, что очень важно для грамматического анализа текста и задания его просодических характеристик. Особую

проблему для данного этапа обработки текста образуют имена собственные.

Формирование просодических характеристик. К просодическим характеристикам высказывания относятся его тональные, акцентные и ритмические характеристики. Их физическими аналогами являются частота основного тона, энергия и длительность. В речи просодические характеристики высказывания определяются не только составляющими его словами, но также тем, какое значение оно несет и для какого слушателя предназначено, эмоциональным и физическим состоянием говорящего и многими другими факторами. Многие из этих факторов сохраняют свою значимость и при чтении вслух, поскольку человек обычно интерпретирует и воспринимает текст в процессе чтения. Таким образом, от системы синтеза следует ожидать примерно того же, т. е. что она сможет понимать имеющийся у нее на входе текст, используя методы искусственного интеллекта. Однако этот уровень развития компьютерной технологии еще не достигнут, и большинство современных систем автоматического синтеза стараются корректно синтезировать речь с эмоционально нейтральной интонацией. Между тем, даже эта задача на сегодняшний день представляется очень сложной.

Формирование просодических характеристик, необходимых для озвучивания текста, осуществляется следующими тремя основными блоками:

- расстановки синтагматических границ (паузы);
- приписывания ритмических и акцентных характеристик (длительность и энергия);
- приписывания тональных характеристик (частота основного тона).

При расстановке синтагматических границ определяются части высказывания (синтагмы), внутри которых энергетические и тональные характеристики ведут себя единообразно и которые человек может произнести на одном дыхании. Если система не делает пауз на границах таких единиц, то возникает отрицательный эффект: слушающему кажется, что говорящий (в данном случае — система) задыхается. Помимо этого, расстановка синтагматических границ существенна и для фонемной транскрипции текста. Самое простое решение состоит в том, чтобы ставить границы там, где их диктует пунктуация. Для наиболее простых случаев, когда пунктуационные знаки отсутствуют, можно применить метод, основанный на использовании служебных слов.

4.3.4. Примеры программного обеспечения синтеза и распознавания речи

В настоящее время существует целый ряд пакетов программных средств (машин) синтеза и распознавания речи, которые в том числе разработаны для использования совместно с MS Speech API.

smARTspeak CS — настраиваемая независимая от языка «машина» распознавания речи для набора цифр, указания имен и речевой навигации, т. е. для приложений, используемых в сотовых телефонах и беспроводных устройствах. Созданный для использования в указанных устройствах, smARTspeak CS удовлетворяет потребностям как пользователей, так и разработчиков: иммунитет к фоновому шуму, малые требования к процессору и памяти, совместимость с MS SAPI 5.0, оптимизация для средств быстрой разработки приложений и для интеграции в сертифицированные устройства.

Digalo. ПО синтеза для русского языка Digalo — продукт французской фирмы Elan Informatique. Digalo различает буквы «Е» и «Ё». Ошибки в ударениях, в основном, приходятся на некоторые фамилии и имена, малоупотребительные слова и термины, замечено не всегда корректное озвучивание чисел и очень акцентированное произнесение слов «нет» и «не».

PC Voice Club. ПО синтеза речи Клуба голосовых технологий при Научном парке МГУ им. М.В. Ломоносова. При его создании использована базовая технология синтеза речи, разработанная на филологическом факультете МГУ. Синтезатор характеризуется высоким качеством синтеза речи, что позволяет прослушивать тексты без их специальной подготовки. Позволяет синтезировать речь на английском и русском языках. Кроме того, имеет около десятка голосовых типажей (робот, эльф, мышь и пр.). Имеются возможности редактирования голосов. Помимо стандартных функций синтеза речи имеется дополнительная функция встраивания в текст управляющих символов, которые позволяют устанавливать паузы, изменять тембр, тон и длительность звучания. К примеру, можно, отредактировав текст, заставить синтезатор петь.

4.3.5. Синтезатор русской речи

Творческий коллектив радиофизиков и программистов разработал серию программных продуктов под общим названием «Говорящая мышь». В основе речевого синтеза лежит идея совмещения методов конкатенации и синтеза по правилам.

Для создания удобного и быстрого режима изменения и верификации правил, включенных в разные блоки синтезирующей системы, был разработан формализованный и в то же время содержательно прозрачный и понятный язык записи правил, который легко компилируется в исходные тексты программ.

Правила *интонационного обеспечения* разработаны, чтобы определить временные и тональные характеристики базовых элементов компиляции, которые при обработке синтагмы выбираются из библиотеки в нужной последовательности специальным процессором (блоком кодировки). Необходимые для этого предварительные операции над синтезируемым текстом — выделение синтагм, выбор типа интонации, определение степени выделенности (ударности-безударности) гласных и символьного звукового наполнения слоговых комплексов осуществляются блоком автоматического транскриптора.

Временной процессор входят также правила, задающие длительность паузы после окончания синтагмы (конечной/неконечной), которые необходимы для синтеза связного текста. Предусмотрена также модификация общего темпа произнесения синтагмы и текста в целом, причем в двух вариантах: в стандартном — при равномерном изменении всех единиц компиляции — и в специальном, дающем возможность изменения длительности только гласных или только согласных.

Тональный процессор содержит правила формирования для одиннадцати интонационных моделей: нейтральная повествовательная интонация, типичная для фокусируемых ответов на вопросы; интонация предложений с контрастивным выделением отдельных слов; интонация специального и общего вопроса; интонация особых противопоставительных или сопоставительных вопросов; интонация обращений, некоторых типов восклицаний и команд; два вида незавершенности, перечислительная интонация; интонация вставочных конструкций.

Аллофонная база данных. Необходимый речевой материал записан в режиме оцифровки с частотой дискретизации 22 кГц с разрядностью 16 бит.

В качестве базовых элементов компиляции выбраны аллофоны, оптимальный набор которых и представляет собой акустико-фонетическую базу синтеза. Набор базовых единиц компиляции включает в себя 1200 элементов. В большинстве случаев элементы компиляции представляют собой сегменты речевой волны фонемной размерности. Для получения необходимой исходной базы единиц компиляции был

составлен специальный словарь, который содержит слова и словосочетания с аллофонами во всех учитываемых контекстах.

На основе данных, полученных от остальных модулей синтеза речи и от аллофонной базы, программа формирования акустического сигнала позволяет осуществлять модификацию длительности согласных и гласных. Она дает возможность модифицировать длительность отдельных периодов на вокальных звуках, используя две или три точки тонирования на аллофонном сегменте, осуществляет модификацию энергетических характеристик сегмента и соединяет модифицированные аллофоны в единую слитную речь.

4.4. Системы автоматизированного и автоматического перевода текстов

Перевод с одного языка на другой человеком происходит путем восприятия и понимания исходного текста и последующей передачи его смысла средствами выходного языка. При этом переводятся не слова и словосочетания, а понятийные образы, порождаемые в сознании переводчика под их воздействием. Однако если в настоящее время пока еще нет возможности моделировать работу человека-переводчика, то, по крайней мере, нужно стремиться оперировать теми единицами языка и речи, которые позволяют наиболее точно передавать содержание текста, написанного на одном языке, средствами другого языка. Такими единицами являются, прежде всего, фразеологические обороты и терминологические словосочетания и, во вторую очередь, отдельные слова. Если в настоящее время полностью автоматический высококачественный научно-технический перевод практически невозможен, то автоматизированный человеко-машинный перевод вполне реален.

4.4.1. Обобщенная технология работы системы машинного перевода

Процесс машинного перевода (МП) текстов с одного естественного языка на другой может быть в общем случае разделен на три этапа:

- семантико-синтаксический анализ;
- трансфер;
- семантико-синтаксический синтез.

Текст на входном языке поступает в систему перевода, на этапе семантико-синтаксического анализа выявляется его грамматическая структура, распознаются наименования понятий и устанавливаются отношения между понятиями.

На этапе трансфера производится переход от наименований понятий и структуры текста на входном языке к наименованиям понятий и структуре текста на выходном языке. В результате семантико-синтаксического синтеза на основании полученных эквивалентов получается текст на выходном языке (его грамматическое оформление), который выдается в качестве результата.

Действующие системы машинного перевода ориентированы на конкретные пары языков (например, французский и русский или японский и английский) и используют, как правило, переводные соответствия либо на поверхностном уровне, либо на некотором промежуточном уровне между входным и выходным языками. Качество машинного перевода зависит от объема словаря, объема информации, приписываемой лексическим единицам, от тщательности составления и проверки работы алгоритмов анализа и синтеза, от эффективности программного обеспечения. Информация может быть представлена как в декларативной (описательной), так и в процедурной (учитывающей потребности алгоритма) форме.

Машинный перевод следует отличать от автоматических словарей, помогающих человеку быстрее подбирать нужный переводной эквивалент. Хотя и в том, и в другом случае компьютер работает вместе с человеком (переводчиком или редактором), в содержание термина «машинный перевод» входит представление о том, что главную, большую часть работы по переводу и отысканию переводных эквивалентов и переводных соответствий машина берет на себя, оставляя человеку лишь контроль и исправление ошибок, в то время как компьютерный словарь в помощь человеку — это чисто вспомогательное средство.

4.4.2. Основные проблемы машинного перевода

Исторически *машинный перевод* является первой попыткой использования компьютеров для решения невычислительных задач (Джорджтаунский эксперимент в США в 1954 г.; работы по машинному переводу в СССР, начавшиеся в 1954 г.). Развитие электронной техники, рост объема памяти и производительности компьютеров

создавали иллюзию быстрого решения этой задачи. Практическая цель была простой: загрузить в память компьютера максимально возможный словарь и с его помощью из иноязычных текстов получать текст на родном языке в удобочитаемом виде. Однако первоначальная эйфория по поводу того, что столь трудоемкую работу можно поручить ЭВМ, сменилась разочарованием в связи с абсолютной непригодностью получаемых текстов.

Конечно, системы, настроенные на определенную предметную область, дают гораздо более приемлемые результаты. Однако и в этом случае системы перевода получаются очень узко ориентированными, а попытка использовать их даже в смежных предметных областях дает совершенно непредсказуемые результаты.

Возникают эти проблемы из-за принципиально разных подходов к переводу человека и машины. Квалифицированный переводчик понимает смысл текста и пересказывает его на другом языке словами и стилем, максимально близкими к оригиналу. Для компьютера этот путь выливается в решение двух задач:

- перевод текста в некоторое внутреннее семантическое представление;
- генерация по этому представлению текста на другом языке.

Поскольку не только не решена сама по себе ни одна из этих задач и даже нет общепринятой концепции семантического представления текстов, при автоматическом переводе приходится фактически делать «подстрочник», заменяя по отдельности слова одного языка на слова другого и пытаясь после этого придать получившемуся предложению некоторую синтаксическую согласованность. Смысл при этом может быть искажен или безвозвратно утерян.

4.4.3. Фразеологический машинный перевод

Концепция фразеологического перевода базируется на понимании того факта, что в естественных языках смысл лексических единиц более высокого уровня (например, фразеологических единиц, являющихся наименованиями понятий или ситуаций), как правило, несводим к смыслу составляющих их лексических единиц более низкого уровня (например, слов).

При решении проблемы перевода ранее делалась ставка прежде всего на грамматически правильный пословный перевод, а полисемия слов разрешалась в основном процедурными средствами на ос-

нове учета их синтаксических и семантических признаков. Поэтому системы МП первых трех десятилетий их развития можно охарактеризовать как системы семантико-синтаксического преимущественно пословного перевода. Словосочетания здесь также использовались, но в меньшей степени.

Семантико-синтаксический пословный машинный перевод текстов не имеет особой перспективы, так как в естественных языках смысл словосочетаний, как правило, несводим или не полностью сводим к смыслу составляющих их слов, и при переводе он не обязательно может быть «вычислен» на основе синтаксических и семантических признаков этих слов.

Принципы построения систем фразеологического машинного перевода текстов были впервые сформулированы Г.Г. Белоноговым в 1975 г. и изложены в 1983 г. в книге Г.Г. Белоногова и Б.А. Кузнецова «Языковые средства автоматизированных информационных систем». В 1984 г. аналогичная идея была высказана японским ученым профессором Нагао из университета Киото. Он предложил в качестве альтернативы подход, основанный на использовании ранее переведенных текстов, представленных одновременно на двух языках (билингв).

Важнейшими среди этих принципов являются следующие:

- основными единицами языка и речи, которые прежде всего следует включать в машинный словарь, должны быть фразеологические единицы (словосочетания, фразы). Отдельные слова также могут включаться в словарь, но они должны использоваться только в тех случаях, когда не удается осуществить перевод, опираясь только на фразеологические единицы;
- наряду с фразеологическими единицами, состоящими из непрерывных последовательностей слов, в системах машинного перевода следует использовать и так называемые речевые модели — фразеологические единицы-шаблоны с «пустыми местами», которые могут заполняться различными словами и словосочетаниями, порождая осмысленные отрезки речи;
- реальные тексты, независимо от их принадлежности к той или иной тематической области, обычно бывают политематическими, если они имеют достаточно большой объем. И отличаются они друг от друга не столько словарным составом, сколько распределениями вероятностей появления в них различных слов из общенационального словарного фонда. Поэтому машинный словарь, предназначенный для перевода текстов даже только из одной тематической об-

ласти, должен быть политематическим, а для перевода текстов из различных предметных областей — тем более;

- для систем фразеологического перевода необходимы машинные словари большого объема. Такие словари могут создаваться на основе автоматизированной обработки двуязычных текстов, являющихся литературными переводами друг друга;
- наряду с основным (политематическим) словарем большого объема в системах фразеологического машинного перевода целесообразно использовать также набор дополнительных тематических словарей. Дополнительные словари должны содержать только ту информацию, которая отсутствует в основном словаре (например, информацию о приоритетных переводных эквивалентах словосочетаний и слов для различных предметных областей, если эти эквиваленты не совпадают с приоритетными переводными эквивалентами основного словаря);
- основным средством разрешения полисемии (многозначности) слов в системах фразеологического перевода является их использование в составе фразеологических словосочетаний. Дополнительным — аппарат тематических словарей, где для каждого многозначного слова или словосочетания указывается его приоритетный переводной эквивалент, специфичный для рассматриваемой предметной области;
- большую роль в системах фразеологического машинного перевода текстов могут играть процедуры морфологического и синтаксического анализа и синтеза, построенные на основе принципа аналогии. Эти процедуры позволяют отказаться от хранения в словарях большого объема грамматической информации и порождать ее по мере необходимости автоматически, в процессе перевода. Они делают систему перевода открытой — способной обрабатывать тексты с «новой» лексикой;
- наряду с переводом текстов в автоматическом режиме в системах фразеологического машинного перевода целесообразно предусмотреть интерактивный режим работы. В этом режиме пользователь должен иметь возможность вмешиваться в процесс перевода и настраивать дополнительные машинные словари на тематику переводимых текстов.

В соответствии с главным тезисом концепции фразеологического перевода система фразеологического машинного перевода должна

включать в свой состав базу знаний, содержащую переводные эквиваленты для наиболее часто встречающихся фраз, фразеологических сочетаний и отдельных слов и программные средства для морфологического и синтаксического анализа и синтеза текстов и для их редактирования человеком.

В процессе перевода текстов система должна использовать хранящиеся в ее базе знаний переводные эквиваленты в следующем порядке: сначала для очередного предложения исходного текста делается попытка перевести его как целостную фразеологическую единицу; затем, в случае неудачи, — входящие в его состав словосочетания; и, наконец, осуществляется пословный перевод тех фрагментов текста, которые не удалось перевести первыми двумя способами. Фрагменты выходного текста, полученные всеми тремя способами, должны грамматически согласовываться друг с другом (с помощью процедур морфологического и синтаксического синтеза).

Контрольные вопросы

1. Перечислите основные принципы распознавания символов (OCR).
2. Что такое OCR A и OCR B?
3. В чем заключается содержание метода сопоставления с образцом?
4. Перечислите основные особенности метода POWR.
5. Каковы возможности программного продукта Finereader?
6. Что такое принципы IPA?
7. В чем заключается MDA?
8. Что такое бинаризация изображения?
9. Какие типы классификаторов-распознавателей вам известны?
10. Перечислите основные принципы систем распознавания речи (STT).
11. Охарактеризуйте программные продукты STT.
12. Перечислите основные принципы систем генерации речи (TTS).
13. Охарактеризуйте программные продукты TTS.
14. Назовите основные принципы систем автоматизированного перевода.
15. Что такое фразеологический машинный перевод?
16. В чем заключается интеграция систем перевода и обработки речи?

Глава 5
ТЕХНОЛОГИИ ДОСТУПА К ДАННЫМ. ФАЙЛОВЫЕ СИСТЕМЫ И СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ (СУБД)

Реальные хранилища данных содержат миллионы записей. Требования, предъявляемые пользователями к информационным системам, обрабатывающим эти данные, в первую очередь предполагают высокую оперативность доступа. Важной особенностью является и то, что архитектура систем и технологий управления данными связана с двумя следующими значительными и разноплановыми обстоятельствами: с одной стороны, с непредсказуемой вариантностью представления данных в прикладной программе, зависящей от особенностей пользовательских задач, с другой — с жесткостью технических решений устройств внешней памяти, выражающейся в функциональной простоте[1] операций и ограниченности форм представления данных.

В общем случае высокая эффективность решений в области обработки данных достигается введением промежуточных слоев специализированных технических и программных средств.

---

[1] Требование операционной простоты определяется производственными и экономическими причинами: устройство должно быть надежным в использовании и дешевым в изготовлении. Функциональная ограниченность управления данными, кроме того, диктуется еще и требованием *унифицированности*: устройство должно одинаково эффективно и стандартным способом использоваться в составе различных вычислительных и операционных систем, даже если со временем отдельные компоненты систем будут меняться.

5.1. Организация данных на машинных носителях

С общепринятой точки зрения к вопросам организации данных относятся:

- выбор типа записи — единицы обмена в операциях ввода-вывода;
- выбор способа размещения записей в файле и, возможно, метода оптимизации размещения;
- выбор способа адресации и метода доступа к записям.

Целесообразность выделения именно таких аспектов организации была предельно очевидна на начальной стадии развития таких запись-ориентированных систем и устройств внешней памяти, как магнитные ленты и диски. Но следует отметить, что широкое использование современных поток-ориентированных систем ввода-вывода не уменьшило принципиальное, да и практическое значение давно известных методов и решений, построенных на запись-ориентированных принципах.

5.1.1. Типы записей

Основные понятия и подходы к физической организации и обработке данных, кратко обсуждаемые ниже, иллюстрируются рис. 5.1.

Логическая запись, с которой работает прикладная программа, — это совокупность элементов или агрегатов данных, воспринимаемая и, обычно, физически отдельно размещаемая прикладной программой в рабочей области памяти как единое целое. Последовательность записей с точки зрения *логики обработки* образует *файл*.

Физическая запись, с которой работает файловая система, — это совокупность данных, которые размещаются в файле обычно на внешнем носителе, и могут быть считаны или записаны как единое целое одной *командой ввода-вывода*. Файл[1] — это последовательность физических записей, размещаемых в линейном пространстве носителя, но, в общем случае, не обязательно в линейном порядке.

Организация данных в случаях логического и физического представления может не совпадать. В частности, одна физическая запись может включать несколько логических (*блокирование записей*). При этом алгоритмы выделения логических записей из физической в зна-

[1] В некоторых операционных системах, например IBM, файл на внешних носителях называют *набором данных* в отличие от логического файла.

Рис. 5.1. Способы организации файлов данных

чительной степени зависят от *типа записи*, рассматриваемого как характер организации последовательности байтов.

На логическом уровне выделяют следующие типы записей:

- *записи фиксированной длины*, для размещения каждой из которых выделяется память фиксированной длины, объявляемой заранее. В этом случае данные, образующие запись, имеют устойчивую природу и представляются жесткими структурами, например, ряд числовых полей или символьная последовательность заданной длины;
- *записи переменной длины*, когда каждый экземпляр записи может иметь длину, отличную от длины другой записи в том же наборе. В этом случае запись содержит либо элементы данных переменной длины (например, текстовую строку), либо переменное число элементов фиксированной длины.

Организация физической записи для случая блокирования логических записей фиксированной или переменной длины представлена на рис. 5.2.

При этом структура представления логической записи *переменной длины*[1] отличается тем, что байтам содержания — собственно данным,

[1] В современных файловых системах практически не используется.

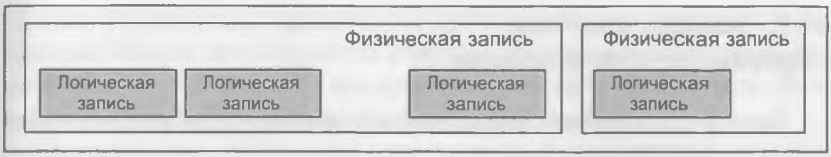


Рис. 5.2. Физическая организация логических записей

образующим логическую запись, предшествуют байты значения длины содержания этой логической записи.

Существует и другая физическая структура представления записей, имеющих переменную длину, — *запись неопределенной длины*, когда данные, образующие логическую запись, завершаются разделителем «конец записи»[1].

Порядок доступа к записи может быть только последовательным, поскольку для определения начала следующей записи надо считать значение длины текущей.

Для файлов записей фиксированной длины доступ будет проще, так как адрес начала любой записи может быть вычислен умножением относительного номера нужной записи на длину записи.

Физические записи на носителе следуют непосредственно друг за другом. При этом выделение отдельной записи может производиться двумя способами, определяемыми технологиями записи данных на носитель.

Первый способ, применяемый в запись-ориентированных устройствах внешней памяти мэйнфреймов, основан на том, что каждая запись отделяется от соседней физическим промежутком, неиспользуемым для записи и воспринимаемым устройством чтения как сигнал «конец записи».

Другой способ — это размещение байтов следующей записи непосредственно за последним байтом предыдущей записи без каких-либо разделителей. Для этого способа характерна меньшая зависимость от особенностей устройства: оптимизация процессов ввода-вывода, в том числе блокирование[2] записей, переносится в прикладную программу.

---

[1] В поток-ориентированных файловых системах этому соответствует организация текстовых файлов, где запись — это последовательность символов, образующих строку, которая завершается специальными кодами «CR», «LF».

[2] Блокирование записей переменной и неопределенной длины в этом случае будет уже нецелесообразно.

5.1.2. Способы адресации и методы доступа к записям

Записи логического файла могут идентифицироваться с помощью уникальной последовательности символов или некоторого числа — *ключа*. Таким ключом обычно является значение поля, расположенное в каждой записи в одной и той же позиции. Иногда бывает необходимо объединить несколько полей, чтобы обеспечить уникальность ключа, который в этом случае называется *сцепленным ключом*.

В некоторых файлах записи имеют несколько ключей. Запись ЗАКУПКА может иметь различные НОМЕР ПОСТАВЩИКА и НОМЕР ПОКУПАТЕЛЯ, каждый из которых является ключом.

Во многих приложениях требуется идентифицировать записи по ключам, которые не являются уникальными. Однако при этом все равно должен существовать *один* уникальный ключ, тот, который используется для размещения записи в файле и выборки ее из файла. Такой ключ называется *первичным* ключом или *идентификатором*.

Основную проблему при адресации файла можно сформулировать следующим образом: *как по первичному ключу определить местоположение записи с данным ключом?* и *как надо организовать набор записей, чтобы поиск потребовал как можно меньше затрат?*

При разработке схем адресации файлов и определяемого ими размещения записей в файлах большое значение имеет вопрос о том, как включаются в файл новые записи и удаляются старые.

Существует несколько различных способов адресации и поиска записей, например, на основе упорядочения, различных индексов, преобразования «ключ-адрес». Приведем обзор следующих способов, количественная оценка эффективности которых представлена в [Мартин].

Последовательное сканирование файла. Наиболее простым способом локализации записи является сканирование файла с проверкой ключа каждой записи. Этот способ при больших объемах требует слишком много времени, и его целесообразно применять, когда каждая запись все равно должна быть прочитана.

Блочный поиск. Если записи упорядочены по значению ключа, то при сканировании файла не требуется чтение каждой записи. ЭВМ могла бы, например, просматривать каждую сотую запись в последовательности возрастания ключей. При нахождении записи с ключом большим, чем искомое значение, просматриваются последние 99 за-

писей, которые были пропущены. Этот способ называется *блочным поиском*. Записи группируются в блоки, и каждый блок проверяется по одному разу до тех пор, пока не будет найден нужный блок. Иногда данный способ называют *поиском с пропусками*.

Двоичный поиск. При двоичном поиске в файле записей, упорядоченных по ключу, анализируется запись, находящаяся в середине поисковой области файла (изначально всего файла), а ее ключ сравнивается с поисковым ключом. Затем поисковая область делится пополам, и процесс повторяется для соответствующей половины области, пока не будет обнаружено искомое значение или длина области не станет равной 1. Число сравнений в этом случае будет меньше, чем для случая блочного поиска.

Индексно-последовательные файлы. Если файл упорядочен по ключам, то для адресации может использоваться *индекс*, связывающий ключ хранимой записи с ее относительным или абсолютным адресом во внешней памяти.

Если записи файла упорядочены по ключу, индекс обычно содержит не ссылки на каждую запись, а ссылки на блоки записей, внутри которых можно выполнять двоичный поиск или сканирование. Хранение ссылок на блоки записей, а не на отдельные записи, в значительной степени уменьшает размер индекса. Причем даже в этом случае индекс часто оказывается слишком большим для поиска и поэтому используется индекс индекса.

Индексно-произвольные файлы. Произвольный (не упорядоченный по ключу) файл можно индексировать точно так же, как и последовательный файл. Однако при этом индекс должен содержать по одному элементу для каждой записи файла, а не для блока записей. Более того, в нем должны содержаться *полные* абсолютные (или относительные) адреса, в то время как в индексе последовательного файла адреса могут содержаться в усеченном виде, так как старшие знаки последовательных адресов будут совпадать.

Произвольные файлы в основном используются для обеспечения возможности адресации записей файла с несколькими ключами. Если файл упорядочен по одному ключу, то он не упорядочен по другому ключу. Для каждого типа ключей может существовать свой индекс: для упорядоченных ключей индекс будет иметь по одному элементу на блок записей, для других ключей индексы будут более длинными, так как должны будут содержать по одному элементу для каждой записи. Ключ, который чаще всего используется при адресации файла, обычно служит для его упорядочения.

В индексно-произвольных файлах часто используются адреса, построенные на абсолютных значениях, так как при добавлении новых или удалении старых записей изменяется местоположение записей. Если в записях имеется несколько ключей, то индекс вторичного ключа может содержать в качестве выхода первичный ключ записи. При определении же местоположения записи по ее первичному ключу можно использовать какой-нибудь другой способ адресации. По этому методу поиск осуществляется медленнее, чем поиск, при котором физический адрес записи определяется по индексу. В файлах, в которых положение записей часто изменяется, символическая адресация может оказаться предпочтительнее.

Адресация с помощью ключей, преобразуемых в адрес. Известно много методов преобразования ключа непосредственно в значение адреса в файле. В тех случаях, когда возможно преобразование значения ключа непосредственно в значение адреса в файле, такой способ адресации обеспечивает самый быстрый доступ; при этом нет необходимости организовывать поиск внутри файла или выполнять операции с индексами. В некоторых приложениях адрес может быть вычислен на основе значений некоторых элементов данных записи.

К недостаткам данного способа относится малое заполнение файла: в файле остаются свободные участки, поскольку ключи преобразуются не в непрерывное множество адресов.

Другим недостатком схем прямой адресации является их малая гибкость. Машинные адреса записей могут измениться при обновлении файла. Для устранения этого недостатка прямую адресацию обычно выполняют в два этапа. Сначала ключ преобразуется в *порядковый номер,* который затем преобразуется в машинный адрес.

Хэширование. Простым и полезным способом вычисления адреса является хэширование (*перемешивание*). В данном методе ключ преобразуется в квазислучайное число, которое используется для определения местоположения записи.

Более экономичным является указание на область, в которой размещается группа записей. Эта область называется *участком записей* (slot, bucket). При первоначальной загрузке файла адрес, по которому должна быть размещена запись, определяется следующим образом.

1. Ключ записи преобразуется в квазислучайное число, находящееся в диапазоне от 1 до числа участков, используемых для размещения записей.

2. Число преобразуется в адрес участка, и если на участке есть свободное место, то логическая запись размещается на нем.

3. Если участок заполнен, запись должна быть размещена на *участке переполнения* — следующем по порядку участке либо участке в отдельной *области переполнения*.

При чтении записей из файла их поиск выполняется аналогично, причем может оказаться, что для поиска записи потребуется чтение нескольких участков переполнения.

Из-за вероятностной природы алгоритма в этом способе не удается достичь 100%-ной плотности заполнения памяти, однако для большинства файлов может быть достигнута плотность 80 или 90 %; при этом память для индексов не требуется. Большинство записей можно найти за одно обращение, но для некоторых потребуется второе обращение (при переполнении), и очень редко потребуется третье или четвертое обращение к файлу. Кроме того, в этом случае менее эффективно используется память, чем в индексных методах; записи не упорядочены для последовательной обработки.

Комбинации способов адресации. При адресации записей некоторых файлов используются комбинации перечисленных выше способов. Например, с помощью индекса может определяться ограниченная поисковая область файла, затем эта область просматривается последовательно либо в ней выполняется двоичный поиск. С помощью алгоритма прямой адресации может определяться нужный раздел индекса, и, таким образом, исчезает необходимость поиска во всем индексе.

5.1.3. Схема адресации и организация набора данных

Схема адресации записей в файле является определяющей для выбора способов размещения записей в файлах с точки зрения процедур включения в файл новых записей, обновления и удаления старых.

Записи располагаются на внешнем запоминающем устройстве в конкретной физической последовательности. Обработка данных усложняется, если последовательность записей файла не соответствует последовательности их обработки: возникает необходимость сортировки записей, что требует значительных временных затрат. Как отмечалось ранее, другой способ — это организация доступа к записям в нужной последовательности, отличной от порядка физического размещения, — использование различных систем адресации.

Можно выделить следующие способы включения в файл новых записей.

1. При включении новых записей файл перезаписывается с размещением записей в соответствии со значением ключа.

2. Записи размещаются в области переполнения, которая находится либо в той же области (файле), что и основная область, либо — в отдельной независимой области (файле). При этом для обеспечения доступа могут использоваться цепочки, указатели из индекса к каждой записи переполнения, отдельные индексы для каждого блока области переполнения.

3. Записи размещаются в распределенной свободной памяти, которая резервируется при создании базы на уровне физических или логических областей в пространстве файла. При переполнении первоначально зарезервированной области производится ее расщепление.

Для иллюстрации взаимосвязи схем адресации и организации наборов данных рассмотрим структуру индексно-последовательной ор-

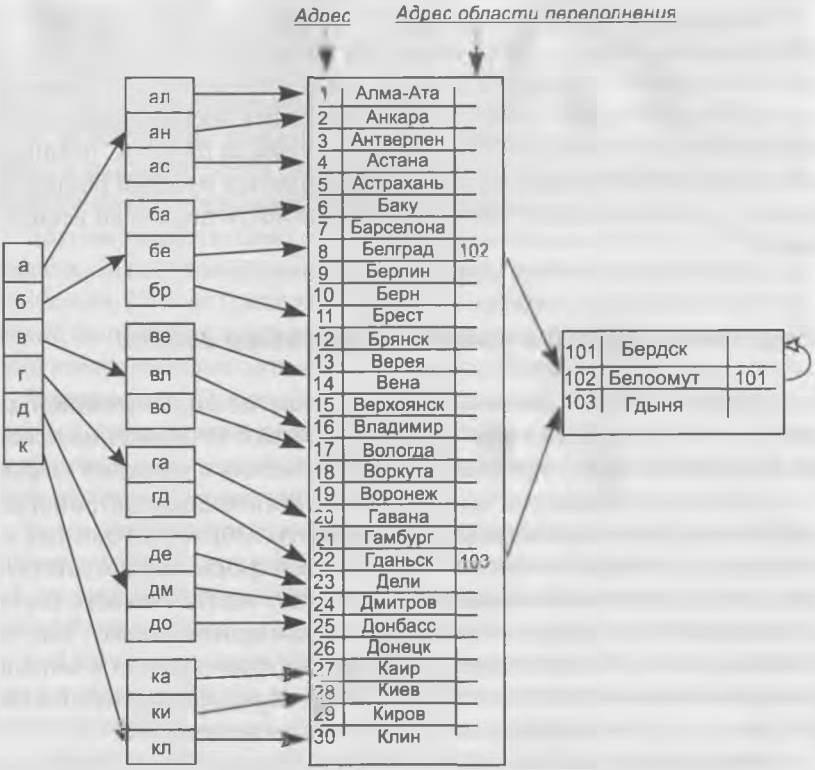


Рис. 5.3. Схема индексно-последовательного файла после добавления трех записей

ганизации, приведенную на рис. 5.3, файла, в который добавлены 3 новые записи.

При индексно-последовательной организации записи физически размещаются последовательно в соответствии с возрастанием их ключей, которые чаще всего используются для адресации этих записей.

Новые записи включаются в конец файла или в область переполнения. При этом для *адресации записей в области переполнения* применяются указатели, расположенные в индексах и указывающие на записи, содержащиеся в области переполнения, или в самой области переполнения используются указатели, связывающие записи в цепочки в порядке возрастания ключа.

5.1.4. Способы размещения записей

Записи файла обычно располагаются на носителе последовательно в том порядке, как они создаются в прикладной программе. Но иногда физическая последовательность размещения записей может отличаться от их логической последовательности.

Последовательность размещения физических записей естественно может быть только одна (если содержание логической записи сознательно не дублируется в другой форме), и она должна быть выбрана с учетом эффективности использования данных в различных приложениях. Выбор связывается с одним из следующих обстоятельств.

1. Ускорение выполнения наиболее частых операций путем размещения записей в той последовательности, которая требуется при последующей обработке.

2. Ускорение или упрощение средств адресации файла (например, средств прямой адресации или хэширования).

3. Уменьшение размера используемого индекса и сокращение таким образом времени поиска в нем.

4. Сокращение среднего времени доступа за счет размещения в наиболее доступных местах записей, к которым происходит наиболее частое обращение.

5. Облегчение операций включения, обновления и удаления записей в интенсивно изменяемых файлах.

Можно выделить две «чистые» стратегии определения места (адреса) для размещения записей: *последовательное* (sequential) и *произвольное* (random) размещение. В этом смысле алгоритм размещения определяет *тип организации* файла.

В первом случае каждая следующая запись будет располагаться физически следом за предыдущей. Во втором — по месту, адрес которого будет определяться в зависимости от некоторых факторов, в том числе упомянутых выше.

Хотя записи на устройствах с прямым доступом могут записываться и читаться в любой последовательности, для каждой структуры данных существует некоторая определенная последовательность, в которой записи можно читать намного быстрее, чем при других способах размещения.

Рассмотрим следующие, наиболее распространенные методы организации файлов, схематично представленные на рис. 5.4, позволяющие оптимизировать доступ к записям.

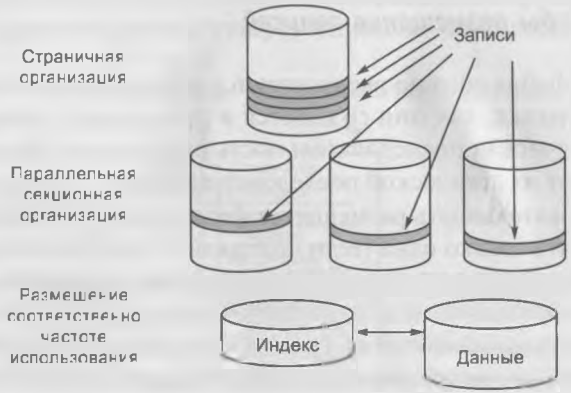


Рис. 5.4. Способы организации файлов

Страничная организация. Данные можно перемещать между внешней и оперативной памятью страницами фиксированной длины. Размер страницы определяется системой (без учета длины записи). Там, где применяется страничная организация памяти, данные логически независимы от размера страницы, но они должны быть физически сгруппированы так, чтобы эффективно заполнять страницы.

Параллельная секционная организация. Если имеется несколько механизмов доступа, которые могут работать одновременно, то для минимизации времени ожидания данные могут быть расположены на запоминающих устройствах так, чтобы одновременно было задействовано как можно большее число механизмов доступа.

При параллельной секционной организации существуют два вида ожиданий. Запросы должны ожидать позиционирования механизма

доступа (операция установки и задержки на вращение), а затем — ждать выполнения операции чтения-записи. Время, в течение которого запись читается, значительно меньше времени, в течение которого позиционируется механизм доступа. Следовательно, полное время доступа к записи при параллельной организации будет меньше.

Размещение соответственно частоте использования. Если в системах используется несколько типов запоминающих устройств или в системе предусмотрены специальные методы доступа, то наиболее часто используемые данные можно хранить на более быстрых устройствах или в файлах с «быстрым» методом доступа.

Аналогичный принцип используется при «кэшировании», когда наиболее часто используемые записи помещаются в промежуточную память с быстрым доступом, обеспечивающимся в основном программными средствами за счет упорядочения размещения и введения избыточности.

5.2. Файловые системы

5.2.1. Схема организации файлового ввода-вывода

Рассмотрим представленные на рис. 5.5 основные способы адресации и последовательность операций[1] выборки данных, обеспечивающих чтение прикладной программой с тома внешней памяти (например, магнитного диска) некоторой записи с номером *I*. Отметим еще раз, что «специализация» компонентов, участвующих в операциях ввода-вывода, выражается прежде всего в используемом способе адресации.

Прикладная программа использует одномерную (или сводимую к одномерной) сквозную адресацию данных на уровне логических записей: запись определяется номером, соответствующим, например, порядку ее размещения.

Система управления физическим вводом-выводом (в рассматриваемом примере — BIOS ПЭВМ) использует трехмерную систему координат: адрес записи составляется из номера дорожки, номера головки чтения-записи (номер поверхности) и номера сектора. То есть операционная система будет использовать следующую одномерную

[1] В целях общности в этом примере не рассматриваются подготовительные операции, такие как открытие файла и выделение памяти для рабочих и системных буферов, хотя они также достаточно ресурсоемки.

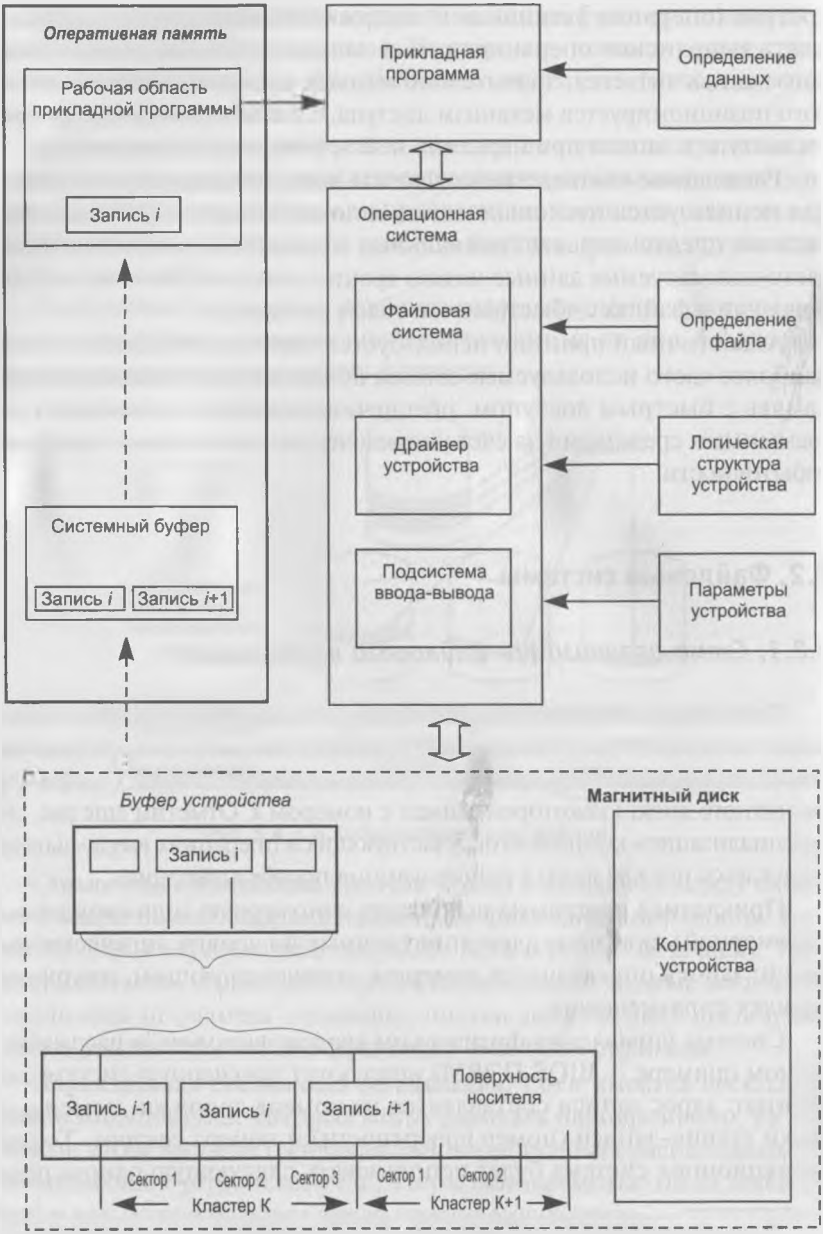


Рис. 5.5. Примерная схема организации ввода-вывода

сквозную «систему координат»: сектора нумеруются от края диска к центру последовательно, причем сначала в рамках одного сегмента цилиндра (кластера), далее сектора следующего сегмента дорожки, после чего происходит переход к следующей дорожке.

Этот способ адресации и, соответственно, порядок использования пространства отчасти отражает специфику аппаратных решений, ориентированных на временную оптимизацию операций ввода-вывода: большее количество данных будет считано при одном обращении к диску за счет одновременного обращения через головки чтения-записи к данным, размещенным на параллельных дорожках в одном секторе одного цилиндра. Фиксированное количество битов, равное размеру сектора и определенное при разметке, умноженное на число головок, будет прямо (без дополнительной обработки, например, проверки логических условий конца файла или записи) передано в буфер оперативной памяти устройства или операционной системы.

Таким образом, если система адресации в прикладной программе является относительной и отражает логику взаимосвязи записей (например, порядок создания файла), то для подсистем ввода-вывода она является абсолютной и определяется *физическим форматом носителя*: размером сектора, количеством секторов на дорожке, количеством поверхностей и дорожек и т. д. При этом независимость от особенностей физического размещения и механизма адресации обеспечивается на уровне *логической структуры носителя*. Например, логически последовательная выборка записей файла обеспечивается таблицей размещения файлов, определяющей используемое файлом пространство как цепочку кластеров, физически находящихся в любой доступной части диска. Доступ к файлу производится по идентификатору (составному имени) через систему каталогов, связывающих идентификатор файла с началом цепочки указателей на кластеры данных в таблице размещения файлов. Кроме того, логическая структура содержит (в составе загрузочной записи) информацию, идентифицирующую пространство в целом, а также данные, *определяющие физическую структуру*.

В общем случае операция чтения физической записи включает следующие действия.

1. Определение адреса записи в координатах устройства (например, для файлов с записями фиксированной длины — пересчетом номера нужной записи в относительный адрес сектора, и далее определение абсолютного номера сектора на диске).

2. Перемещение головки чтения в соответствующую координату: позиционирование к дорожке и сектору на дорожке, складывающееся

из двух действий — собственно радиального перемещения головки на расстояние от текущего положения до нужной дорожки и ожидания подхода указанного сектора вращающегося диска к позиции, где находится головка. Следует также отметить, что высокая плотность записи данных означает, что промежуток между секторами[1] и дорожками сравнительно мал (сопоставим с погрешностями механизма перемещения и тепловым расширением), и поэтому правильность позиционирования определяется по служебным данным заголовка[2] сектора, считываемым до начала передачи прикладных данных.

3. Пересылка данных, расположенных в области кластера, в буфер, который физически может быть как частью устройства, так и областью оперативной памяти.

4. Завершение операции (проверка корректности чтения) и возврат управления ОС.

5. Выделение системой данных, относящихся к затребованным записям. Причем во многих случаях в системный буфер считываются не только данные логической записи, нужные прикладной программе, но и соседние. Это позволяет сократить суммарные затраты времени при чтении нескольких записей, исключив наиболее долгую операцию позиционирования. Указание на такое *блокирование* может выдаваться явно прикладной программой при открытии файла или операционной системой, использующей собственные механизмы кэширования для оптимизации[3] ввода-вывода.

---

[1] Если контроллер не успевает завершить обработку передачи и подготовиться к передаче данных, размещаемых на физически следующем секторе, то придется ожидать завершения полного оборота диска. С целью исключения таких потерь диск форматируется так, что логически последовательные секторы разделены одним или несколькими физическими секторами (коэффициент чередования) так, что контроллер будет готов выполнить операцию со следующим логическим сектором, не ожидая дополнительного оборота.

[2] Такой подход форматирования (разметки) пространства внешней памяти используется и в случае таких устройств «истинно» последовательного доступа, как магнитные ленты, для обеспечения ускоренного «прямого» доступа к сектору по его номеру — прямому адресу (с тех времен, когда еще не были созданы дисковые накопители, например, ЭВМ 2-го поколения Минск-22). При этом, поскольку данные секторов, предшествующих нужному, передавать не надо, позиционирование будет выполняться с максимальной скоростью (перемотки ленты).

[3] Автоматическое использование системы кэширования и упреждающего чтения (не учитывающее особенности порядка обращения к данным, обусловленного алгоритмом обработки) может привести к обратному результату, например, в случае обращения к логическим записям в произвольной последовательности (случайной), не соответствующей физическому следованию записей.

6. Передача в рабочую область прикладной программы данных запрошенной ею логической записи или указателя на соответствующую область памяти в системном буфере.

В этой последовательности наиболее медленными операциями являются механическое позиционирование головок и чтение данных с поверхности носителя (выполняемые на порядки медленнее, чем операции пересылки). Поэтому выигрыш во времени может быть получен только в случае выполнения ряда запросов на доступ к данным, причем экономия может достигаться следующими путями:

- суммарным сокращением перемещения головок за счет организации такой последовательности обращения к записям (или такого порядка их физического размещения), когда перемещение от текущего положения к следующему будет минимальным;
- формированием логических записей таким образом, чтобы их формат соответствовал физическому формату хранения; в случае кратности длин, т. е. если длина логической записи будет кратной длине кластера или в кластере будет размещаться целое число записей, будет исключена передача данных, не запрошенных текущей операцией.

5.2.2. Основные требования к файловым системам

С появлением магнитных дисков по существу и началась история систем управления данными. До этого разработчик каждой прикладной программы, которой требовалось хранить данные во внешней памяти, сам определял расположение каждой порции данных на магнитной ленте или барабане и планировал обмены между оперативной памятью и устройствами внешней памяти. Историческим шагом явился переход к использованию централизованных систем управления файлами. С точки зрения прикладной программы файл — это именованная логически непрерывная область внешней памяти, в которую можно записывать и из которой можно считывать данные. Правила именования файлов, способ доступа к данным, хранящимся в файле, и структура этих данных зависят от конкретной системы управления файлами, и, возможно, от типа файла. Система управления файлами или файловая система (ФС), берет на себя распределение внешней памяти, отображение имен файлов в соответствующие адреса во внешней памяти и обеспечение доступа к данным.

Идентификация файлов. Практически все современные файловые системы поддерживают многоуровневое именование файлов за счет ведения во внешней памяти дополнительных файлов со специальной структурой — каталогов. Каждый каталог содержит имена каталогов (подкаталогов, папок) и/или файлов, содержащихся в данном каталоге. Таким образом, полное имя файла состоит из последовательности имен каталогов плюс имя файла в каталоге, непосредственно содержащем данный файл.

Защита файлов. Поскольку файловые системы являются общим хранилищем файлов, принадлежащих, вообще говоря, разным пользователям, системы управления файлами должны обеспечивать авторизацию доступа к файлам. В общем виде подход состоит в том, что по отношению к каждому зарегистрированному пользователю данной вычислительной системы для каждого существующего файла указываются действия, которые разрешены или запрещены данному пользователю.

В большинстве современных систем управления файлами применяется подход к защите файлов, впервые реализованный в ОС UNIX: каждому зарегистрированному пользователю соответствует пара целочисленных идентификаторов — идентификатор группы, к которой относится этот пользователь, и его собственный идентификатор в группе. При каждом файле хранится полный идентификатор пользователя, который создал этот файл, и фиксируется, какие действия с файлом может производить его создатель, какие действия с файлом доступны для других пользователей той же группы и что могут делать с файлом пользователи других групп.

Режим многопользовательского доступа. Если операционная система поддерживает многопользовательский режим, вполне реальна ситуация, когда два или более пользователя одновременно пытаются работать с одним и тем же файлом. Если все пользователи собираются только читать файл, конфликтная ситуация не возникнет. Но если хотя бы один из них будет изменять файл, для корректной работы этих пользователей требуется взаимная синхронизация.

В этом случае обычно применяется следующий подход. В операции открытия файла (первой и обязательной операции, с которой должен начинаться сеанс работы с файлом) среди прочих параметров указывался режим работы (чтение или изменение). Если к моменту выполнения этой операции от имени некоторого пользовательского процесса *A* файл уже находился в открытом состоянии от имени некоторого другого процесса *B*, причем файл был открыт в режиме, ко-

торый несовместим с желаемым режимом открытия (совместимы только режимы чтения), то в зависимости от особенностей системы процессу *A* либо сообщалось о невозможности открытия файла в желаемом режиме, либо он блокировался до тех пор, пока в процессе *B* не выполнялась операция закрытия файла.

5.2.3. Структура файловой системы (на примере NTFS)

Как и многие другие системы, NTFS делит все полезное пространство диска на кластеры (размеры кластеров — от 512 байт до 64 Кбайт).

Диск NTFS делится на две части. Первые 12 % диска отводятся под так называемую MFT-зону — пространство, в котором размещен метафайл MFT (Master File Table). Запись каких-либо данных в эту область невозможна — это делается для того, чтобы файл MFT не фрагментировался при обновлении. Остальное пространство диска представляет собой пространство для размещения файлов.

Каждый элемент файловой системы NTFS представляет собой файл (в том числе и служебная информация). MFT — таблица файлов, представляет собой централизованный каталог, включающий записи фиксированного размера (обычно 1 Кбайт), причем каждая запись соответствует одному файлу.

Все файлы на томе NTFS идентифицируются номером файла, который определяется позицией файла в MFT. Имена файлов NTFS могут быть длиной до 255 16-битовых символов UNICODE. Для генерации короткого имени файла в стиле MS-DOS (формат «8.3») NTFS удаляет все запрещенные символы, точки, а также пробелы из длинного имени файла. Далее имя файла усекается до 6 символов, добавляется тильда (~) и номер (всего восемь символов). Расширение имени файла усекается до трех символов.

Каждый файл и каталог на томе NTFS определяются набором атрибутов. Каждый *атрибут* файла NTFS представлен следующим набором полей: тип атрибута, длина атрибута, значение атрибута и, возможно, имя атрибута. Имеется *системный набор атрибутов*, определяемых структурой тома NTFS. Системные атрибуты имеют фиксированные имена и коды их типа (табл. 5.1). Могут применяться также атрибуты, определяемые пользователями. Их имена и типы задаются исключительно пользователем.

Таблица 5.1. **Список системных атрибутов NTFS**

№	Системный атрибут	Пояснение
1	Attribute List	Определяет список атрибутов, которые являются допустимыми для данного конкретного файла
2	File Name	Содержит длинное имя файла, а также номер вхо да в таблице MFT для родительского каталога
3	MS-DOS Name	Имя файла в формате «8.3»
4	Version	Атрибут содержит номер последней версии файла
5	Security Descriptor	Информация о защите файла: список прав доступа ACL и поле аудита, которое определяет, какого рода операции над этим файлом нужно регистрировать
6	Volume Version	Версия тома, используется только в системных файлах тома
7	Volume Name	Метка тома
8	Volume Information	Номер версии NTFS
9	Data	Содержит собственно данные файла
10	MFT bitmap	Содержит карту использования секторов на томе
11	Index Root	Корень B-дерева, используемого для поиска файлов в каталоге
12	Index Allocation	Нерезидентные части индексного списка B-дерева
13	External Attribute Information	Номер первого кластера и количество кластеров нерезидентного атрибута
14	Standard Information	Хранит информацию о файле, которую трудно связать с каким-либо из других атрибутов файла, например, время создания файла, время обновления и др.

Размещение файлов может иметь различные схемы в зависимости от размера файла.

Небольшой файл (small) имеет размер, позволяющий ему целиком располагаться внутри одной записи MFT (так называемый «резидентный»).

Большой файл (large), не вмещающийся в одну запись MFT, в значении атрибута «данные» содержит признак того, что файл является нерезидентным (т. е. находится вне MFT), а также номера первых кластеров каждого фрагмента данных и, соответственно, количество кластеров в каждом фрагменте.

Очень большой файл (huge), т. е. атрибут «данные» не помещается в одной записи и становится нерезидентным, т. е. размещается в другой записи MFT, ссылка на которую помещена в исходной записи о файле (внешний атрибут). Нерезидентный атрибут содержит указатели на фрагменты данных.

Сверхбольшой файл (extremely huge), т. е. внешний атрибут может указывать на несколько нерезидентных атрибутов. Кроме того, внешний атрибут, как и любой другой атрибут, может храниться в нерезидентной форме, поэтому в NTFS не может быть атрибутов слишком большой длины, которые система не сможет обработать.

Каталоги. Каждый каталог NTFS представляет собой один вход в таблицу MFT, который содержит список файлов, называемый индексом (index). Индексы позволяют сортировать файлы для ускорения поиска, основанного на значении определенного атрибута. Списки файлов также могут иметь различные схемы в зависимости от их размера.

Небольшие списки файлов (small indexes). Если количество файлов в каталоге невелико, то список файлов может быть резидентным в записи в MFT.

Большие списки файлов (large index). По мере того как каталог растет, список файлов может потребовать нерезидентной формы хранения. Однако начальная часть списка всегда остается резидентной в корневой записи каталога в таблице MFT. Имена файлов резидентной части списка являются узлами B-дерева. Остальные части списка файлов размещаются вне MFT. Для их поиска используется специальный атрибут «размещение списка» (Index Allocation), представляющий собой набор номеров кластеров, которые указывают на остальные части списка. Одни части списков являются *листьями дерева*, а другие — *промежуточными узлами*, т. е. содержат наряду с именами файлов атрибут Index Allocation, указывающий на списки файлов более низких уровней.

Надежность NTFS. NTFS является восстанавливаемой (recoverable) файловой системой. Журналирование — средство, позволяющее восстанавливать полностью корректное состояние при сбоях за счет использования стандартной процедуры регистрации транзакций[1] (каждая операция ввода-вывода, которая изменяет файл на томе NTFS). При модификации файла специальный компонент ФС — сер-

[1] Транзакция — действие, совершаемое целиком и корректно или не совершаемое вообще.

вис регистрации файлов (Log File Service) — фиксирует всю информацию, необходимую для повторения (redo) или отката (undo) транзакции в специальном файле с именем $LogFile. Если транзакция нормально не завершается, то NTFS пытается повторить транзакцию или производит ее откат.

NTFS поддерживает «горячее» переназначение секторов, когда при возникновении ошибки записи из-за плохого сектора данные переписываются в другой сектор, а сбойный исключается из работы.

Сжатие. Любой файл или каталог в индивидуальном порядке может храниться на диске в сжатом виде — этот процесс прозрачен для приложений. Сжатие файлов осуществляется с высокой скоростью, однако при этом часто возникает отрицательный эффект — фрагментация сжатых файлов.

Hard Link. Один и тот же файл в NTFS может иметь более одного имени (несколько указателей файла-каталога или разных каталогов ссылаются на одну и ту же MFT-запись). Допустим, один и тот же файл имеет имена 1.txt и 2.txt, и если пользователь удалит файл 1.txt , останется файл 2.txt и, наоборот, если сотрет 2.txt — останется файл 1.txt. Файл физически удаляется лишь тогда, когда будет удалено его последнее имя.

Шифрование. Каждый файл или каталог может быть зашифрован, что не даст возможность прочесть его другой инсталляцией ОС.

5.3. Базы данных и СУБД

Развитие вычислительной техники и появление емких внешних запоминающих устройств прямого доступа предопределило интенсивное развитие автоматических и автоматизированных систем разного назначения и масштаба, в первую очередь заметное в области бизнес-приложений. Такие системы работают с большими объемами информации, которая обычно имеет достаточно сложную структуру, требует оперативности в обработке, часто обновляется и в то же время требует длительного хранения. Примерами таких систем являются автоматизированные системы управления предприятием, банковские системы, системы резервирования и продажи билетов и т. д.

Это привело к появлению новой информационной технологии интегрированного хранения и обработки данных — *концепции баз данных,* в основе которой лежит механизм предоставления обрабаты-

вающей программе из всех хранимых данных только тех, которые ей необходимы, и в форме, требуемой именно этой программе.

Под *базои данных (БД)* обычно понимается именованная совокупность данных, отображающая состояние объектов и их отношений в рассматриваемой предметной области. Характерной чертой баз данных является *постоянство*: данные *постоянно* накапливаются и используются; состав и структура данных, необходимых для решения тех или иных прикладных задач, обычно *постоянны* и стабильны во времени; отдельные или даже все элементы данных могут меняться — но и это есть проявление постоянства — *постоянная* актуализация.

Система управления базами данных (СУБД) — это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Главной отличительной чертой баз данных является использование централизованной системы управления данными, причем как на уровне файлов (что свойственно и ФС), так и на уровне элементов данных. Централизованное хранение совместно используемых данных приводит не только к сокращению затрат на создание и поддержание данных в актуальном состоянии, но и к сокращению избыточности информации, упрощению процедур поддержания непротиворечивости и целостности данных.

Эффективное управление внешней памятью является основной функцией СУБД. Эти, обычно специализированные, средства настолько важны с точки зрения эффективности, что при их отсутствии система просто не сможет выполнять некоторые задачи уже потому, что их выполнение будет занимать слишком много времени.

Большинство СУБД работают в среде операционной системы и тесно с ней связаны. Многопользовательские приложения, обработка распределенных запросов, защита данных требуют эффективно использовать ресурсы, управление которыми обычно является функцией ОС. Использование многопроцессорных систем и мультипоточных технологий обработки данных позволяет эффективно обслуживать параллельно выполняемые запросы, но требует координации использования ресурсов между ОС и СУБД. Соответственно, управление доступом и обеспечение защиты также обычно интегрируются с соответствующими средствами операционной системы.

Именно централизованное управление данными обеспечивает:

- сокращение избыточности в хранимых данных;
- совместное использование хранимых данных;

- стандартизацию представления данных, упрощающую эксплуатацию БД;
- разграничение доступа к данным;
- целостность данных, обеспечиваемую процедурами, предотвращающими включение в БД неверных данных и обеспечивающими ее восстановление после отказов системы.

5.3.1. Многоуровневые модели предметной области и управление данными

Обычно отдельная база данных содержит (отражает) информацию о некоторой предметной области — наборе объектов, представляющих интерес для актуальных или предполагаемых пользователей. То есть реальный мир отображается совокупностью конкретных и абстрактных понятий, между которыми существуют (и, соответственно, фиксируются) определенные связи. Выбор для описания предметной области (ПрО) существенных понятий и связей является предпосылкой того, что пользователь будет иметь практически все необходимые ему в рамках задачи знания об объектах предметной области. Но следует отметить, что пользователь, который хочет работать с базой данных, должен владеть основными понятиями, представляющими предметную область.

И в этом смысле абстрагирование позволяет построить такое описание (модель предметной области), которое другой человек сможет не только воспринять, но и безошибочно использовать для работы с описаниями экземпляров объектов, хранимых в базе данных.

Модель предметной области соотносится с реальными объектами и связями так же, как схема маршрутов городского пассажирского транспорта — с фактической траекторией движения автобуса. Схема адекватно отражает действительность на уровне основных понятий — маршрутов и остановок: выбрав по схеме маршрут, пассажир достигнет цели (прибудет на нужную остановку) независимо от того, в каком транспортном ряду будет двигаться автобус.

Представление предметных областей в БД сводится к следующим этапам: 1) фиксация логической точки зрения на данные (т. е. данные рассматриваются независимо от особенностей их хранения и поиска в конкретной вычислительной среде); 2) определение физического представления данных с учетом выбранных структур хранения данных и архитектуры ЭВМ.

Абстрагированное описание предметной области с фиксированной (логической) точки зрения называется *концептуальной схемой*. Само представление логической точки зрения, используемое при абстрагировании, — совокупность функциональных характеристик объектов и особенностей структурной организации данных, а также управления данными, называется *моделью данных*.

Выделяют три классических типа модели данных: иерархические, сетевые, реляционные. Развитие технологий обработки данных привело к появлению постреляционных, объектно-ориентированных, многомерных БД, которые в той или иной степени соответствуют трем упомянутым классическим моделям.

Отображение концептуальной схемы на физический уровень будем называть *внутренней схемой*.

Соотношение этих понятий приведено на рис. 5.6.

Рис. 5.6. Соотношение понятий концептуальной и внутренней схем

Отражение взгляда (точки зрения) отдельного пользователя на концептуальную схему (как вариант восприятия предметной области) будем называть *внешней схемой*. Внешняя схема использует те же абстрактные категории, что и концептуальная, а на практике соответствует логической организации данных в прикладной программе.

В общем случае концепция уровневого представления не требует более трех уровней, однако с практической точки зрения иногда удобно включать схемы дополнительных уровней, например, дополнительный уровень, учитывающий особенности СУБД или различия представлений в случае распределенных БД.

Теоретически вопрос о многообразии уровней абстракции был решен еще в 1960—1970-х годах. Основой для его решения является концепция многоуровневой архитектуры системы базы данных. Например, в отчете CODASYL [CODASYL] предусматривался архитектурный уровень подсхемы, который позволял для каждого конкретного приложения строить свое собственное «видение» используемого подмножества базы данных путем определения его «персональной» подсхемы базы данных.

В более общем виде этот вопрос решен в архитектурной модели ANSI/X3/SPARC [ANSI/SPARC]. Здесь на внешнем уровне может поддерживаться совсем иная модель данных (или даже несколько моделей), чем на концептуальном уровне. Поддержка разнообразных возможностей абстрагирования в такой системе достигается благодаря средствам определения и поддержки межуровневого отображения моделей данных.

Помимо этого, для решения указанной проблемы может использоваться внутримодельная структура, например, механизмы *представлений* (view). В объектных системах для этих целей может использоваться отношение наследования.

На рис. 5.7 приведены некоторые варианты решений. На рис. 5.7, *б* выделена логическая схема, учитывающая особенности СУБД. Пример, приведенный на рис. 5.7, *в*, характерен для варианта распределенной базы данных, объединяющей информацию, представленную разными внутренними схемами.

Рассмотренная трехуровневая архитектура обеспечивает выполнение основных требований, предъявляемых к системам баз данных:

- адекватность отображения предметной области;
- возможность взаимодействия с БД разных пользователей при решении разных прикладных задач;
- обеспечение независимости программ и данных;
- надежность функционирования БД и защита от несанкционированного доступа.

С точки зрения пользователей различных категорий трехуровневая архитектура имеет следующие достоинства:

- системный аналитик, создающий модель предметной области, не обязательно должен быть специалистом в области программирования и вычислительной техники;
- администратор баз данных, обеспечивающий отображение концептуальной схемы во внутреннюю, не должен беспокоиться о корректности представления предметной области;

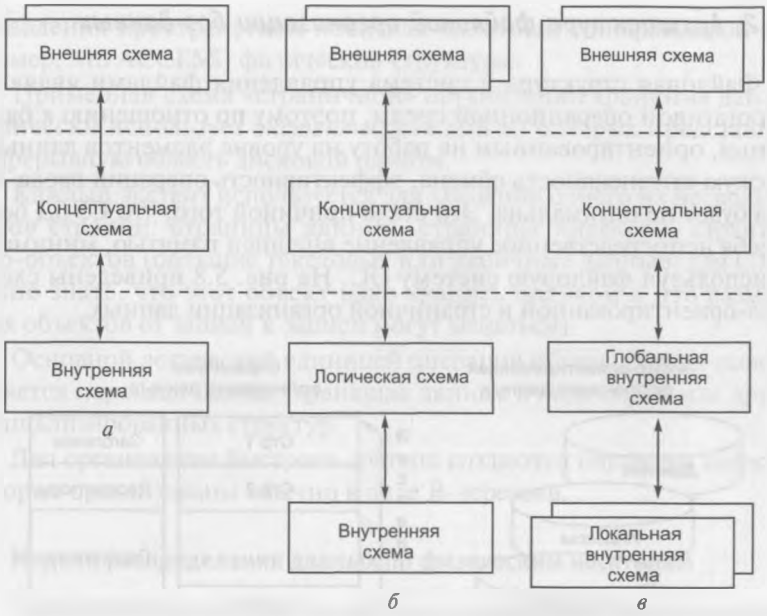


Рис. 5.7. Примеры трехуровнего представления

- конечные пользователи, используя внешнюю схему, могут не вдаваться полностью в предметную область, обращаясь только к необходимым составляющим. При этом исключается возможность несанкционированного обращения к данным вне объявленных внешней схемой, так как формирование ее находится в сфере деятельности администратора базы данных;
- системный аналитик, как и конечный пользователь, не вмешивается во внутреннее представление данных.

Это отражает распространенную практику специализации и разделения ответственности. Главное же заключается в том, что работу по проектированию и эксплуатации баз данных можно разделить на три достаточно самостоятельных этапа. Хотя надо отметить, что на практике создание концептуальной схемы не всегда предшествует построению внешней. Иногда трудно с самого начала полностью определить предметную область, но, с другой стороны, уже известны требования пользователей (именно поэтому создание базы уже имеет смысл). И, кроме того, адекватность модели предметной области, в конце концов, должна подтверждаться практикой пользовательских представлений.

5.3.2. Архитектура файловой организации баз данных

Файловая структура и система управления файлами являются прерогативой операционной среды, поэтому по отношению к базам данных, ориентированным на работу на уровне элементов данных и высокую интенсивность обмена, эффективность операций ввода-вывода будет не оптимальна. Это стало причиной того, что СУБД берут на себя непосредственное управление внешней памятью, минимально используя файловую систему ОС. На рис. 5.8 приведены схемы файл-ориентированной и страничной организации данных.

Рис. 5.8. Файл-ориентированная и страничная организация данных

Файл-ориентированная организация данных

Этот подход отражает точку зрения «идейно чистого» программирования: «сколько типов структур записей — столько и файлов». Именно такой подход обеспечил возможность реализации надежных, достаточно эффективных СУБД, функционирующих по современным меркам в крайне скромных рамках наличных вычислительных ресурсов. Таким образом, БД физически состоит из нескольких файлов: основного, индексного, файла метаданных, файлов указателей и т. д.

Страничная организация данных

Другой подход отражает стремление сосредоточить в СУБД управление данными на всех уровнях — от логической обработки до

управления пространством носителя вплоть до однофайловой (например, MS ACCESS) физической структуры.

Примерная схема «страничной» организации хранения данных физически использует *экстенты*, каждый из которых представляет непрерывную область дисковой памяти.

Каждый экстент используется для хранения одного из нескольких типов страниц: страницы данных, страницы индексов, страницы blob-объектов (большие текстовые или двоичные данные: для СУБД важно знать, что этот объект надо хранить целиком и что размеры этих объектов от записи к записи могут меняться).

Основной логической единицей операций обмена (ввода-вывода) является *страница данных*, хранящая данные в виде *строк* или других специализированных структур.

Для организации быстрого доступа создаются *страницы индексов*, которые организованы обычно в виде B-деревьев.

Модели распределения данных по физическим носителям

Важным фактором, влияющим на производительность подсистемы ввода-вывода, является распределение данных по носителям (дискам). Размещение всех данных БД на одном и том же диске почти всегда приводит к неудовлетворительной производительности. В частности, может оказаться, что процесс формирования журнала, который должен записываться синхронно, в действительности будет выполняться в режиме произвольного, а не последовательного доступа к диску.

Кроме того, выполнение запросов, выбирающих записи из таблицы данных путем последовательного сканирования индекса, будет сильно увеличивать время ожидания ввода-вывода. Обычно сканирование индекса выполняется последовательно, но в данном случае головка диска должна перемещаться для поиска каждой записи данных между выборками индексов. Наконец, следует отметить, что объединение разных функций на одних и тех же физических ресурсах приводит к резкому увеличению времени подвода головок на диске.

Примером, иллюстрирующим подход с точки зрения практических компромиссов выбора решения, являются RAID-массивы. На рис. 5.9 приведены два варианта: RAID-0, обеспечивающий максимальную производительность при «стандартной» надежности, и RAID-1, обеспечивающий «двойную» надежность при «стандартной» производительности.

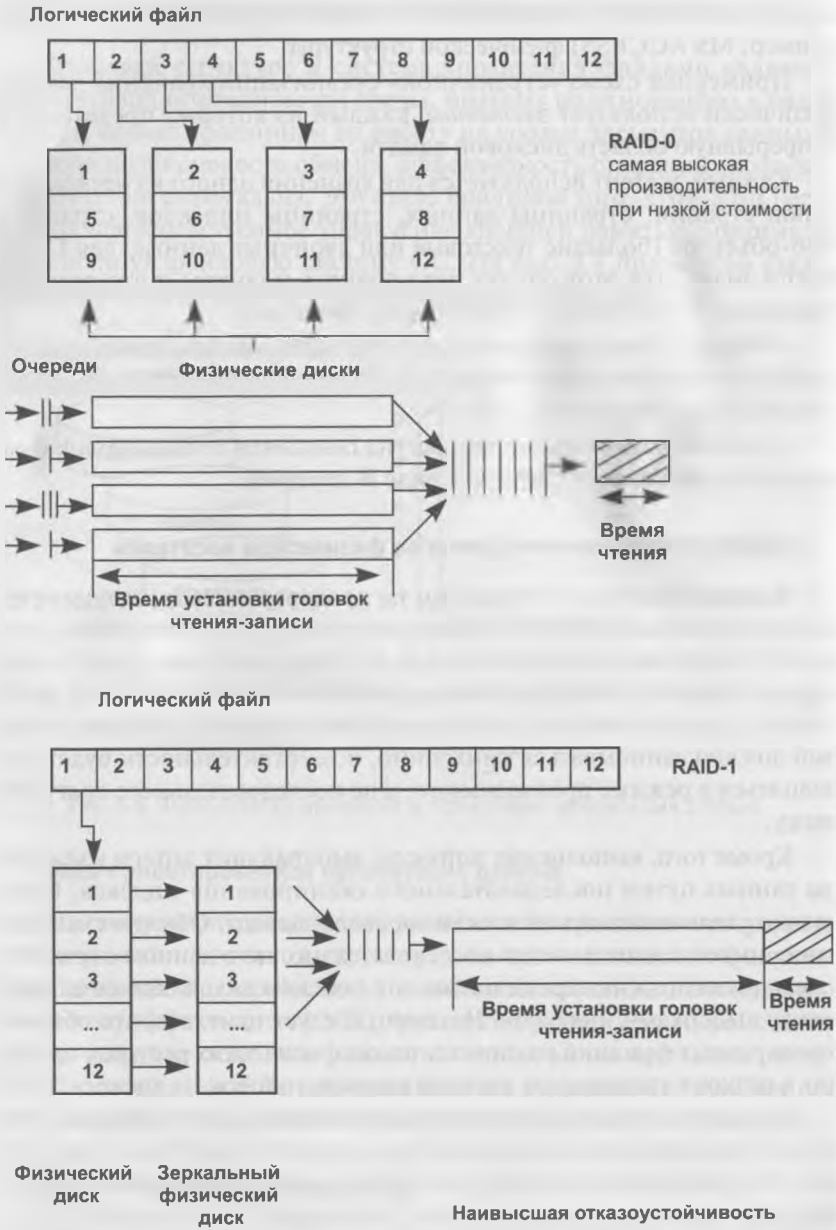


Рис. 5.9. Распределение данных в RAID-массивах

5.3.3. Схема управления данными в СУБД

Рассмотрим примерную последовательность операций, обеспечивающих чтение прикладной программой из базы данных, представленную на рис. 5.10.

- (1) Прикладная программа (клиентское приложение) формирует и выдает системе управления базами данных запрос на чтение необходимых данных, содержащихся в базе.
- (2—3) СУБД отыскивает описание затребованных данных в структуре описания данных прикладного уровня (внешняя схема).
- (4—5) СУБД по глобальному описанию БД (концептуальная схема) определяет необходимые данные на логическом уровне.
- (6—7) СУБД по описанию физической структуры БД (физическая схема) определяет физическую запись (или совокупность

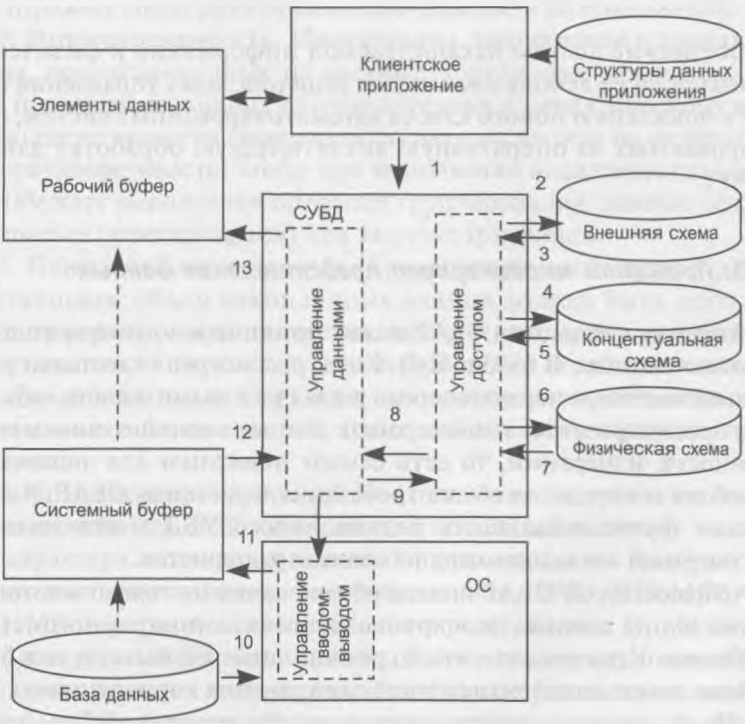


Рис. 5.10. Схема обработки запроса на выборку данных из БД

записей), которую необходимо считать для выборки данных, затребованных прикладной программой.
- (8—9) СУБД через подсистему управления потоками данных выдает операционной системе запрос на чтение хранимой записи.
- (10—11) Подсистема управления вводом-выводом операционной системы осуществляет физическое чтение записи в системный буфер ОС.
- (13) СУБД выделяет необходимую логическую запись, осуществляет форматные преобразования, обусловленные различиями описаний на глобальном и прикладном уровнях, и передает для функциональной обработки приложением данные в рабочий буфер, выделяемый прикладной программой или самой СУБД.

5.4. Хранилища данных и анализ информации

Осознание пользы накапливаемой информации и физические возможности ее использования для решения задач управления привело к появлению нового класса автоматизированных систем, ориентированных на оперативную аналитическую обработку данных (OLAP).

5.4.1. Принципы многомерного представления данных

В основе концепции OLAP лежит принцип многомерного представления данных. В 1993 г. Е.Ф. Кодд, рассмотрев недостатки реляционной модели, в первую очередь указал на невозможность «объединять, просматривать и анализировать данные с точки зрения множественности измерений, то есть самым понятным для аналитиков способом» и определил общие требования к системам OLAP, расширяющим функциональность реляционных СУБД и включающим многомерный анализ как одну из своих характеристик.

Аббревиатурой OLAP иногда обозначается не только многомерный взгляд на данные, но и хранение самих данных в многомерной БД. Однако Кодд отмечал, что «...реляционные БД были, есть и будут наиболее подходящей технологией для хранения корпоративных данных. Необходимость существует не в новой технологии БД, а, скорее, в средствах анализа, дополняющих функции существующих СУБД и

достаточно гибких, чтобы предусмотреть и автоматизировать разные виды интеллектуального анализа, присущие OLAP».

В большинстве случаев средства анализа данных на основе хранилищ данных (ХД) используются для решения следующих задач.

1. Выделение групп данных, сходных по некоторым признакам (кластерный анализ).

2. Нахождение и аппроксимация зависимостей, связывающих анализируемые параметры или события, а также поиск параметров, наиболее значимых в конкретной задаче.

3. Поиск данных, существенно отклоняющихся от выявленных закономерностей.

4. Прогнозирование развития объектов различной природы на основе хранящейся ретроспективной информации об их состоянии в прошлом.

Приведем основные свойства, характерные для хранилищ данных.

1. Ориентация на предметную область. Хранилище в первую очередь отражает специфику предметной области, а не приложений.

2. Интегрированность. Информация, загружаемая в хранилище из баз, ориентированных на частные прикладные задачи, должна быть приведена к единому синтаксическому и семантическому виду. Важно также провести проверку поступающих данных на целостность и непротиворечивость. Чтобы при выполнении аналитических запросов избежать выполнения операций группирования, данные должны обобщаться (агрегироваться) при загрузке хранилища.

3. Неизменяемость данных. Хранилищам свойственна ретроспективность: объем накопленных данных должен быть достаточным для решения аналитических задач. Поэтому важное отличие аналитических систем от систем операционной обработки состоит в том, что данные после загрузки в них остаются неизменными, внесение каких-либо изменений, кроме добавления записей, не предполагается.

4. Поддержка хронологии. Для выполнения большинства аналитических запросов необходим анализ тенденций развития явлений или характера изменения значений переменных во времени, что обычно достигается введением атрибутов типа «ДАТА/ВРЕМЯ»

5. Многомерное концептуальное представление (multi-dimensional conceptual view) представляет собой множественную перспективу, состоящую из нескольких независимых измерений, вдоль которых могут быть проанализированы определенные совокупности данных. Одновременный анализ по нескольким измерениям определяется как

многомерный анализ. Каждое измерение включает направления консолидации данных, состоящие из серии последовательных уровней обобщения, где каждый вышестоящий уровень соответствует большей степени агрегации данных по соответствующему измерению. Так, измерение *Исполнитель* может определяться направлением консолидации, состоящим из уровней обобщения «предприятие — подразделение — отдел — служащий». Измерение *Время* может даже включать два направления консолидации: «год — квартал — месяц — день» и «неделя — день». В этом случае становится возможным произвольный выбор желаемого уровня детализации информации по каждому из измерений.

Кодд определил 12 свойств, которым должна обладать система этого класса (табл. 5.2).

Таблица 5.2. **Свойства систем класса OLAP**

№	Требование (свойство)	Примечание
1	Многомерное концептуальное представление данных	Концептуальное представление модели данных в продукте OLAP должно быть многомерным по своей природе, т. е. позволять аналитикам выполнять интуитивные операции «анализа вдоль и поперек», выбора направлений консолидации и т. д.
2	Прозрачность	Пользователь не должен знать о том, какие конкретные средства используются для хранения и обработки данных, как данные организованы и откуда берутся
3	Доступность	Аналитик должен иметь возможность выполнять анализ в рамках общей концептуальной схемы, но при этом данные могут оставаться под управлением оставшихся от старого наследства СУБД
4	Устойчивая производительность	С увеличением числа измерений и размеров базы данных аналитики не должны столкнуться с каким бы то ни было уменьшением производительности
5	Клиент-серверная архитектура	Серверный компонент должен быть достаточно интеллектуальным и обладать способностью строить общую концептуальную схему на основе обобщения и консолидации различных логических и физических схем корпоративных баз данных для обеспечения эффекта прозрачности

Окончание табл. 5.2

№	Требование (свойство)	Примечание
6	Равноправие измерений	Все измерения данных должны быть равноправны. Дополнительные характеристики могут быть предоставлены отдельным измерениям, но поскольку все они симметричны, данная дополнительная функциональность может быть предоставлена любому измерению
7	Динамическая обработка разреженных матриц	Инструмент OLAP должен обеспечивать оптимальную обработку разреженных матриц. Скорость доступа должна сохраняться вне зависимости от расположения ячеек данных и быть постоянной величиной для моделей, имеющих разное число измерений и различную разреженность данных
8	Поддержка многопользовательского режима	Инструмент OLAP должен предоставлять пользователям конкурентный доступ, обеспечивать целостность и защиту данных, если они имеют необходимость работать одновременно с одной аналитической моделью или создавать различные модели на основе одних корпоративных данных
9	Неограниченная поддержка кроссмерных операций	Вычисления и манипуляция данными по любому числу измерений не должны запрещать или ограничивать любые отношения между ячейками данных. Преобразования, требующие произвольного определения, должны задаваться на функционально полном формульном языке
10	Интуитивное манипулирование данными	Переориентация направлений консолидации, детализация данных в колонках и строках, агрегация и другие манипуляции, свойственные структуре иерархии направлений консолидации, должны выполняться в максимально удобном, естественном и комфортном пользовательском интерфейсе
11	Гибкий механизм генерации отчетов	Должны поддерживаться различные способы визуализации данных, т. е. отчеты должны представляться в любой возможной ориентации
12	Неограниченное количество измерений и уровней агрегации	OLAP инструмент должен иметь несколько измерений в аналитической модели. Каждое из этих измерений должно допускать практически неограниченное количество определенных пользователем уровней агрегации по любому направлению консолидации

5.4.2. Архитектуры хранилищ данных

Рассмотрим некоторые распространенные архитектуры хранилищ.

Виртуальное хранилище данных. В его основе — репозиторий метаданных, которые описывают источники информации (БД транзакционных систем, внешние файлы и др.), SQL-запросы для их считывания и процедуры обработки и предоставления информации. Непосредственный доступ к последним обеспечивает ПО промежуточного слоя. В этом случае избыточность данных нулевая. Конечные пользователи фактически работают с транзакционными системами напрямую со всеми вытекающими отсюда плюсами (доступ к «живым» данным в реальном времени) и минусами (интенсивный сетевой трафик, снижение производительности и реальная угроза их работоспособности вследствие неудачных действий пользователей).

Витрина данных. Витрина данных (Data Mart) — это набор тематически связанных баз данных, которые содержат информацию, относящуюся к отдельным аспектам предметной области (рис. 5.11). По

Рис. 5.11. Структура корпоративной информационно-аналитической системы (ИАС)

сути дела, витрина данных — это облегченный вариант хранилища данных, содержащий только тематически объединенные данные. Витрина данных существенно меньше по объему, чем хранилище данных, и для ее реализации не требуется особо мощная вычислительная техника.

Глобальное хранилище данных. Все более популярной становится идея совместить концепции хранилища и витрины данных в одной реализации и использовать хранилище данных в качестве единственного источника интегрированных данных для всех витрин данных. Метаданные должны содержать описание структур данных хранилища, структур данных, импортируемых из разных источников, сведения о методах загрузки и обобщения данных, средствах доступа и правилах представления информации и т. д.

Тогда естественной становится следующая трехуровневая архитектура системы.

1. Сфера детализированных данных. Это область действия большинства систем, нацеленных на поиск информации. В большинстве случаев реляционные СУБД отлично справляются с возникающими здесь задачами. Общепризнанным стандартом языка манипулирования реляционными данными является SQL. Информационно-поисковые системы, обеспечивающие интерфейс конечного пользователя в задачах поиска детализированной информации, могут использоваться в качестве надстроек как над отдельными базами данных транзакционных систем, так и над общим хранилищем данных.

2. Сфера агрегированных показателей. Комплексный взгляд на собранную в хранилище данных информацию, ее обобщение и агрегация, многомерный анализ являются задачами систем оперативной аналитической обработки данных. Здесь можно или ориентироваться на специальные многомерные СУБД, или оставаться в рамках реляционных технологий. Во втором случае заранее агрегированные данные могут собираться в БД звездообразного вида, либо агрегация информации может производиться на лету в процессе сканирования детализированных таблиц реляционной БД.

3. Сфера закономерностей. Интеллектуальная обработка производится методами *интеллектуального анализа данных* (ИАД, Data Mining), главными задачами которых являются поиск функциональных и логических закономерностей в накопленной информации, построение моделей и правил, которые объясняют найденные аномалии и/или прогнозируют развитие некоторых процессов.

5.4.3. Интеллектуальный анализ данных

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

В общем случае процесс ИАД состоит из трех стадий (рис. 5.12):

1) выявление закономерностей (свободный поиск);

2) использование выявленных закономерностей для предсказания неизвестных значений (прогностическое моделирование);

3) анализ исключений, предназначенный для выявления и толкования аномалий в найденных закономерностях.

Рис. 5.12. Стадии процесса интеллектуального анализа данных

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

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

В первом случае исходные данные могут храниться в детализированном виде и непосредственно использоваться для прогностического моделирования и/или анализа исключений. Это так называемые методы рассуждений на основе анализа прецедентов. Главной проблемой этой группы методов является затрудненность их использования на больших объемах данных, хотя именно при анализе больших хранилищ данных методы ИАД приносят наибольшую пользу.

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

5.5. Особенности и компромиссы реализаций управления данными

В заключение приведем основные отличительные особенности обработки данных, характерные для файловых систем и систем управления базами данных.

Файлы обладают следующими свойствами:

- файл, как правило, представляет собой совокупность записей одного типа, доступ к которым определяется типом *организации* файла и осуществляется только средствами операционной системы;
- файл описывают и используют в прикладной программе, работающей с данными.

Базы данных имеют следующие особенности:

- база данных представляет собой совокупность данных разного типа, причем часто по одним данным получают другие;
- база данных существует независимо от конкретной прикладной программы — база создается с целью интеграции данных, объединяющей данные многих приложений (но определенного назначения). База данных предназначена для *совместного, многофункционального* использования *многими* пользователями *один раз введенных* данных.

Надо отметить, что с точки зрения управления данными СУБД оперируют данными на содержательном уровне, хотя физические структуры, используемые для этих целей, могут и совпадать с аналогичными структурами, создаваемые ОС.

Коренное же отличие СУБД от файловых систем ОС состоит в том, что СУБД устанавливает связь между *содержанием и адресом*, а ОС — между *именем и адресом* данных.

В общем случае можно сказать, что основные задачи обработки данных, решаемые на основе концепций баз данных, сводятся к следующим вопросам.

1. Каким образом сложные нелинейные структуры данных представить в виде линейных — наиболее соответствующих принципу последовательного представления (хранения) в машинной памяти.

2. Каким образом организовать данные, чтобы была возможность эффективного внесения, удаления и редактирования данных.

3. Как организовать данные, чтобы использование пространства памяти (плотность данных) было достаточно рациональным, а скорость доступа к записям данных — высокой.

4. Каким образом организовать данные, чтобы поиск был эффективным и позволял отыскивать записи по нескольким ключам.

При этом, с точки зрения прагматики, создание базы данных — это по существу попытка найти компромисс сразу по нескольким направлениям и сочетаниям нескольких взаимообратных факторов (с точки зрения их влияния на показатель общей эффективности системы), в том числе следующих:

1) эффективность — простота;

2) скорость выборки — стоимость (сложность) аппаратных средств;

3) скорость выборки — сложность процедур доступа;

4) плотность данных — время доступа и сложность процедур;

5) независимость данных — производительность;

6) гибкость средств поиска — избыточность данных или

7) гибкость поиска — скорость поиска;

8) сложность процедур доступа — простота обслуживания.

Контрольные вопросы

1. Перечислите основные задачи обработки данных, решаемые на основе концепций баз данных.
2. Проведите сравнительный анализ понятий «физического» и «логического» представлений.
3. Определите соотношение физической и логической записи.
4. Приведите примерную схему организации файлового ввода-вывода.
5. Проведите сравнительный анализ процессов обработки данных средствами файловой системы и СУБД.

6. Перечислите функции файловых систем.
7. Какова общая организация ФС NTFS?
8. Какие атрибуты файлов вам известны?
9. Охарактеризуйте разновидности размещения файлов в NTFS.
10. Каким образом осуществляется сжатие данных в NTFS?
11. Дайте определение понятия «База данных».
12. Перечислите преимущества и недостатки использования баз данных.
13. Определите основные функции и назначение СУБД.
14. Перечислите основные показатели эффективности обработки данных.
15. Приведите схему управления данными в СУБД.
16. Назовите отличительные особенности использования баз данных в ИС.
17. Перечислите основные требования, предъявляемые к базам данных.
18. Приведите принципы страничной организации данных.
19. Что такое хранилища данных?
20. Перечислите основные свойства OLAP-технологий.

Глава 6
СЕТЕВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ. INTERNET

В исторической перспективе, с появлением в первой половине 1970-х годов видеотерминалов, первоначально возникли структуры «терминал—хост» (локальный — удаленный компьютер). Чуть раньше и независимо развивались глобальные сети (пакетной коммутации), используемые как для функций связи общего назначения, так и для коммуникаций «хост—хост», с целью (в то время) выравнивания использования вычислительных мощностей по часовым поясам (подобно тому, как это осуществляется в сетях энергопередачи). Это были именно *вычислительные сети*. Эта ситуация сохраняется до середины 1980-х годов, когда появление и взрывообразное распространение ПК (как выразился один из тогдашних научных острословов, «карлики-млекопитающие на планете вычислительных динозавров»). Появляются локальные сети, интегрирующие прежде всего информационные ресурсы (файл-сервер), редкие или дорогостоящие технические средства (принт-сервер) и т. п.

Изучение трафика (потоков данных) в развивающихся сетях показало смещение акцентов с распределенных вычислений на обмен информацией — доступ к удаленным базам данных, обмен сообщениями по электронной почте и пр. Таким образом складывалось понятие *информационные сети*.

Наконец, в 1980—1990-е годы широко распространяется технология TCP/IP, обеспечивая рост и развитие «сети сетей» — Internet, которая представляет собой *глобальную информационно-вычислительную сеть*.

6.1. Основные понятия

Системы терминал—хост

Первые системы совместной эксплуатации информационных и вычислительных ресурсов (*системы коллективного пользования*) появляются в 1960—1970-е гг. и относятся к вычислительным системам с разделением времени. Первоначально операционные системы ЭВМ (ОС) были рассчитаны на пакетную обработку информации, затем с созданием интерактивных терминальных устройств появляется возможность совместной работы пользователей в реальном масштабе времени. Основные этапы развития систем доступа к информационным ресурсам представлены на рис. 6.1 и включают следующие схемы.

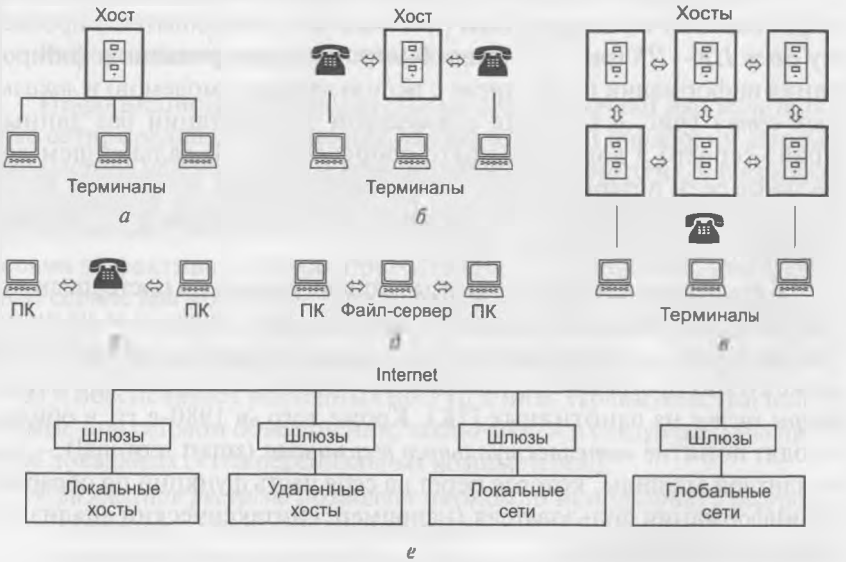


Рис. 6.1. Варианты коллективного использования информационно-вычислительных ресурсов:
а — локальный хост; *б* — удаленный хост; *в* — глобальная сеть; *г* — коммуникации ПК — ПК; *д* — локальная сеть; *е* — Internet

1. Взаимодействие *терминала* (конечный пользователь, источник запросов и заданий) и *хоста* (центральная ЭВМ, держатель всех информационных и вычислительных ресурсов) (рис. 6.1, *а, б*). Может

осуществляться как в *локальном*, так и в *удаленном* режиме, во втором случае, как правило, некоторая совокупность пользователей (дисплейный класс) размещается в так называемом *абонентском пункте* — комплексе, снабженном контроллером (устройством управления), принтером, концентратором и обеспечивающим параллельную работу пользователей с удаленным хостом. Связь между хостом и абонентским пунктом в этом случае осуществлялась с помощью *модемов*, по телефонным каналам.

2. На следующем этапе (рис. 6.1, *в*) формируются *сети передачи данных* (из существующих общих и специальных цифровых каналов), позволяющие как осуществлять более тесное взаимодействие *терминал—хост,* так и обмен *хост—хост* для реализации распределенных баз данных и децентрализации процессов обработки информации.

3. Появление и массовое распространение *персональных компьютеров* выводит на первый план (для массового пользователя) проблему *связи ПК—ПК* (рис. 6.1, *г*) для быстрого резервирования и копирования информации (в том числе с использованием модемов) и *локальных сетей* (рис. 6.1, *д*) для совместной эксплуатации баз данных (файл-сервер) и дорогостоящего оборудования. В дальнейшем локальные сети потеряли самостоятельное значение вследствие интеграции с глобальными в *двухуровневые сети,* строящиеся по единому принципу в рамках Internet (рис. 6.1, *е*).

В последующем перечисленные конфигурации не претерпели существенных изменений, однако понятия *хост и терминал* из чисто аппаратурных трансформировались в аппаратурно-программные и даже сугубо программные (например, *эмуляторы терминала и эмуляторы хоста* на однотипных ПК). Кроме того, в 1980-е гг. в обиход входит понятие *интеллектуального терминала* (smart terminal) — сателлитной машины, которая берет на себя часть функций по обработке информации пользователя (например, синтаксический анализ запроса или программы).

Системы клиент—сервер

Таким образом, по мере развития представлений о распределенных вычислительных процессах и процессах обработки данных складывается концепция *архитектуры «клиент—сервер»* — обобщенное представление о взаимодействии двух компонентов информационной технологии (технического и/или программного обеспечения) в вы-

числительных системах и сетях, среди которых логически или физически могут быть выделены:

- активная сторона (источник запросов, клиент);
- пассивная сторона (сервер, обслуживание запросов, источник ответов).

Взаимодействие клиент—сервер в сети осуществляется в соответствии с определенным стандартом, или *протоколом*, — совокупностью соглашений об установлении/прекращении связи и обмене информацией.

Обычно клиент и сервер работают в рамках единого протокола — telnet, ftp, gopher, http и пр., однако в связи с недостаточностью такого подхода появляются *мультипротокольные клиенты и серверы*, например — браузер Netscape Navigator.

Разновидности функциональных структур «клиент—сервер» рассмотрены в следующей главе.

Информационно-вычислительные сети

Информационно-вычислительные сети включают вычислительные сети, предназначенные для распределенной обработки данных (совместное использование вычислительных мощностей), и информационные сети, предназначенные для совместного использования информационных ресурсов. Сетевая технология обработки информации весьма эффективна, так как предоставляет пользователю необходимый сервис для коллективного решения различных распределенных прикладных задач, увеличивает степень использования имеющихся в сети ресурсов (информационных, вычислительных, коммуникационных) и обеспечивает удаленный доступ к ним. Преимущества, получаемые при сетевом объединении, заключаются в следующем (на примере локальных сетей персональных компьютеров):

- *разделение ресурсов* позволяет экономно использовать ресурсы, например, управлять периферийными устройствами, такими как печатающие устройства, со всех присоединенных рабочих станций;
- *разделение данных* предоставляет возможность доступа и управления базами данных с периферийных рабочих мест, нуждающихся в информации;
- *разделение программных средств* предоставляет возможность одновременного использования централизованных, ранее установленных программных средств;

- *разделение ресурсов процессора* — при этом возможно использование вычислительных мощностей для обработки данных другими системами, входящими в сеть;
- *многопользовательский режим* содействует одновременному использованию централизованных прикладных программных средств, ранее установленных и управляемых, например, если пользователь системы работает с другим заданием, то текущая выполняемая работа отодвигается на задний план.

Коммутация связей. Рассмотрим подробнее технологии коммутации связей в сетях большого масштаба. Распределение потоков сообщений с целью доставки каждого сообщения по адресу осуществляется на узлах коммутации (УК) с помощью коммутационных устройств. Система распределений потоков сообщений в УК получила название *системы коммутации*.

Под коммутацией в сетях передачи данных имеется в виду совокупность операций, обеспечивающих в узлах коммутации передачу информации между входными и выходными устройствами в соответствии с указанным адресом. При коммутации с накоплением (КН) абонент имеет постоянную прямую связь со своим УК и передает на него информацию. Затем эта информация передается через узлы коммутации другим абонентам, причем в случае занятости исходящих каналов информация запоминается в узлах и передается по мере освобождения каналов в нужном направлении.

Коммутация пакетов. Широкое распространение получил метод коммутации пакетов (КП), или пакетной коммутации, являющийся разновидностью коммутации с накоплением. При КП сообщения разбиваются на меньшие части, называемые пакетами, каждый из которых имеет установленную максимальную длину. Эти пакеты нумеруются и снабжаются адресами и прокладывают себе путь по сети (методом передачи с промежуточным хранением).

Множество пакетов одного и того же сообщения может передаваться одновременно, что и является одним из главных преимуществ систем КП. Приемник в соответствии с заголовками пакетов выполняет сборку пакетов в исходное сообщение и отправляет его получателю.

Благодаря возможности не накапливать сообщения целиком в узлах коммутации не требуется внешних запоминающих устройств, и вполне можно ограничиться оперативной памятью, а в случае ее переполнения использовать различные механизмы задержки передаваемых пакетов в местах их генерации.

Части одного и того же сообщения могут в одно и то же время находиться в различных каналах связи, более того, когда начало сообщения уже принято, его конец отправитель может еще даже не передавать в канал.

В сети с КП осуществляется следующий процесс передачи (рис. 6.2):

- вводимое в сеть сообщение разбивается на части — пакеты длиной обычно до 1000—2000 единичных интервалов, содержащие адрес ОП получателя. Указанное разбиение осуществляется или в оконечном пункте, если он содержит ЭВМ, или в ближайшем к ОП УК;
- если разбиение сообщения на пакеты происходит в УК, то дальнейшая передача пакетов осуществляется по мере их формирования, не дожидаясь окончания приема в УК целого сообщения;
- в узле КП пакет запоминается в оперативной памяти (ОЗУ) и по адресу определяется канал, по которому он должен быть передан;
- если этот канал к соседнему узлу свободен, то пакет немедленно передается на соседний узел КП, в котором повторяется та же операция;
- если канал к соседнему узлу занят, то пакет может небольшое время храниться в ОЗУ до освобождения канала;

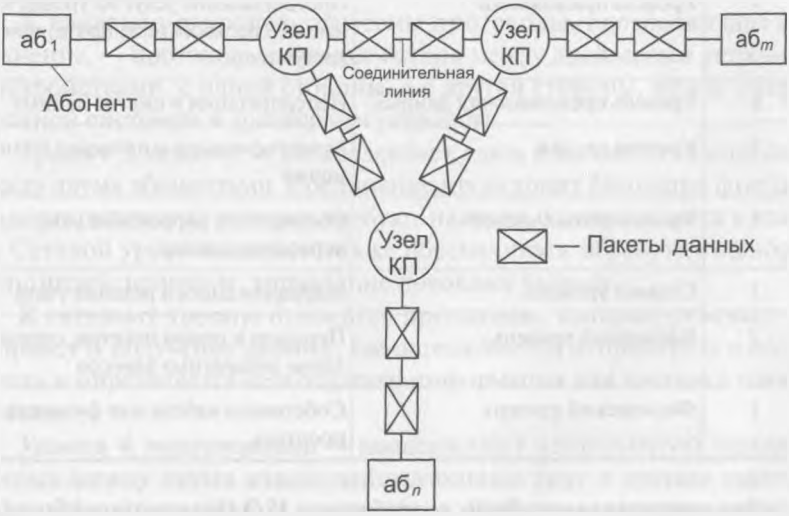


Рис. 6.2. Схема коммутации пакетов

- при хранении пакеты устанавливаются в очереди по направлению передачи, причем длина очереди не превышает 3—4 пакета. Если длина очереди превышает допустимую, пакеты стираются из ОЗУ, и их передача должна быть повторена.

Пакеты, относящиеся к одному сообщению, могут передаваться по разным маршрутам в зависимости от того, по какому из них в данный момент они с наименьшей задержкой могут пойти к адресату. В связи с тем, что время прохождения до сети пакетов одного сообщения может быть различным (в зависимости от маршрута и задержек в УК), порядок их перехода в ОП (к получателю) может не соответствовать порядку пакетов.

Эталонная модель внутри- и межсетевого взаимодействия (OSI Reference Model). Многослойный (многоуровневый) характер сетевых процессов приводит к необходимости рассмотрения многоуровневых моделей телекоммуникационных сетей. В качестве эталонной модели утверждена семиуровневая модель, в которой все процессы, реализуемые открытой системой, разбиты на взаимно подчиненные уровни. В данной модели обмен информацией может быть представлен в виде стека (табл. 6.1).

Таблица 6.1. **Семиуровневая модель (стек) протоколов межсетевого обмена OSI**

№ уровня	Наименование уровня	Содержание
7	Уровень приложений	Предоставление услуг на уровне конечного пользователя: почта, теледоступ и пр
6	Уровень представления данных	Интерпретация и сжатие данных
5	Уровень сессии	Аутентификация и проверка полномочий
4	Транспортный уровень	Обеспечение корректной сквозной пересылки данных
3	Сетевой уровень	Маршрутизация и ведение учета
2	Канальный уровень	Передача и прием пакетов, определение аппаратных адресов
1	Физический уровень	Собственно кабель или физический носитель

Эти представления были разработаны ISO (International Standard Organization) и получили название «Семиуровневой модели сетевого

обмена» (Open System Interconnection Reference Model), или ВОС (Взаимодействие открытых систем).

Основная идея этой модели заключается в том, что каждому уровню отводится конкретная роль, в том числе и в транспортной среде. Благодаря этому общая задача передачи данных расчленяется на отдельные, легко обозримые задачи. Необходимые соглашения для связи одного уровня с выше- и нижерасположенными называют *протоколом*.

Рассмотрим далее основные характеристики модели ВОС.

Уровень 1, физический — определяет характеристики физической сети передачи данных, которая используется для межсетевого обмена. Это такие параметры, как напряжение в сети, сила тока, число контактов на разъемах, электрические, механические, функциональные и процедурные параметры для физической связи в системах.

Уровень 2, канальный — представляет собой комплекс процедур и методов управления каналом передачи данных, организованный на основе физического соединения. Канальный уровень формирует из данных, передаваемых 1-м уровнем, так называемые «кадры», последовательности пакетов. Каждый пакет содержит адреса источника и места назначения, а также средства обнаружения ошибок. На этом уровне осуществляются управление доступом к передающей среде, используемой несколькими ЭВМ, синхронизация, обнаружение и исправление ошибок.

К канальному уровню отнесены протоколы, определяющие соединение, — протоколы взаимодействия между драйверами устройств и устройствами, с одной стороны, а с другой стороны, между операционной системой и драйверами устройств.

Уровень 3, сетевой — устанавливает связь в вычислительной сети между двумя абонентами. Соединение происходит благодаря функциям маршрутизации, которые требуют наличия сетевого адреса в пакете. Сетевой уровень должен также обеспечивать обработку ошибок, мультиплексирование, управление потоками данных.

К сетевому уровню относятся протоколы, которые отвечают за отправку и получение данных, где определяются отправитель и получатель и определяется необходимая информация для доставки пакета по сети.

Уровень 4, транспортный — поддерживает непрерывную передачу данных между двумя взаимодействующими друг с другом удаленными пользовательскими процессами. Качество транспортировки, безошибочность передачи, независимость вычислительных сетей,

сервис транспортировки из конца в конец, минимизация затрат и адресация связи гарантируют непрерывную и безошибочную передачу данных.

Транспортный протокол связывает нижние уровни (физический, канальный, сетевой) с верхними уровнями, которые реализуются программными средствами. Этот уровень как бы разделяет средства формирования данных в сети от средств их передачи. Сетевой уровень предоставляет услуги транспортному, который требует от пользователей запроса на качество обслуживания сетью.

Уровень 5, сеансовый — на данном уровне осуществляется управление сеансами (сессиями) связи между двумя взаимодействующими прикладными пользовательскими процессами (пользователями). Определяются начало и окончание сеанса связи: нормальное или аварийное; определяются время, длительность и режим сеанса связи, точки синхронизации для промежуточного контроля и восстановления при передаче данных, восстанавливается соединение после ошибок во время сеанса связи без потери данных.

Уровень 6, представления данных (представительский) — управляет представлением данных в необходимой для программы пользователя форме, обеспечивает генерацию и интерпретацию взаимодействия процессов, кодирование/декодирование данных, в том числе компрессию и декомпрессию данных.

Уровень 7, прикладной (уровень прикладных программ или приложений) — определяет протоколы обмена данными этих прикладных программ — в его ведении находятся прикладные сетевые программы, обслуживающие файлы, а также выполняются вычислительные, информационно-поисковые работы, логические преобразования информации, передача почтовых сообщений и т. п. Одна из задач этого уровня — обеспечить удобный интерфейс пользователя.

Таким образом, на разных уровнях обмен происходит в различных единицах информации: *биты, кадры, фреймы, пакеты, сеансовые сообщения, пользовательские сообщения*. Уровень может «ничего не знать» о содержании сообщения, но он должен «знать», что дальше делать с этим сообщением.

Базовые сетевые топологии

Проиллюстрируем (на примере локальных сетей) основные принципы комплексирования сетевого оборудования (или *топологии сетей*). При создании сети в зависимости от задач, которые она должна

будет выполнять, может быть реализована одна из трех базовых топологий: «звезда», «кольцо» и «общая шина» — рис. 6.3, табл. 6.2.

Концепция *топологии сети в виде звезды* заимствована из области больших ЭВМ: головная (хост) машина получает и обрабатывает все данные с периферийных устройств (терминалов или рабочих станций пользователя), являясь единственным активным узлом обработки данных, поэтому коллизий (столкновений) данных не возникает.

Информация между любыми двумя пользователями в этом случае проходит через центральный узел вычислительной сети. Пропускная способность сети определяется вычислительной мощностью узла и гарантируется для каждой рабочей станции.

При *кольцевой топологии сети* рабочие станции связаны одна с другой по кругу, т. е. рабочая станция 1 с рабочей станцией 2, рабочая

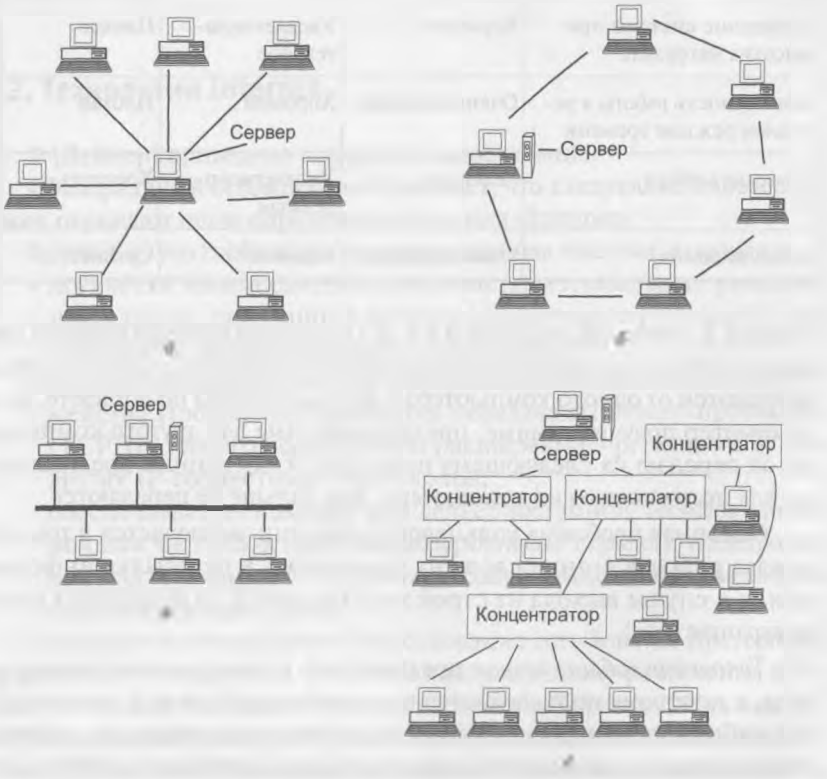


Рис. 6.3. Базовые сетевые топологии:
а — «звезда»; *б* — «кольцо»; *в* — шинная топология; *г* — логическое кольцо

Таблица 6.2. **Основные характеристики сетей разной топологии**

Характеристики	Топология		
	Звезда	Кольцо	Шина
Стоимость расширения	Незначительная	Средняя	Средняя
Присоединение абонентов	Пассивное	Активное	Пассивное
Защита от отказов	Незначительная	Незначительная	Высокая
Размеры системы	Любые	Любые	Ограничены
Защищенность от прослушивания	Хорошая	Хорошая	Незначительная
Стоимость подключения	Незначительная	Незначительная	Высокая
Поведение системы при высоких нагрузках	Хорошее	Удовлетворительное	Плохое
Возможность работы в реальном режиме времени	Очень хорошая	Хорошая	Плохая
Разводка кабеля	Хорошая	Удовлетворительная	Хорошая
Обслуживание	Очень хорошее	Среднее	Среднее

станция 3 с рабочей станцией 4 и т. д. Последняя рабочая станция связана с первой. Коммуникационная связь замыкается в кольцо, данные передаются от одного компьютера к другому как бы по эстафете. Если компьютер получит данные, предназначенные для другого компьютера, он передает их следующему по кольцу. Если данные предназначены для получившего их компьютера, они дальше не передаются.

Основная проблема кольцевой топологии заключается в том, что каждая рабочая станция должна участвовать в пересылке информации, и в случае выхода из строя хотя бы одной из них работа в сети прекращается.

Топология «общая шина» предполагает использование одного кабеля, к которому подключаются все компьютеры сети. В данном случае кабель используется совместно всеми станциями по очереди. Принимаются специальные меры для того, чтобы при работе с общим кабелем компьютеры не мешали друг другу передавать и принимать данные.

Рабочие станции в любое время, без прерывания работы всей вычислительной сети, могут быть подключены к ней или отключены. Функционирование вычислительной сети не зависит от состояния отдельной рабочей станции.

Надежность здесь выше, так как выход из строя отдельных компьютеров не нарушает работоспособность сети в целом. Но так как используется только один кабель, в случае его повреждения нарушается работа всей сети.

Наряду с отмеченными базовыми на практике применяются *комбинированные топологические решения*. К таковым относится, например, *логическая кольцевая сеть*, которая физически монтируется как соединение звездных топологий (рис. 6.3, *г*). Отдельные «звезды» включаются с помощью специальных коммутаторов (англ. Hub — концентратор).

6.2. Технологии Internet

В [Лейнер] приведено следующее определение:

«Федеральный сетевой совет признает, что следующие словосочетания отражают наше определение термина «Internet».

Internet — это глобальная информационная система, которая:

- логически взаимосвязана пространством глобальных уникальных адресов, основанных на Internet-протоколе (IP) или на последующих расширениях или преемниках IP;
- способна поддерживать коммуникации с использованием семейства Протокола управления передачей/Internet-протокола (TCP/IP) или его последующих расширений/преемников и/или других IP-совместимых протоколов;
- обеспечивает, использует или делает доступной, на общественной или частной основе, высокоуровневые сервисы, надстроенные над описанной здесь коммуникационной и иной связанной с ней инфраструктурой».

За два десятилетия своего существования сеть Internet претерпела кардинальные изменения. Она зарождалась в эпоху разделения времени, но сумела выжить во времена господства персональных компьютеров, одноранговых сетей, систем клиент/сервер и сетевых компьютеров. Она проектировалась до первых ЛВС, но впитала эту новую сетевую технологию, равно как и появившиеся позднее сервисы коммутации ячеек и кадров. Она задумывалась для поддержки широкого

спектра функций, от разделения файлов и удаленного входа до разделения ресурсов и совместной работы, породив электронную почту и, в более поздний период, Всемирную паутину.

Рассмотрим некоторые основные Internet-технологии.

Система адресов Internet

Сеть сетей — Internet базируется на принципах пакетной коммутации и реализует многоуровневую совокупность протоколов, подобную рассмотренной выше модели OSI. Прежде чем перейти к описанию данных протоколов, отметим, что на каждом из уровней используются определенные системы адресации, позволяющие осуществлять передачу сообщений и адресацию информационных ресурсов. Основными типами адресов являются следующие:

- адрес Ethernet;
- IP-адрес (основной адрес в Internet);
- доменные адреса;
- почтовые адреса;
- номера портов;
- универсальный локатор (идентификатор) сетевого ресурса (URI/URL).

Адрес Ethernet. Internet поддерживает разные физические среды, из которых наиболее распространенным аппаратурным средством реализации локальных сетей (нижний уровень многоуровневых сетей) является технология *Ethernet*.

В локальной сети обмен осуществляется *кадрами Ethernet*, каждый из которых содержит адрес назначения, адрес источника, поле типа и данные. Каждый *сетевой адаптер* (интерфейс, карта Ethernet — физическое устройство, подключающее компьютер к сети) имеет свой *сетевой адрес,* размер которого — 6 байт.

Существенно то, что такой адрес является *глобально уникальным* — фирмам-производителям выделены списки адресов, в рамках которых они обязаны выпускать карты. Адрес записывается в виде шести групп шестнадцатеричных цифр по две в каждой (шестнадцатеричная запись байта). Первые три байта называются префиксом (что определяет 2^{24} различных комбинаций, или почти 17 млн адресов), и именно они закреплены за производителем.

Адаптер «слушает» сеть, принимает адресованные ему кадры и широковещательные кадры с адресом *FF:FF:FF:FF:FF:FF* и отправляет кадры в сеть, причем в каждый момент времени в сегменте узла сети находится только один кадр.

Собственно Ethernet-адрес соответствует не компьютеру, а его сетевому интерфейсу. Таким образом, если компьютер имеет несколько интерфейсов, то это означает, что каждому интерфейсу будет назначен свой Ethernet-адрес. Каждой карте Ethernet соответствуют Ethernet-адрес и IP-адрес, которые уникальны в рамках Internet.

IP-адрес. Представляет собой 4-байтовую последовательность, причем каждый байт этой последовательности записывается в виде десятичного числа. Адрес состоит из двух частей: *адреса сети и номера хоста*. Обычно под хостом понимают компьютер, но в общем случае — это любое устройство, которое имеет свой сетевой интерфейс.

Существует несколько *классов адресов*, отличающихся друг от друга количеством битов, отведенных на адрес сети и адрес хоста в сети. В табл. 6.3 приведены характеристики основных классов.

Таблица 6.3. **Классы IP-адресов**

Класс сети	Байт 1	Байт 2	Байт 3	Байт 4
A	0 Сеть	Номер хоста		
B	10 Номер сети		Номер хоста	
C	110 Номер сети			Хост

Назначение классов IP-адресов:

A — использование в больших сетях общего доступа;

B — в сетях среднего размера (большие компании, научно-исследовательские институты, университеты);

C — в сетях с небольшим числом компьютеров (небольшие компании и фирмы).

Среди IP-адресов несколько зарезервировано под специальные случаи (табл. 6.4).

Таблица 6.4. **Выделенные IP-адреса**

IP-адрес		Интерпретация
Номер сети	Номер хоста	
0.0 (0000_{16})	0.0	Данный узел сети
Номер сети	0.0	Данная ip-сеть
0.0	Номер узла	Узел в данной (локальной) сети
255.255 ($FFFF_{16}$)	255.255 ($FFFF_{16}$)	Все узлы в данной локальной ip-сети
Номер сети	255.255	Все узлы указанной IP-сети

Для установления соответствия IP-адреса адресу Ethernet в локальных сетях используется *Adress resolution Protocol (ARP)*. ARP-таблица заполняется автоматически; если нужного адреса в таблице нет, то в сеть посылается широковещательный запрос типа «чей это IP-адрес?», который получают все сетевые интерфейсы, но отвечает только владелец адреса.

Система доменных имен. Хотя числовая адресация удобна для машинной обработки таблиц маршрутов, она очевидно неприемлема для использования человеком. Для облегчения взаимодействия вначале применялись таблицы соответствия числовых адресов именам машин. Эти таблицы сохранились и используются многими прикладными программами.

Пользователь для обращения к машине может использовать как IP-адрес, так и имя.

По мере роста сети была разработана *система доменных имен* — DNS (Domain Name System), которая строится по иерархическому принципу, однако эта иерархия не является строгой. Фактически нет единого корня всех доменов Internet. В 1980-е гг. были определены первые домены (национальные, США) верхнего уровня: `gov`, `mil`, `edu`, `com`, `net`. Позднее появились национальные домены других стран: `uk`, `jp`, `au`, `ch` и т. п.

Вслед за доменами верхнего уровня следуют домены, определяющие либо регионы (rf, msk), либо организации (kiae); следующие уровни иерархии могут быть закреплены за небольшими организациями либо за подразделениями больших организаций.

Наиболее популярной программой поддержки DNS является *BIND (Berkeley Internet Name Domain) — сервер доменных имен*, реализованный в университете Беркли, который широко применяется в Intenet. Он обеспечивает поиск доменных имен и IP-адресов для любого узла сети. BIND обеспечивает также рассылку сообщений электронной почты через узлы Internet.

Почтовые адреса. В Internet принята система адресов, которая базируется на доменном адресе машины, подключенной к сети. Почтовый адрес состоит из двух частей: идентификатора пользователя, который записывается перед знаком «коммерческое АТ» — «@», и доменного адреса машины, который записывается после знака «@».

Различают следующие типы адресов:

- *местный адрес* — адрес на машине, с которой осуществляется отправка почты;

- *адреса UUCP*, которые могут иметь вид:
  ```
  host!user
  host!host!user
  user@host.uucp
  ```
- *адреса SMTP* — стандартные для Internet, как например:
  ```
  usr@host
  usr@host.domain
  user@[remote.host`s.internet.address]
  ```

Если машина, с которой отправляется почта, имеет прямую линию связи по протоколу UUCP со следующей машиной (в адресе), то почта передается на эту машину; если такого соединения нет, то почта не рассылается и выдается сообщение об ошибке. (Программа рассылки почты Sendmail сама преобразует адреса формата SMTP в адреса UUCP, если доставка сообщения осуществляется по этому протоколу.) Если в системе для адресации используется Berkeley Internet Name Domain сервер, то sendmail может определять адреса получателей, используя сервис BIND, если нет, то sendmail сама определяет адреса.

При рассылке может использоваться и смешанная адресация:

- `user%hostA@hostB` — почта отправляется с машины hostB на машину hostA;
- `user!hostA@hostB` — почта отправляется с машины hostB на машину hostA;
- `hostA!user%hostB` — почта отправляется с hostA на hostB.

TCP/UDP-порт — условный номер соединения с хост-машиной по определенному протоколу прикладного уровня (точнее, информационный сервис, WKS — Well Known Services, прикладная программа, которая осуществляет обслуживание по определенном порту TCP или UDP). К сервисам относятся: доступ в режиме удаленного терминала, доступ к файловым архивам FTP, доступ к серверам World Wide Web и т. п.

Система универсальных индентификаторов ресурсов (URI/URL). Система разработана для использования в WWW, и в ее основу заложены следующие принципы.

Расширяемость — новые адресные схемы должны были легко вписываться в существующий синтаксис URI; была достигнута за счет выбора определенного порядка интерпретации адресов, который базируется на понятии «адресная схема». Идентификатор схемы стоит перед остатком адреса, отделен от него двоеточием и определяет порядок интерпретации остатка.

Полнота — по возможности любая из существовавших схем должна была описываться посредством URI.

Читаемость — адрес должен легко пониматься человеком, что вообще характерно для технологии WWW — документы вместе с ссылками могут разрабатываться в обычном текстовом редакторе.

Формат URL включает:

- схему адреса (тип протокола доступа — http, gopher, wais, telnet, ftp и т. п.);
- IP- или доменный адрес машины;
- номер TCP-порта;
- адрес ресурса на сервере (каталог или путь);
- имя HTML-файла и метку;
- критерий поиска данных.

Для каждого вида протокола приложений выбирается свое подмножество полей из представленного выше списка.

Схема HTTP — основная для WWW. Содержит идентификатор или адрес машины, TCP-порт, путь (директорию в сервере) и, возможно, метку, параметры (критерии поиска). Например:

```
http://polyn.net.kiae.su/polyn/index.html
```

В данном случае путь состоит из доменного адреса машины, на которой установлен сервер HTTP, и пути от корня дерева сервера к файлу «index.html».

```
http://polyn.net.kiae.su/altai/volume4.html#first
```

Символ «#» отделяет имя документа от имени метки.

Схема FTP позволяет адресовать файловые архивы FTP из программ-клиентов World Wide Web. При этом возможно указание не только имени схемы, адреса FTP-архива, но и идентификатора пользователя и даже его пароля. Наиболее часто данная схема используется для доступа к публичным архивам FTP:

```
ftp://polyn.net.kiae.su/pub/index.txt
```

В данном случае записана ссылка на архив «polyn.net.kiae.su» с идентификатором «anonymous» или «ftp» (анонимный доступ). Если есть необходимость указать идентификатор пользователя и его пароль, то можно это сделать перед адресом машины:

```
ftp://nobody:password@polyn.net.kiae.su/users/local/pub
```

В данном случае эти параметры отделены от адреса машины символом «@», а друг от друга — двоеточием. В некоторых системах можно указать и тип передаваемой информации, но данная возможность не стандартизована.

Схема Gopher используется для ссылки на ресурсы распределенной информационной системы Gopher; состоит из идентификатора и пути, в котором указываются адрес Gopher-сервера, тип ресурса и команда Gopher.

```
gopher://gopher.kiae.su:70:/7/software
```

В данном примере осуществляется доступ к Gopher-серверу gopher.kiae.su через порт 70 для поиска (тип 7) слова *software*. Следует заметить, что тип ресурса, в данном случае 7, передается не перед командой, а вслед за ней.

Схема MAILTO предназначена для отправки почты по стандарту RFC-822 (стандарт почтового сообщения). Общий вид схемы выглядит так:

```
mailto:paul@quest.polyn.kiae.su
```

Схема NEWS — просмотр сообщений системы Usenet. При этом используется следующая нотация:

```
news:comp.infosystems.gopher
```

В данном примере пользователь получит идентификаторы статей из группы «comp.infosystems.gopher» в режиме уведомления. Можно получить и текст статьи (например, 86-я статья из группы), но тогда необходим ее идентификатор:

```
news:086@comp.infosystems.gopher
```

Схема TELNET осуществляет доступ к ресурсу в режиме удаленного терминала. Обычно клиент вызывает дополнительную программу для работы по протоколу telnet. При использовании этой схемы необходимо указывать идентификатор пользователя, допускается использование пароля:

```
telnet://guest:password@apollo.polyn.kiae.su
```

Схема WAIS (протокол Z39.50). WAIS — распределенная информационно-поисковая система, работающая в режимах поиска и про-

смотра. При поиске используется форма со знаком «?», отделяющим адресную часть от ключевых слов:

```
wais://wais.think.com/wais?guide
```

В данном случае обращаются к базе данных wais на сервере wais.think.com с запросом на поиск документов, содержащих слово *guide*. Сервер возвращает клиенту список идентификаторов документов, после получения которого можно использовать вторую форму схемы wais — запрос на просмотр документа:

```
wais://wais.think.com/wais/wtype/039=/user/letter.txt,
```

где 039 — идентификатор документа.

Схема FILE — WWW-технология используется как в сетевом, так и в локальном режимах. Например, для локального режима используют

```
file:///C|/text/html/index.htm
```

В данном примере приведено обращение к документу на персональном компьютере.

Существует еще несколько схем, которые на практике не используются или находятся в стадии разработки, поэтому останавливаться на них мы не будем.

Совокупность протоколов Internet

Совокупность протоколов Internet (*стек, или семейство протоколов TCP/IP*) отличается от вышерассмотренной модели OSI и обычно ограничивается схемой, представленной в табл. 6.4. Обе архитектуры включают похожие уровни, в TCP/IP несколько слоев OSI-модели объединены в один.

Взаимодействие на уровне прикладных протоколов осуществляется путем обмена командами установления/прекращения соединений (типа **open/close**), приема/передачи (**send/receive**) и собственно данными. Рассмотрим собственно протоколы TCP/IP канального, сетевого, транспортного уровней:

- *TCP* — Transmission Control Protocol — базовый транспортный протокол, давший название всему семейству протоколов TCP/IP;
- *UDP* — User Datagram Protocol — второй по распространенности транспортный протокол семейства TCP/IP;
- *IP* — Internet Protocol — межсетевой протокол;

Таблица 6.5. **Структура стека протоколов TCP/IP**

Модель OSI	TCP/IP (Internet)
Прикладной	Уровень приложений (прикладные программы конечных пользователей)
Представительный	
Сеансовый	
Транспортный	Транспортный уровень (связь между программами в сети)
Сетевой	Сетевой уровень (базовые коммуникации, адресация и маршрутизация)
Передача данных	Канальный уровень (сетевые аппаратные средства и драйверы устройств)
Физический	

- *ARP* — Address Resolution Protocol — используется для определения соответствия IP-адресов и Ethernet-адресов;
- *SLIP* — Serial Line Internet Protocol — для передачи данных по телефонным линиям;
- *PPP* — Point to Point Protocol — протокол обмена данными «точка-точка»;
- *RPC* — Remote Process Control — протокол управления удаленными процессами;
- *TFTP* — Trivial File Transfer Protocol — простой протокол передачи файлов;
- *DNS* — Domain Name System — система доменных имен;
- *RIP* — Routing Information Protocol — протокол маршрутизации.

На каждом из уровней схемы рис. 6.4 коммуникация осуществляется физически блоками (пакетами), и при переходе с уровня на уровень реализуются следующие преобразования форматов: *инкапсуляция/экскапсуляция; фрагментация/дефрагментация.*

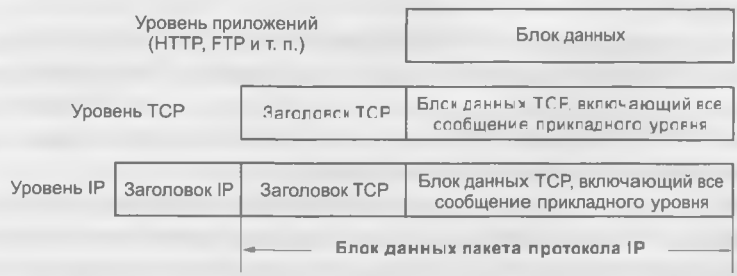


Рис. 6.4. Инкапсуляция протоколов верхнего уровня в протоколы TCP/IP

Инкапсуляция — способ упаковки данных в формате вышестоящего протокола в формат нижестоящего протокола. При этом один или несколько первичных пакетов преобразуются в один вторичный пакет и снабжаются управляющей информацией, характерной для принимающего уровня.

Фрагментация — реализуется, если разрешенная длина пакета нижнего уровня недостаточна для размещения первичного пакета, при этом осуществляется «нарезка» пакетов (например, на пакеты SLIP или фреймы PPP), аналогично при возврате на первичный уровень пакет должен быть *дефрагментирован.*

Протоколы канального уровня SLIP и PPP. Применяются при использовании телефонных каналов.

Serial Line IP (SLIP) обычно применяют как на выделенных, так и на коммутируемых линиях связи со скоростью передачи от 1200 до 19200 бит в секунду.

В рамках протокола SLIP осуществляется фрагментация IP-пакетов, при этом SLIP-пакет должен начинаться символом ESC (восьмеричное 333 или десятичное 219) и заканчиваться символом END (восьмеричное 300 или десятичное 192).

PPP (Point to Point Protocol) — протокол типа «точка-точка» — обеспечивает стандартный метод взаимодействия двух узлов сети. Предполагается, что обеспечивается двунаправленная одновременная передача данных. Как и в SLIP, данные разбиваются на пакеты, которые передаются от узла к узлу упорядоченно. В отличие от SLIP PPP позволяет одновременно передавать по линии связи пакеты различных протоколов. Кроме того, PPP предполагает процесс автоконфигурации обеих взаимодействующих сторон. Собственно говоря, PPP состоит из трех частей: механизма инкапсуляции (encapsulation), протокола управления соединением (link control protocol) и семейства протоколов управления сетью (network control protocols).

Под *датаграммой* в PPP понимается информационная единица сетевого уровня (применительно к IP — IP-пакет). Под *фреймом* понимают информационную единицу канального уровня (согласно модели OSI). Для обеспечения быстрой обработки информации длина фрейма PPP должна быть кратна 32 битам.

Фрейм состоит из *заголовка и хвоста,* между которыми содержатся данные. Датаграмма может быть инкапсулирована в один или несколько фреймов (рис. 6.5). Обычно каждому пакету ставится в соответствие один фрейм, за исключением тех случаев, когда канальный

уровень требует большей фрагментации данных или, наоборот, объединяет пакеты для более эффективной передачи.

Рис. 6.5. PPP-фрейм

Протокол управления соединением предназначен для установки соглашения между узлами сети о параметрах инкапсуляции (размер фрейма и т. п.), кроме того, он позволяет проводить идентификацию узлов. Первой фазой установки соединения является *проверка готовности физического уровня* передачи данных. При этом такая проверка может осуществляться периодически, позволяя реализовать механизм автоматического восстановления физического соединения, как это бывает при работе через модем по коммутируемой линии. Если физическое соединение установлено, то узлы начинают обмен пакетами протокола управления соединением, настраивая параметры сессии. Любой пакет, отличный от пакета протокола управления соединением, не обрабатывается во время этого обмена. После установки параметров соединения возможен переход к *идентификации*. После всех этих действий происходит *настройка параметров работы* с протоколами межсетевого обмена (IP, IPX и т. п.). Для каждого из них используется свой протокол управления. Для завершения работы по протоколу PPP по сети передается пакет завершения работы протокола управления соединением.

Межсетевые протоколы. Протокол *IP* является основным в иерархии протоколов TCP/IP и используется для управления рассылкой TCP/IP-пакетов по сети Internet. Среди различных функций, возложенных на IP, обычно выделяют следующие:

- определение пакета, который является базовым понятием и единицей передачи данных в сети Internet. Некоторые авторы называют такой IP-пакет датаграммой;
- определение адресной схемы, которая используется в сети Internet;
- передача данных между канальным уровнем (уровнем доступа к сети) и транспортным уровнем (другими словами, преобразование транспортных датаграмм во фреймы канального уровня);
- маршрутизация пакетов по сети, т. е. передача пакетов от одного шлюза к другому с целью передачи пакета машине-получателю;
- фрагментация и дефрагментация пакетов транспортного уровня.

Таким образом, вся информация о пути, по которому должен пройти пакет, определяется по состоянию сети в момент прохождения пакета. Эта процедура называется *маршрутизацией* в отличие от коммутации, используемой для предварительного установления маршрута следования отправляемых данных.

Маршрутизация представляет собой ресурсоемкую процедуру, так как предполагает анализ каждого пакета, который проходит через шлюз или маршрутизатор, в то время как при коммутации анализируется только управляющая информация, устанавливается канал (физический или виртуальный), и все пакеты пересылаются по этому каналу без анализа маршрутной информации. Однако при неустойчивой работе сети пакеты могут пересылаться по различным маршрутам и затем собираться в единое сообщение. При коммутации путь придется устанавливать заново для каждого пакета, и при этом потребуется больше накладных затрат, чем при маршрутизации.

Структура пакета IP включает заголовок пакета, где определены все основные данные, необходимые для перечисленных выше функций протокола IP: адрес отправителя, адрес получателя, общая длина пакета и тип пересылаемой датаграммы. Используя данные заголовка, машина может определить, на какой сетевой интерфейс отправлять пакет. Если IP-адрес получателя принадлежит одной из ее сетей, то на интерфейс этой сети пакет и будет отправлен, в противном случае пакет отправят на другой шлюз.

Протоколы управления маршрутизацией. Протокол *RIP (Routing Information Protocol)* предназначен для автоматического обновления таблицы маршрутов, при этом используется информация о состоянии сети, которая рассылается маршрутизаторами (routers). В соответствии с протоколом RIP любая машина может быть маршрутизатором. При этом все маршрутизаторы делятся на активные и пассивные. Активные маршрутизаторы сообщают о маршрутах, которые они поддерживают в сети. Пассивные маршрутизаторы читают эти широковещательные сообщения и исправляют свои таблицы маршрутов, но при этом сами информации в сеть не предоставляют. Обычно в качестве активных маршрутизаторов выступают шлюзы, а в качестве пассивных — обычные машины (hosts).

Протоколы транспортного уровня. *User Datagram Protocol (UDP)* — один из двух протоколов транспортного уровня, используемых в стеке протоколов TCP/IP. UDP позволяет прикладной программе передавать свои сообщения по сети с минимальными издержками, связанными с преобразованием протоколов уровня приложения в про-

токол IP. Однако при этом прикладная программа сама должна обеспечивать подтверждение того, что сообщение доставлено по месту назначения. Заголовок UDP-датаграммы (сообщения) имеет вид, показанный на рис. 6.6.

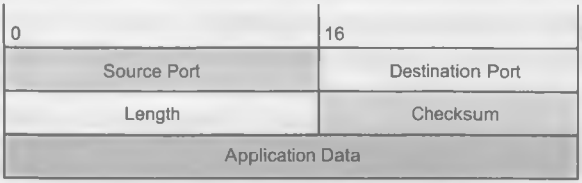


Рис. 6.6. Структура заголовка UDP-сообщения

Transfer Control Protocol (TCP) используется в том случае, когда контроль качества передачи данных по сети имеет особое значение для приложения. Этот протокол также называют *надежным, ориентированным на соединение, потокоориентированным*. Рассмотрим формат передаваемой по сети датаграммы (рис. 6.7). Согласно этой структуре в TCP, как и в UDP, используются порты. В поле *Sequence Number* определен номер пакета в последовательности пакетов, которая составляет сообщение, затем идет поле подтверждения *Asknowledgment Number* и другая управляющая информация.

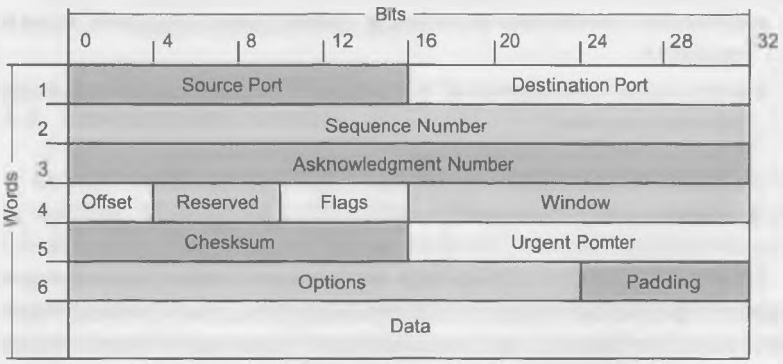


Рис. 6.7. Структура пакета TCP

Надежность TCP обеспечивается тем, что источник данных повторяет их передачу, если только не получит в определенный промежуток времени от адресата подтверждение об их успешном получении. Этот механизм называется *Positive Asknowledgement with Re-transmission (PAR)*.

Далее будут рассмотрены основные протоколы прикладного уровня, обеспечивающие доступ к *информационным ресурсам Internet* (и не только к ним), а также соответствующее программное обеспечение (программы-клиенты и программы-серверы), в том числе:

- протокол эмуляции терминала Telnet;
- протоколы электронной почты SMTP, UUCP;
- протоколы распределенных файловых систем — NNTP, Gopher, FTP;
- протокол гипертекстового доступа к WWW — HTTP.

Каждый из перечисленных протоколов предполагает наличие некоторой совокупности команд (командный язык), которыми обмениваются программы-клиенты и программы-серверы данного протокола. Естественно, целью такого взаимодействия является обмен пользовательскими данными.

Кроме того, к протоколам прикладного уровня Internet относится Z39.50 — протокол управления поиском в распределенных базах данных.

6.3. Прикладные протоколы коммуникации Internet

К данной группе протоколов относятся:

- протокол эмуляции терминала ***Telnet*** (коммуникация в режиме онлайн);
- протоколы электронной почты ***SMTP***, ***UUCP*** (коммуникация в режиме офлайн).

6.3.1. Telnet

Протокол эмуляции удаленного терминала Telnet — одна из самых старых информационных технологий Internet. Этот протокол может быть использован и для организации взаимодействий «терминал—терминал» (связь) и «процесс—процесс» (распределенные вычисления)

Клавиатура должна иметь возможность ввода всех символов ASCII, а также генерировать следующие специальные стандартные команды управления терминалом (эти команды могут или присутствовать в реальном терминале, и тогда они должны представляться в

стандартной форме, или отсутствовать, и тогда заменяться командой *NO* (No-Operation)):

IP — Interrupt Process (прервать процесс). Данная команда реализует стандартный для многих систем механизм прерывания процесса выполнения задачи пользователя;

AO — Abort Output (прервать процесс выдачи). Отличие от выполнения IP, в котором не происходит очистка буфера вывода, т. е. процесс может быть остановлен, а буфер вывода будет продолжать передаваться на экран;

AYT — Are You There (вы еще здесь?). Назначение этой команды — дать возможность пользователю убедиться, что в процессе работы по медленным линиям он не потерял связи с удаленной машиной;

EC — Erase Character (удалить символ). Обеспечивает возможность редактирования командной строки путем введения символов «забой» или удаления последнего напечатанного символа на устройстве отображения;

EL — Erase Line (удалить строку). Команда аналогична EC, но удаляет строку ввода целиком;

open host [port] — начать telnet-сессию с машиной host по порту port. Адрес машины можно задавать как в форме IP-адреса, так и в форме доменного адреса;

close — завершить telnet-сессию и вернуться в командный режим;

quit — завершить работу telnet.

6.3.2. Электронная почта

Электронная почта (ЭП) является самым массовым средством электронных коммуникаций Internet, через нее можно получить доступ практически ко всем ресурсам Internet, а также к информационным ресурсам других сетей.

При коммуникации в режиме ЭП корреспонденция готовится пользователем посредством программы подготовки почты, включающей текстовой редактор. Затем следует вызвать программу отправки почты (программа подготовки почты вызывает программу отправки автоматически). Для работы электронной почты в Internet используется протокол прикладного уровня SMTP (Simple Mail Transfer Protocol), который использует транспортный протокол TCP. Однако совместно с этим протоколом может использоваться и UUCP.

При работе по протоколу SMTP почта реально отправляется только тогда, когда установлено интерактивное соединение с программой-сервером на машине — получателе почты. При этом происходит обмен командами между клиентом и сервером протокола SMTP в режиме on-line, и почта достигает почтового ящика получателя за считанные минуты.

При использовании UUCP почта передается по принципу «Stop-Go», т. е. почтовое сообщение передается по цепочке почтовых серверов, пока не достигнет машины-получателя, что позволяет доставлять почту по плохим телефонным каналам, поскольку не требуется поддерживать связь все время доставки от отправителя к получателю.

При смешанной адресации доставка почты происходит по смешанному сценарию. О том, как шла доставка и как маршрутизировалось сообщение, можно узнать из заголовка сообщения, которое вы получили.

Согласно схеме почтового обмена взаимодействие между участниками этого обмена строится по схеме «клиент — сервер». При этом схему можно подразделить на несколько этапов. Первый — взаимодействие по протоколу SMTP между почтовым клиентом (Internet Mail, Netscape Messager, Eudora и т. п.) и почтовым транспортным агентом (sendmail, smail, ntmail и т. п.), второй — взаимодействие между транспортными агентами в процессе доставки почты получателю, результатом которого является доставка почтового сообщения в почтовый ящик пользователя, и третий — выборка сообщения из почтового ящика пользователя (на сервере) почтовым клиентом и передача в почтовый ящик пользователя на машине пользователя по протоколу POP3 или IMAP.

Протоколы обслуживания электронной почты

Наиболее распространенными являются протоколы SMTP, POP3, IMAP.

Протокол SMTP *(Simple Mail Transfer Protocol)*. Не зависит от транспортной среды и может использоваться для доставки почты в сетях с протоколами, отличными от TCP/IP и X.25. При этом отправитель инициирует соединение и посылает запросы на обслуживание, выступая в роли клиента, а получатель отвечает на эти запросы (выполняя функции сервера).

После того как определены отправитель и получатель, можно отправлять сообщение командой `data`, которая вводится без парамет-

ров и идентифицирует начало ввода почтового сообщения. Сам протокол SMTP не накладывает каких-либо ограничений на информацию, которая заключена между командой `data` и «.» в первой позиции последней строки.

В протоколе SMTP поддерживается и *прямая рассылка сообщений*. В этом случае сообщение будет отправляться не в почтовый ящик, а непосредственно на терминал пользователя, если пользователь в данный момент находится за своим терминалом.

Протокол обмена почтовой информацией POP3 *(Post Office Protocol, версия 3)*. Предназначен для пересылки почты из почтовых ящиков пользователей (на сервере) на их рабочие места при помощи программ-клиентов. Если по протоколу SMTP пользователи отправляют корреспонденцию через Internet, то по протоколу POP3 пользователи получают корреспонденцию из своих почтовых ящиков на почтовом сервере в локальные файлы, однако сообщения можно принимать, но нельзя отправлять. Формально взаимодействие по протоколу POP3 можно разделить на две фазы: фазу аутентификации и фазу обмена данными.

Протокол IMAP *(Interactive Mail Access Protocol)*. Представляет собой более надежную альтернативу протоколу POP3 и к тому же обладает более широкими возможностями по управлению процессом обмена с сервером. Главное отличие от POP состоит в возможности поиска нужного сообщения и осуществления разбора заголовков сообщения.

Интерфейсные программы (почтовые клиенты)

Простейшей и самой распространенной программой подготовки и отправки почты в режиме командной строки является *mail*. После ввода почтового адреса программа выдаст предложение ввести тему сообщения, а после ввода пользователем темы (subject) программа перейдет на следующую строку и будет ждать текста сообщения. Для завершения ввода сообщения следует нажать <Ctrl+D> (конец ввода), после чего сообщение будет отправлено. Чтобы прочитать сообщения, необходимо выполнить команду mail без аргументов.

Начиная с Windows 95 в состав операционных систем включена программ-клиент *MS Outlook Express*, которая предназначена для работы с электронной почтой и новостями. Для чтения электронной почты из программы Outlook Express необходимо, чтобы используемая система обмена сообщениями поддерживала протоколы SMTP и

POP3 или IMAP. Программу Outlook Express можно использовать для чтения групп новостей, таких как Usenet. Работа с группами новостей осуществляется через серверы новостей NNTP.

Outlook Express включает в себя программу адресной книги Windows. Данная программа предоставляет широкие возможности управления контактными данными, включая создание групп контактов и папок для сортировки сообщений и размещения адресов электронной почты. Адресная книга Windows обеспечивает доступ к каталогам Internet, использующим протокол LDAP. Каталоги Internet облегчают поиск обычных адресов и адресов электронной почты. В программе адресной книги уже настроен доступ к нескольким популярным каталогам.

Можно набрать любой телефонный номер, указанный в адресной книге, используя программу номеронабирателя, установленную на компьютере. Программа Outlook Express может сохранять незаконченные сообщения в папке Черновики, а отправленные сообщения — в папке Отправленные на сервере IMAP. Можно редактировать гипертекстовые (HTML) сообщения и использовать в них теги HTML.

Программа может определять, произошел ли обрыв телефонного соединения или отключение компьютера от локальной сети. Программа Outlook Express может восстановить разорванное соединение автоматически либо после подтверждения, вводимого пользователем. Возможно установить параметры правил (автоматического подключения, проверку почты, подключение к поставщику услуг и др.).

6.4. Распределенные файловые системы Internet

Система архивов FTP

FTP-архивы — это распределенный депозитарий разнообразных данных, представленных в виде файлов.

Информация в FTP-архивах разделена на три категории:

- *защищенная информация*, режим доступа к которой определяется ее владельцами и разрешается по специальному соглашению с потребителем;
- *информационные ресурсы ограниченного использования*, к которым относятся, например, ресурсы ограниченного времени использования или ограниченного времени действия, т. е. потребитель

может использовать текущую версию на свой страх и риск, но никто не будет оказывать ему поддержку;

- *свободно распространяемые информационные ресурсы* — все, что можно свободно получить по сети без специальной регистрации. Это могут быть документация, программы или что-либо еще.

Протокол FTP (File Transfer Protocol) — один из старейших протоколов в Internet; обмен данными в FTP проходит по TCP-каналу и построен по технологии «клиент-сервер»

В FTP соединение инициируется интерпретатором протокола пользователя. Управление обменом осуществляется по каналу управления в стандарте протокола Telnet. Команды FTP генерируются интерпретатором протокола пользователя и передаются на сервер. Ответы сервера отправляются пользователю также по каналу управления.

Команды FTP определяют параметры канала передачи данных и самого процесса передачи. Они также определяют и характер работы с удаленной и локальной файловыми системами.

В протоколе большое внимание уделяется различным способам обмена данными между машинами разных архитектур, которые могут иметь различную длину слова и часто различный порядок битов в слове. Кроме того, различные файловые системы работают с разной организацией данных.

Практически для любой платформы и операционной среды существуют как серверы, так и клиенты. Ниже описываются стандартные сервер и клиент Unix-подобных систем.

Для работы с FTP-архивами необходимо следующее программное обеспечение: сервер, клиент и поисковая программа. Сервер обеспечивает доступ к ресурсам архива из любой точки сети, клиент — доступ пользователя к любому архиву в сети, а поисковая система — навигацию во всем множестве архивов сети.

Функции FTP-клиента встроены, например, в программную оболочку Windows Commander. Конфигурации каждой настройки включают в себя

- адрес FTP-сервера;
- имя пользователя и пароль;
- имя удаленного каталога в файловой системе FTP.

После установления связи на одной из панелей отображается удаленный каталог.

Передача файлов в обе стороны (Upload и Download) осуществляется обычным выделением файлов (директорий) и копированием их по команде <F5>.

Usenet

Распределенная файловая система Usenet — система телеконференций Internet. (Данный термин не очень удачен — в Internet есть и другие средства, которые также реализуют принцип телеконференций.) Пользователи Usenet предпочитают придерживаться термина *newsgroup*, или *group*, который можно перевести как *группа новостей* (*группа*) — это постоянно изменяющийся набор сообщений, входящих в область интересов участников данной группы. *Статья*, или *сообщение*, отправляется в телеконференцию пользователем и становится доступной для всех подписчиков группы. *Подписка* подразумевает процедуру оповещения пользователя о появлении новых статей по интересующей его теме. Сообщение оформляется в соответствии со стандартом почтового сообщения Internet.

Имена доменов для групп новостей могут принадлежать к одной из нескольких категорий верхнего уровня, как, например, comp — темы, посвященные компьютерам; misc — темы, не относящиеся ни к одной из стандартных категорий; news — информация о группах новостей; rec — различные развлекательные темы и т. п.

С точки зрения структуры информационного ресурса Usenet организована как иерархический каталог, узлами которого являются группы новостей. Сообщения в группе обычно не задерживаются более нескольких дней.

Пользователь осуществляет подписку на одном из серверов Usenet, который территориально ближе для него (обычно это машина, на которой расположены все информационные ресурсы организации или учебного заведения). По мере поступления новых сообщений от пользователей серверы обмениваются между собой этой информацией.

Протокол обмена новостями NNTP пришел на смену UUCP, и его целью было упорядочить обмен информацией между серверами Usenet.

Протокол NNTP определяет запросно-ответный механизм обмена сообщениями между серверами и сервером и программами-клиентами. Для этой цели в протоколе определен набор команд и ответов на них. Весь диалог осуществляется в текстах ASCII, причем каждая команда состоит из идентификатора и параметров.

Для работы по протоколу NNTP разработано довольно много программ просмотра новостей. Как правило, программы используют для просмотра меню и реализуют просмотр новостей в режиме скрол-

линга. Наиболее популярным методом доступа в Usenet остается электронная почта. Подготавливая почтовое сообщение, пользователь фактически записывает сценарий диалога с сервером NNTP, который затем будет выполнен почтовым роботом Usenet.

Gopher

Файловая система Gopher была разработана для реализации распределенной базы документов, которые хранятся на машинах сети и предоставляются пользователю в виде единой иерархической файловой системы. Модель файловой системы наилучшим образом подходит для отображения структуры хранения документов по следующим очевидным соображениям:

- иерархическое представление данных привычно большинству пользователей;
- Gopher рассчитан на применение недорогих решений как в аппаратной части, так и при программировании, поскольку первоначально он был ориентирован на разработку информационной системы университета (шт. Миннесота);
- модель файловой системы может быть легко расширена путем добавления к традиционным файлам и директориям других объектов, которые можно назвать *виртуальными файлами*. Такие виртуальные объекты могут быть поисковыми запросами, или шлюзами, в другие информационные ресурсы Internet.

Gopher представляет весь Internet (серверы Gopher) в виде единой иерархической системы.

Протокол Gopher предназначен для работы по модели «клиент — сервер», при этом программа-клиент установлена на рабочем месте пользователя. После ответа сервера соединение разрывается, а при новом запросе оно должно быть установлено заново.

Возвращаемый сервером текст представляет собой справку о содержании текущей директории (каталога), каждый элемент которой включает:

- тип (объекта в директории);
- имя (используется для отображения и в запросах);
- неотображаемую строку выбора, которая обычно описывает путь, используемый удаленным хостом для доступа к объекту (селектор);
- имя хоста (машины, к которой надо обращаться за информацией);

- номер порта (на котором сервер данного объекта ожидает запрос).

Для использования поискового объекта из директории Gopher клиент посылает запрос специальному поисковому серверу Gopher, а получает от сервера список адресов документов, удовлетворяющих запросу.

6.5. Распределенные информационные системы Internet

Файловые системы Internet, рассмотренные выше, во многом аналогичны файловым системам операционных систем ЭВМ (UNIX, DOS и пр.), которые они, собственно, и имитируют. *Навигация* в таких структурах весьма ограничена — «вверх» и «вниз» по ветвям каталогов (директорий). Поиск информации почти исключен, поскольку связь между содержанием данных и наименованиями файлов или каталогов весьма ограничена. Альтернативными подходами является организация информационных систем, позволяющих проводить содержательный поиск данных в распределенной БД. Применительно к INTERNET такими технологиями являются WWW (World Wide Web) и Z39.50.

6.5.1. Информационные технологии WWW

Основными компонентами данных технологий, состоящих в применении гипертекстовой модели к информационным ресурсам, распределенным в Internet, являются (рис. 6.8):

- HTML — язык гипертекстовой разметки документов;
- URL — универсальный способ адресации ресурсов в сети;
- HTTP (HyperText Transfer Protocol) — протокол обмена гипертекстовой информацией;
- а также дополнительные средства (CGI, Java, JavaScript).

Ранее (в главе 2) уже были рассмотрены основные возможности HTML как приложения SGML к описанию типов документов. Здесь мы вкратце остановимся на навигационных компонентах HTML.

Гипертекстовая база данных в концепции WWW — это набор текстовых файлов, написанных на языке HTML, который определяет форму представления информации (разметка) и структуру связей этих файлов (гипертекстовые ссылки).

Рис. 6.8. Архитектура WWW-технологии

Такой подход предполагает наличие еще одного компонента технологии — интерпретатора языка. В WWW функции интерпретатора разделены между сервером гипертекстовой базы данных и интерфейсом пользователя.

Сервер, кроме обеспечения доступа к документам и реализации гипертекстовых ссылок, осуществляет также препроцессорную обработку документов, в то время как интерфейс пользователя проводит интерпретацию конструкций языка, связанных с представлением информации.

Язык HTML включает 2 основных компонента:

- средства отображения документа (рассмотрены в главе 2);
- средства навигации и построения интерфейсов с пользователем.

Гипертекстовые ссылки

Все рассмотренные ранее средства управления отображением текста являются дополнительными к основным элементам документа — гипертекстовым ссылкам. Вот некоторые элементы HTML, реализующие данный механизм.

LINK — элемент *заголовка* — может быть использован для описания общих для всего документа гипертекстовых ссылок. Элемент имеет три атрибута: REL, REV и HREF. REL задает тип ссылки, REV задает обратную ссылку, а HREF определяет ссылку в форме URL. На данный элемент возложена нагрузка по программированию средств управления интерфейсом пользователя.

При выборе соответствующей позиции в меню или соответствующей пиктограммы программа интерфейса должна генерировать запрос к серверу на получение документа, указанного в атрибуте HREF (HyperText Reference). Например:

```
<LINK REL=Help  HREF="http://polyn.net.kiae.su/dss/syshelp.html">
```

Данное предложение в заголовке HTML-документа означает, что при выборе режима «Help» на экране отобразится документ, который хранится по адресу `http://polyn.net.kiae.su/dss/syshelp.html`. Таким образом, появляется возможность строить системы контекстно-зависимых справок в интерфейсах, построенных по технологии WWW.

Элемент `<A>...</A>`, или *«якорь» (anchor)*, применяется для записи гипертекстовой ссылки из *тела документа*; имеет несколько атрибутов, главным из которых является HREF. Простую ссылку можно записать в виде:

```
<A HREF="http://polyn.net.kiae.su/index.html">Индекс базы
данных "Полынь"</A>.
```

Здесь значением атрибута HREF является записанный в формате URL адрес документа «index.html» на машине «polyn.net.kiae.su», доступ к которой осуществляется по протоколу HTTP.

Представление multimedia-информации

Система WWW была ориентирована на графические средства представления информации. Первым шагом на этом пути была реализация возможности вставлять в текст графические объекты, затем появилась возможность запуска внешней программы для просмотра файла в форматах, отличных от ASCII (например, GIF). Таким образом, на любой информационный объект можно сослаться из документа HTML, вызвав его через внешнюю программу просмотра. Графические объекты могут использоваться в качестве идентификаторов гипертекстовых ссылок и для перехода по гипертекстовой сети.

Для встраивания в документ графических образов используются элементы IMG и FIG.

Пример использования элемента *IMG*:

```
<IMG SRC="http://polyn.net.kiae.su/gif/sarclast.gif"
ALT="Sarcofagus.Winter, 1997">
```

В данном примере атрибут SRC определяет адрес графического объекта, который надо встроить в документ, а атрибут ALT предназначен для отображения в интерфейсах, которые не поддерживают встраиваемую графику, чтобы вместо картинки отображалось содержание атрибута ALT.

IMG можно использовать внутри гипертекстовой ссылки:

```
<A HREF="doc.html"><IMG SRC="icon.gif" ALIGN=RIGHT></A>
```

В этом случае весь рисунок целиком используется как идентификатор гипертекстовой ссылки. Кроме того, в данном примере используется атрибут элемента IMG — ALIGN, который может принимать значения TOP, MIDDLE, BOTTOM, LEFT, RIGHT и определяет, где относительно других символов текста в строке будет располагаться рисунок.

Элемент *FIG* (развитие IMG) введен в стандарт языка для улучшения отображения графической информации и использования ее для разработки гипертекстовых баз данных. При использовании IMG текст разбивается на две части — до рисунка и после, при этом реализуется обтекание картинки текстом.

Элементы реализации интерактивных интерфейсов в HTML

ISINDEX — элемент *заголовка документа* — определяет использование HTML-документа для ввода запроса на поиск по ключевым словам:

```
<ISINDEX HREF="http://polyn.net.kiae.su/cgi-bin/search"
PROMPT="Enter Keywords:">
```

В приведенном примере атрибут HREF определяет адрес программы обработки запроса, а атрибут PROMPT — содержание приглашения.

FORM — средства встраивания элементов интерфейса в *тело документа*. Посредством форм осуществляется передача параметров внешним программам, которые вызываются сервером, что сделало WWW универсальным интерфейсом ко всем ресурсам сети.

Некоторые вложенные в FORM элементы HTML представлены в табл. 6.6.

INPUT — наиболее универсальный из всех элементов формы. Способ его отображения определяется атрибутом TYPE, который мо-

Таблица 6.6. **Элементы интерфейса**

Элемент	Назначение
INPUT	Поля ввода информации имеют множество типов
TEXTAREA	Поле ввода многострочного текста
SELECT	Описание меню
OPTION	Описание элемента меню

жет принимать значения: `text`, `password`, `checkbox`, `radio`, `submit`, `reset` и др. `Submit` активирует передачу параметров (значений переменных) серверу, в то время как `Reset` восстанавливает значения полей формы по умолчанию.

Элементы *SELECT и OPTION* предназначены для организации меню, которое может быть выпадающим, множественным и графическим.

HTTP (Hypertext Transfer Protocol)

HTTP — протокол прикладного уровня, который разработан для обмена гипертекстовой информацией в сети Internet.

Протокол реализует принцип «запрос/ответ». Запрашивающая программа — клиент — инициирует взаимодействие с отвечающей программой — сервером — и посылает запрос, включающий в себя метод доступа, адрес URI, версию протокола, сообщение с модификаторами типа передаваемой информации, информацию клиента и, возможно, тело сообщения клиента. Сервер отвечает строкой состояния, включающей версию протокола и код возврата, за которой следует сообщение в формате, аналогичном MIME. Данное сообщение содержит информацию сервера, метаинформацию и тело сообщения. В принципе одна и та же программа может выступать как в роли сервера, так и в роли клиента (что и происходит при использовании proxy-серверов).

Программа-клиент посылает после установления соединения *запрос серверу*. Этот запрос может быть в двух формах: *в форме полного запроса и в форме простого запроса.*

В настоящее время в практике WWW реально используются три метода: GET, HEAD, POST.

GET — метод, позволяющий получить данные, заданные в форме URI в запросе ресурса. Если ссылаются на программу, то возвращает-

ся результат выполнения этой программы, но не ее текст. Дополнительные данные, которые надо передать для обработки, кодируются в запрос ресурса.

HEAD — в отличие от GET не возвращает тела ресурса. Используется для получения информации о ресурсе и для тестирования гипертекстовых ссылок.

POST — метод разработан для передачи большого объема информации на сервер. Им пользуются для аннотирования существующих ресурсов, посылки почтовых сообщений, работы с формами интерфейсов к внешним базам данных и внешним исполняемым программам. В отличие от GET и HEAD в POST передается тело ресурса, которое является значениями полей форм или других источников ввода.

Ответ сервера может быть, как и запрос, упрощенным или полным. При упрощенном ответе сервер возвращает только тело ресурса (например, текст HTML-документа). При полном ответе клиенту возвращаются строка состояния (Status-Line), общий заголовок, заголовок ответа, заголовок ресурса и тело ресурса. Строка состояния состоит из версии протокола, кода возврата и краткого описания этого кода. Заголовок ответа сервера может состоять из адреса URI запрашиваемого ресурса, и/или наименования программы сервера, и/или кода идентификации для работы в защищенном режиме. Состав полей заголовка ресурса является общим и для запроса клиента, и для ответа сервера и состоит из разрешения на метод доступа, типа кодировки, тела ресурса (содержания ресурса), длины тела ресурса, типа ресурса, времени действия данной копии ресурса, времени последнего изменения ресурса и расширения заголовка.

6.5.2. Программное обеспечение для World Wide Web

Программное обеспечение WWW можно разделить на группы по направлениям использования. Каждое из этих направлений определяется либо схемой взаимодействия компонентов Web-технологии, либо особенностями применения субъектами обмена информацией в рамках WWW. Принята следующая классификация программного обеспечения WWW:

- программы-клиенты (в том числе мультипротокольные программы-браузеры);
- программы просмотра документов в форматах, отличных от стандартных форматов Web;

- программы-серверы протокола обмена гипертекстовой информацией (Web-серверы);
- программы подготовки публикаций;
- поисковые машины;
- программы анализа статистики посещений.

Программы-клиенты. Наиболее распространенными являются *мультипротокольные программы-браузеры*. В настоящее время на роль стандартов в этом классе программного обеспечения претендуют две программы: *Mozilla Firefox* (продолжение линии Netscape Communicator) и *Microsoft Internet Explorer*. По своим возможностям и внешнему оформлению они довольно похожи. Основная задача этих программ — интерпретация разметки на языке HTML, интерпретация встроенных в HTML программ на одном из командных языков Web (JavaScript или VBScript), интерпретация Java — байт кодов, разбор спецификации ресурсов сети (обработка URI), взаимодействие с серверами по протоколам прикладного уровня стека протоколов TCP/IP.

Данные программы являются функционально полными браузерами и могут не только отображать статические страницы, но и работать с динамическими страницами, содержащими многокадровые картинки, JavaScript-программы и Java-коды.

Программы-серверы. *Сервер WWW* — программа, которая принимает запросы от WWW-клиентов и отвечает на них. В качестве ответа может быть возвращен HTML-документ, хранящийся в базе данных сервера, графический образ, аудиозапись, фильм или ответ внешней программы. Сервер обменивается данными не только с клиентами, но и с CGI-скриптами.

В настоящее время серверы WWW существуют для всех типов компьютерных платформ и операционных систем.

6.5.3. Протокол Z39.50

Протокол ориентирован на информационный поиск в базах данных. Это протокол прикладного уровня в рамках семиуровневой эталонной модели взаимодействия открытых систем, разработанной Международной организацией стандартов (ISO), и поэтому может быть реализован в различных типах сетей (например, в сетях TCP/IP, IPX/SPX, OSI), независимо от реализации транспортного уровня. Его назначение — предоставить компьютеру, работающему в режиме

«клиент», возможности поиска и извлечения информации из другого компьютера, работающего как информационный сервер.

Особенностями протокола Z39.50 является возможность сохранения состояний системы и присвоение каждому состоянию соответствующего идентификатора. Эта особенность протокола позволяет производить «навигацию во времени», т. е. в любой момент можно вернуться в определенную точку поиска, произведенного ранее. Наличие такой «памяти» позволяет использовать результаты, полученные ранее, в составлении дальнейших запросов.

Протокол также позволяет выполнять параллельные операции поиска, уведомлять пользователя о состоянии сервера, сортировать данные на сервере, получать информацию о подключенных базах, наборах атрибутов, синтаксисах записей и т. д. Для описания баз данных внутри протокола был создан соответствующий формат описания.

Первоначально многие Z39.50-приложения создавались исключительно для использования с библиографическими данными (например, электронные онлайн-версии библиотечных каталогов). Однако в настоящее время протокол развит настолько, что позволяет обрабатывать различные данные — финансовую, химическую, техническую информацию, тексты и изображения.

Технология сетевого доступа к базам данных по протоколу Z39.50 существенно отличается от других технологий. Отличие обусловлено самой сутью протокола: его ориентацией на работу с базами данных, абстрагированными от конкретных систем.

В основе Z39.50 лежит модель *абстрактной базы данных*. Каждый элемент этой модели имеет описание с однозначным толкованием и стандартизуется с присвоением уникального идентификатора.

Модель службы Z39.50 предусматривает обмен сообщениями типа «запрос—ответ» между соответствующими приложениями — клиентом и сервером. Формат таких сообщений и определяется протоколом Z39.50. После установления TCP-соединения (или любого другого, зависящего от способа передачи данных) устанавливается Z39.50-соединение.

Получив от клиента запрос на инициализацию сессии, сервер формирует ответ — сообщения о параметрах сеанса, видах услуг, поддерживаемых клиентом и сервером, после получения которого клиентом Z39.50-соединение считается установленным. Далее клиент может либо продолжить работу с такими параметрами, либо закрыть соединение и попытаться затем установить новое — быть может, с другими параметрами, и передавать запрос на поиск информации.

Таким образом, протокол Z39.50 описывает *интерактивную сессию* между источником запросов и приемником, обслуживающим эти запросы.

Контрольные вопросы

1. Что такое архитектура «клиент—сервер» и каковы основные разновидности программно-аппаратных средств на клиентской и серверной стороне?
2. Дайте определение протокола в информационных сетях.
3. В чем преимущества систем с коммутацией пакетов?
4. Определите семиуровневую модель протоколов в открытых системах.
5. На что ориентированы протоколы 1—3-го уровня в семиуровневой модели OSI?
6. На что ориентированы протоколы 5—7-го уровня в семиуровневой модели OSI?
7. С помощью какого пакета прокладывается путь в сети с датаграммным способом передачи?
8. Какой уровень прокладывает путь через сеть?
9. Какой уровень обеспечивает обнаружение и исправление ошибок?
10. Какой уровень определяет процедуру представления передаваемой информации в нужную сетевую форму?
11. Что входит в систему адресов Internet?
12. Какую структуру имеет адрес Ethernet?
13. Какую структуру имеет IP-адрес?
14. Что такое выделенные IP-адреса?
15. Что из себя представляет система доменных имен?
16. Что такое сервер доменных имен?
17. Какие разновидности URL вам известны?
18. Какие протоколы транспортного уровня вы знаете?
19. Что такое инкапсуляция и фрагментация?
20. Что такое TCP/UDP порт?
21. Что представляют собой протоколы управления маршрутизацией?
22. Какова структура пакета TCP?
23. Что представляет собой ARP?
24. Расставьте на места уровни в архитектуре протокола TCP/IP.

25. Какую функцию описывает протокол TCP?
26. Какую функцию описывает протокол IP?
27. Что такое класс локальной сети, входящей в Internet?
28. Каковы преимущества и недостатки конфигурации «звезда»? В каких локальных сетях она применяется?
29. Каковы преимущества и недостатки конфигурации «общая шина»? В каких локальных сетях она применяется?
30. Каковы преимущества и недостатки конфигурации «кольцо»? В каких локальных сетях она применяется?
31. Какие смешанные топологии вам известны и с помощью какого сетевого оборудования они реализуются?
32. Какие прикладные протоколы Internet вы знаете?
33. Что представляют собой мосты? Дайте классификацию мостов.
34. Какие информационные ресурсы Internet вы знаете?
35. Какова структура ресурса Usenet?
36. Какова структура распределенной ФС FTP?
37. Перечислите команды Telnet.
38. Какие протоколы электронной почты вам известны?
39. Что такое почтовый сервер?
40. Перечислите программы-клиенты электронной почты.
41. Какие разновидности почтовых адресов вам известны?
42. Перечислите команды прикладных протоколов электронной почты.
43. Назовите программы-клиенты и серверы протокола FTP.
44. Каков состав средств Web-технологий?
45. Каков состав программного обеспечения WWW?
46. Перечислите основные программы-клиенты WWW.
47. Расскажите об организации гипертекстовых ссылок.
48. В чем состоит отличие протокола Z39.50 от других прикладных протоколов?

Глава 7
ИНФОРМАЦИОННО-ПОИСКОВЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

В контексте рис. 1.3 место рассматриваемого в этой главе класса ИС соответствует верхнему уровню, т. е. в целом средства ИС должны обеспечить и синтактику, и семантику, и прагматику. Последнее означает, что в сферу ИПС, помимо поиска, входит также согласование результата и условий взаимодействия с внешней средой, в частности, определяемых особенностями человеко-машинного взаимодействия.

7.1. Генерация и использование информационных ресурсов

7.1.1. Введение в информационный поиск

С развитием информационных коммуникаций поиск информации стал для пользователей делом довольно обыденным, но как показывает анализ выражений запросов и действий пользователей профессиональных баз данных научной информации [Голицына, 2007], вряд ли грамотным (с осознанием существа выполняемых действий) и потому — малоэффективным. Причиной такого состояния является не только то, что поиск для человека — это естественная, «встроенная» функция и выполняется скорее интуитивно, сколько то, что среды, в которых этот процесс осуществляется, принципиально различаются. Для сознания человека характерны целостность и образность представления, а также ассоциативность отбора, а для ЭВМ — дискретность и точечность (двоичность) представления, а также четкие алгоритмы идентификации и соотнесения объектов. Автоматизированный информационный поиск, таким образом, должен быть интег-

рирующим процессом, выполняемым в обеих средах, и потому требующим согласования форм представления, методов обработки и средств, обеспечивающих взаимодействие. Для этого наряду с формами и методами представления и обработки информации в машинной среде необходимо обстоятельно рассмотреть особенности порождения, поиска и использования информации в основной деятельности человека. Необходимость такого систематизированного рассмотрения определяется следующими факторами, имеющими как объективную, так и субъективную природу.

1. Целью информационного поиска в большинстве случаев является отыскание документов, содержащих сведения, нужные для решения конкретных управленческих, научных или практических задач, в том числе генерации нового знания. При этом характер информации, способ ее представления может быть самым разным — от объявлений о продаже товаров до интерактивных научных конференций, от технического описания, предназначенного для непосредственного применения, до неформализуемой в явном виде совокупности фактов, приводящих к творческому озарению или принятию неординарного решения. В дальнейшем изложении мы будем придерживаться именно профессиональной точки зрения: отыскиваемая информация предназначена для использования в основной (профессиональной) деятельности и она (точнее, публикации об искомых объектах или каким-то образом связанных с ними) должна быть не только найдена, но также должны быть обоснованы ее полнота, точность и достоверность.

2. Требования к полноте, точности и достоверности информации, характеру процесса поиска, а в большей степени — к выбору типов и набора информационных ресурсов, а также последующей обработке найденного зависят от характера задачи (в том числе особенностей текущего этапа жизненного цикла объекта задачи). Действительно, когда задача сформулирована в сложившейся предметной области и ее актуальность не вызывает сомнений, цель поиска очевидна: найти полноценное изложение метода решения задачи данного типа (например, отчет о НИР, статью, учебник и т. д.), достоверность которого не подлежит сомнениям. Во многих случаях (обычно когда мы ищем уже известный человечеству метод) это можно сделать, не прибегая к сложным процедурам, использующим разнообразные, но *вспомогательные* и, по сути, *вторичные* средства: указатели, реферативно-библиографические БД и т. д. Достаточно пролистать разделы соответствующих учебников или монографий или, в крайнем случае, подшивку специальных журналов. Предложения библиотек и инфор-

мационных служб использовать специальные справочно-поисковые средства (каталоги, реферативно-библиографические БД — так называемую вторичную информацию), а не непосредственно полные тексты[1], кажутся многим современным пользователям абсурдными. Но их использование становится неизбежным, когда собственные «подручные» ресурсы (как доступная коллекция полных текстов, так и отведенные временные ресурсы) не позволяют найти решение, а характер ОД предполагает реальную ответственность (экономическую или юридическую) за принятие решения. Общеизвестными примерами являются задачи патентного поиска, позволяющего подтвердить приоритет изобретения, или научного поиска, доказывающего новизну решения.

Особую роль играет вторичная информация на начальном и заключительном этапах ОД, в бизнес-планировании и в задачах управления качеством. Например, при определении направления деятельности, выборе решения при неполной информации, принятии решения о начале или завершении деятельности, оценке эффективности и применимости, оценке новизны и конкурентоспособности найденного решения. То есть информация такого рода — общее заключение, может быть только синтезирована на основе многоаспектного содержательного и статистического анализа *потока* публикаций, отражающего не только разные подходы к решению, но и разные этапы жизненного цикла идеи.

3. Обычно объектом информационного поиска является *предметное* содержание — данные, методы, инструкции и т. д., позволяющие решить или построить решение конкретной задачи ОД. При этом наиболее распространенной *коммуникативной* формой представления содержания является документ. Документ по своему статусу соответствует «завершенности» процесса ОД: излагаются, так или иначе, *проверенные* решения, *обоснованные* подходы, *принятые* гипотезы. Однако в некоторых случаях, когда исследование не завершено или мы не знаем об этом, будет естественным обратиться к «источнику» — автору, генерирующему новое знание. Отметим, что в информационной практике термин «источник» часто ассоциируется и с конкретными *средствами передачи* информации. С одной стороны, это отдельные публикации (издания, специализирующиеся в данной предметной

[1] Необходимо отметить, что профессиональные БД вторичной научной информации обыкновенно позволяют почти всегда автоматически выйти на полные тексты публикаций.

области), а с другой — организации (издательства, библиотечные коллекторы, книготорговцы), обеспечивающие распространение публикаций, в том числе по тематическому принципу.

4. С технологической точки зрения процесс поиска — это рутинный перебор документов[1], сосредоточенных в традиционных или электронных хранилищах и более или менее полно представляющих интересующую нас тему. Отбор обыкновенно производится по содержанию документов. Однако здесь следует уточнить, что слово «содержание» в этом случае надо понимать условно. Содержание документа представляется в поисковой системе достаточно поверхностно — поисковым образом, перечисляющим основные понятия, которые *система* и использует для «отбора» документов (точнее, для формирования списка ссылок на отобранные документы), и только на следующем шаге *человек*, обращаясь через сформированный системой список, получает собственно содержание документа и осмысливает возможность его использования.

Кроме того, достаточно очевидно, что по отношению к традиционным библиотечным *способы организации массивов и методы автоматизированного поиска* не отличаются принципиальной новизной. Поиск ведется либо путем последовательного просмотра ряда документов[2] до тех пор, пока не будет найдена нужная информация, либо с использованием указателей и каталогов, систематизирующих размещение документов по предметному, алфавитному или какому-либо другому принципу и, соответственно, облегчающих доступ к ним (просто сокращая объем перебора при просмотре).

5. Важным фактором, влияющим на функциональные особенности реализаций информационных систем, являются характер и организация хранения (доступа) информации. В этом смысле системы можно условно разделить на два класса:

- электронные каталоги и документальные ИС (как локальные базы данных);
- поисковые машины (как системы поиска в распределенных массивах).

---

[1] Здесь и далее термин «документ» будет использоваться для обозначения собирательного понятия, связанного с такой формой представления информации, для которой характерна логическая завершенность (цельность содержания), а также физическая доступность и идентифицируемость (т. е. документ всегда имеет структуру, адрес, методы обработки и т. д.).

[2] Что, очевидно, возможно только в случае очень маленького массива документов.

Однако следует отметить, что подобное деление скорее отражает не только единственно возможный на сегодняшний день компромисс практических потребностей и реальных возможностей промышленной реализации, сколько историю развития средств поиска. Первые — документальные ИС — берут начало от библиотечных информационных систем, ориентированных на традиционный каталожный поиск; вторые появились как вспомогательное средство поиска в сетевых средах, изначально предназначаемых для оперативных коммуникаций.

В основном именно электронные каталоги и документальные системы обеспечивают профессиональный поиск информации в локальных или распределенных базах данных. Наиболее известными примерами являются информационные ресурсы ВИНИТИ и ИНИОН РАН, INIS, электронный каталог РГБ или библиотеки Конгресса США, которые обеспечивают, в основном, библиографический и тематический поиск.

Библиографический поиск обеспечивает *выявление* публикаций по их *выходным данным*, например, по именам авторов, датам публикаций и т. д. Основополагающей предпосылкой здесь является фиксированная для конкретной базы данных модель представления информации, в соответствии с которой обеспечивается нормализованная (единообразная) запись элементов данных. Все это требует от пользователя достаточно специальных знаний.

Тематический поиск обеспечивает *отбор* документов по *семантическим признакам*, *обобщенно* представляющим его содержание. Здесь концептуальным положением является то, что содержание документа может быть представлено некоторой совокупностью понятий, характеризующих основной объект. Это позволяет достаточно эффективно использовать запросы в виде компактных комбинаций терминов, обычно двух-трех слов естественного или естественно-научного языка. Такое представление информации хорошо соответствует теоретико-множественным моделям поиска, однако для многих пользователей создает тупиковые ситуации, обусловленные непониманием поискового языка и процесса получения результата.

Отдельным направлением в развитии информационного поиска является *полнотекстовый поиск*, основная цель которого — обеспечить точный отбор за счет применения критериев, основанных на семантических категориях. Но здесь, несмотря на достаточно серьезные достижения в области анализа текста и появление промышленных полнотекстовых поисковых систем (в основном в сфере СМИ), ожи-

дать скорого широкого внедрения систем полнотекстового поиска, в том числе и в области научной информации, не приходится уже хотя бы потому, что выявить и воспринять смысл (и тем более новые идеи) в научных публикациях может не всегда и не всякий естественный интеллект. И уж тем более сомнительно автоматически построить понятийную, точно отражающую смысл структуру по тексту запроса, содержащего максимум три-четыре термина.

Также важным, но часто незамечаемым фактором является реальная ограниченность полноты представления информации в конкретном массиве (точнее, источников информации, которые используются для формирования массива). Это особенно существенно при поиске в Internet: глобальная сеть сетей физически объединяет компьютеры практически всех крупнейших библиотек мира, однако вход пользователя в сеть не приводит автоматически к возможности использовать электронный каталог какой-либо из таких библиотек. То есть подключение к сети обеспечивает физическую доступность вычислительного комплекса, хранящего ресурс, но доступность собственно *информационного ресурса* — обычно совокупности баз данных (документальных и фактографических массивов) и информационных технологий, часто ограничена технологическими, организационными, финансовыми или какими-либо другими условиями.

6. Наконец, следует отметить и некоторые особенности взаимоотношений человека и информационно-поисковых систем (ИПС), обусловленные «заторможенным развитием» последних: ИПС сохранили вопросно-ответную идеологию, свойственную уже ушедшим, традиционным системам пакетного информационного обслуживания, когда гарантом качества поиска был информационный работник. Задачей такого информационного посредника были не только понимание и интерпретация потребностей пользователя, но и выбор ресурса и собственно поиск, что в итоге и обеспечивало эффективность (по крайней мере — профессиональность) поиска. В современных же условиях «информационного самообслуживания» пользователь, привыкший к интуитивному освоению программных средств, большинство из которых имеет существенно более простой интерфейс, часто неадекватно оценивает состояние и результаты поиска. Типичными, но стратегически фатальными ошибками является принятие пользователем безапелляционного решения о «плохой» базе данных или поисковой системе после получения неудовлетворительного или нулевого результата по первому же запросу (иногда даже не

являющемуся правильным выражением поискового языка) или прекращение поиска после получения известных публикаций (а не *новой* информации).

7.1.2. Обобщенная схема воспроизводства информации

Схему взаимодействия потребителей-поставщиков информации (см. рис. 1.6) в общем случае можно преобразовать в схему воспроизводства информации, представленную на рис. 7.1.

Здесь информационные ресурсы, наряду с оригинальным авторским представлением материала, в большинстве своем характеризуются высокой систематизированностью (тематической профильностью источников и ядерностью тематических потоков), а также практически обязательным наличием метаинформации: поисковых образов документов и систем вторичной информации — рубрикаторов и тезаурусов, обеспечивающих единообразие представления и организации доступа к ресурсам.

Операционными объектами собственно машинного поиска являются поисковый образ документа (ПОД) и поисковый образ запроса (ПОЗ), соответствие которых устанавливается поисковым механизмом АИПС на формальном уровне. Установление же истинного соответствия содержания документа информационной потребности предполагает соотнесение на смысловом уровне: пользователь как бы реконструирует возможное содержание по основным понятиям, представленным в ПОД, и далее полученный образ соотносит с реальной потребностью. При этом адекватность образа действительному содержанию документа определяется не только качеством индексирования, но и уровнем знания субъектом средств отражения — концептуальной схемы предметной области и возможностей информационно-поискового языка.

Применительно к конкретному информационному сообщению можно сказать, что каждый элемент ИД осуществляет семантические или форматные (но не синтетические) преобразования этого сообщения, внося ту или иную неопределенность. Например, неопределенность, вносимая структурно-форматными преобразованиями (выделение формальных поисковых признаков, форма представления и распространения сообщения и т. д.), может приводить к *ненахождению*, а семантическими (адаптация содержания требованиям издания, например) — к *неузнаванию*.

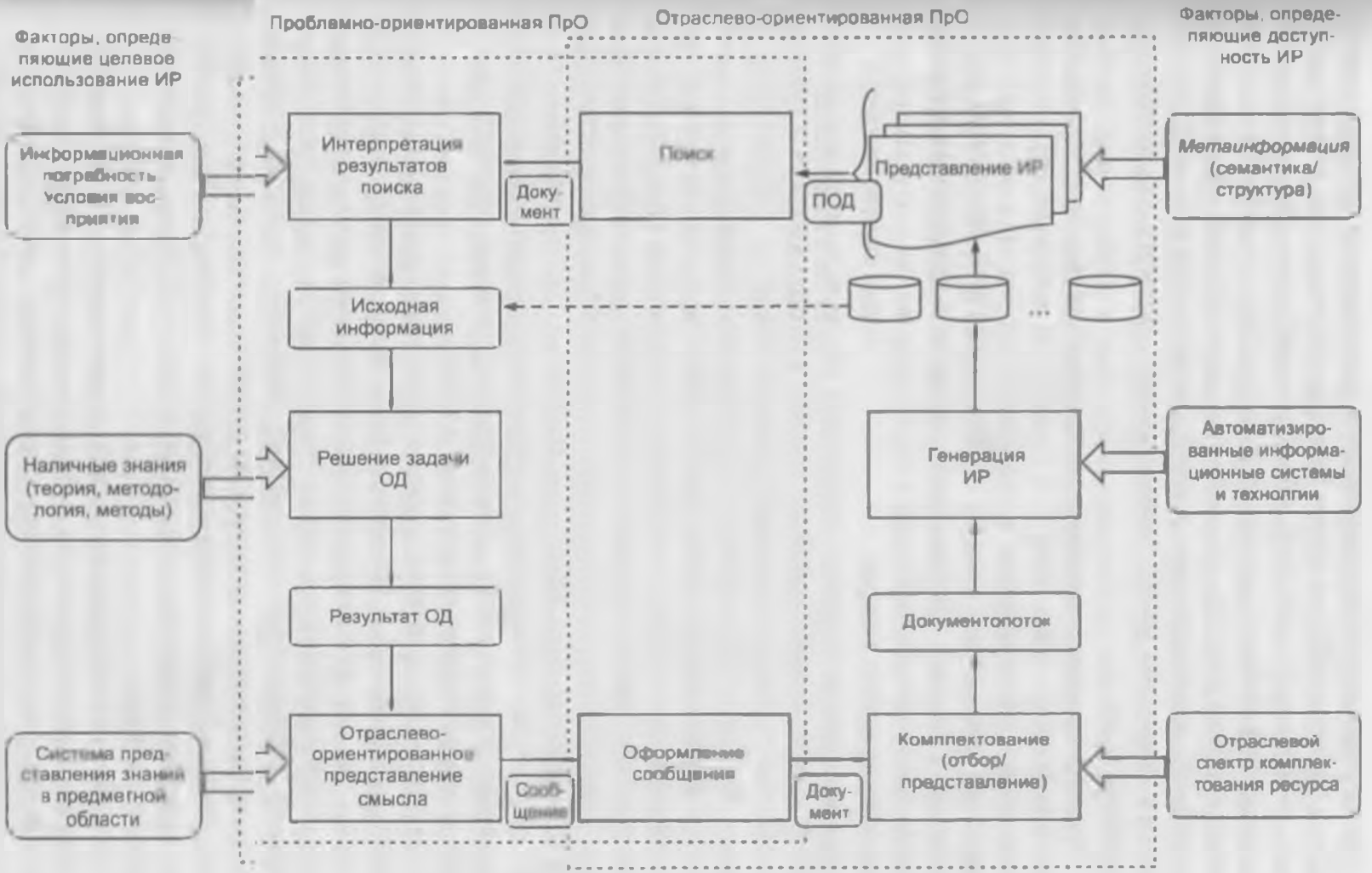


Рис. 7.1. Обобщенная схема воспроизводства информации

Вследствие этого информационно-поисковая деятельность должна представлять собой *не одноактную*, в общем случае, *итеративную* последовательность действий, обеспечивающих не только получение полезной информации, изменяющей состояние потребителя (в части решения задачи его ОД), но и данных, позволяющих объективно (и, желательно, количественно) оценить прагматические свойства найденной с помощью АИПС информации — полноту, достоверность, актуальность и т. д.

Это подчеркивает *активную* роль потребителя в процессе получения информации: связь типа 5—1 на рис. 1.6, выполняющая передачу информации, инициируется не системой, как это следовало бы из направления стрелки, а наоборот — *обращением* потребителя *посредством АИПС* к ресурсу, выбираемому им же, и именно он должен оценить результат взаимодействия и принять решение о его завершении или выборе другого ресурса.

Следует также отметить, что важной, но мало замечаемой особенностью является принципиально разное для пользователя и системы восприятие основных информационных объектов — документов и запросов. Человек рассматривает их как носители информации (смысл которой в общем случае может быть различным в зависимости от точки зрения конкретного пользователя), выделяя и преобразуя при этом отдельные фрагменты (часто не совпадающие с текстовым предложением, параграфом и т. п.) так, чтобы в сознании возникли устойчивые образы и понятия. Для АИПС те же объекты — это совокупности данных, из которых *механически* (не извлекая и не преобразуя смысла) выделены термины (слова, словосочетания, шифры, даты и т. д.), которые и сравниваются с терминами запроса.

Далее, человек считает документ полезным (наиболее соответствующим запросу), если тот несет новую, ранее не известную информацию, т. е. при решении практической задачи дает или позволяет найти ответ на некоторый *вопрос* («как?», «что?»). Система же считает наиболее соответствующим запросу документ, который содержит наибольшее количество терминов из запроса. То есть вполне вероятно, что пользователю в первую очередь будут выданы наиболее знакомые документы, возможно, написанные им же, что вряд ли принесет ему новое знание.

Еще один важный момент связан с понятием *структура документа*. Для человека это понятие в большинстве случаев (по крайней мере, для текстовых документов) ассоциируется с удобством восприятия, т. е. *описание структуры* практически не используется, поскольку

ку отдельные информационные поля документа *узнаются* обычно по косвенным признакам (угадываются). Для автоматизированных систем понятие структуры является неотъемлемым и изначально определяющим. Более того, для каждого структурного элемента (реквизита) документа обязательно определены свои формат, имя и, возможно, свой метод обработки. Например, способы записи дат или разные правила выделения терминов в разных текстовых полях (во многих системах знак пробела в поле ключевых слов не считается признаком разделения, позволяя таким образом выделять словосочетания).

7.1.3. Типология поисковых задач и форма выражения запроса

Типы поисковых задач

В зависимости от характера задачи основной деятельности пользователя по степени соотношения известного/неизвестного в предмете поиска можно выделить три типа поисковых задач.

К задачам первого типа (*атрибутивный поиск)* относится поиск объекта, когда известно, что этот объект существует (например, поиск фактографии или трудов конкретного автора). Знания пользователя об искомом объекте полные, цель поиска — найти его документальное представление. Модель такого «атрибутивного» поиска может быть представлена как логическое выражение над именами понятий, задаваемыми терминами или их комбинациями.

Второй тип задач (*тематический поиск*) — подбор информации по некоторой теме, например, для обзора научной проблемы, обоснования или поиска метода решения научной или практической задачи. Пользователь, уже обладая знаниями, определяет место задачи (как вновь вводимое понятие в системе уже известных понятий), ищет документы, содержащие материал, с необходимой полнотой раскрывающий новую для него тему или дающий возможность построения нового метода решения задачи. Поисковая модель в этом случае — это частично известные понятия, связи или комбинации. Тематический поиск реализуется как последовательность атрибутивных поисков, каждый из которых соответствует определенному (априорно заданному) аспекту представления объекта поиска.

Третий тип задач (*проблемный поиск)* — это поиск, который, по сути, является основной составляющей творческого процесса: определение путей решения профессиональной задачи пользователя.

Здесь изначально может отсутствовать четкость структуры знания: пользователь располагает отдельными фактами, возможно, не имеющими между собой доказанных связей. Проблемный поиск — это нахождение описаний объектов или их составляющих, актуально или потенциально существующих, и в совокупности, возможно, образующих целое, свойства которого, возможно, будут больше суммы свойств частей. То есть этим свойствам в явной форме могут не соответствовать «собственные» атрибуты, а новое свойство, например, может быть задано комбинацией уже известных атрибутов. В этом случае к неопределенности отображения объекта на предметную область ИС, свойственной тематическому поиску, добавляется неопределенность на уровне «субъект-объект ОД»: представление, которое субъект имеет об объекте поиска, может не соответствовать представлению конкретного источника. Логическая поисковая модель для этого случая — поиск похожих документов, содержание которых некоторым образом ассоциируется с задачей пользователя.

Формы информационных потребностей

Информационная потребность также имеет несколько форм [Mizzaro 1998; Tailor 1968], соответствующих разным стадиям процесса познания (когнитивным состояниям потребителя информации), для которых характерны разные формы проявления знания о незнании объекта поиска.

Реальная информационная потребность, отражая проблемную ситуацию пользователя в несистематизированной форме (она еще не вполне осознана), характерна для начальной стадии ОД.

В процессе понимания проблемной ситуации реальная ИП преобразуется в *осознанную ИП*, представленную в виде *вопроса* или *задачи* на привычном естественном или научном языке, и затем преобразуется в *поисковый запрос*. Для запроса характерно то, что вопросы типа «Как?» и «Почему?» должны быть преобразованы в вопрос типа «Есть ли?», поскольку именно такая форма представления потребности является наиболее адекватной теоретико-множественной модели поиска. Преобразование вопроса в запрос происходит в сознании человека и имеет качественный характер. Переход от реальной к осознанной ИП тем сложнее, чем менее определена задача ОД: для задач проблемного типа этот переход наиболее труден, так как пользователь не представляет, *какая именно* информация нужна для решения его задачи и не изменит ли она саму постановку задачи. Наиболее

адекватной формой представления осознанной ИП как поискового запроса может быть семантическая сеть — граф понятий, характерных для объекта поиска.

Поисковый образ запроса — это *выраженная ИП* — представляется лингвистическими средствами конкретной АИПС, причем лексический состав ПОЗ уже в значительной степени будет зависеть от особенностей выбранного информационного ресурса. Формирование ПОЗ, в отличие от других форм ИП, производится в среде АИПС, но эффективность процесса его составления определяется не только интерфейсными возможностями системы, но также и информационной грамотностью и профессионализмом пользователя.

Формы выражения запроса

Для человека идеальной коммуникативной формой представления знаний и потребностей является *вербальная* — в виде терминологического выражения. Принципиально важной особенностью вербального способа является изначальная контекстная определенность (хотя этот контекст, скорее всего, представлен только в сознании). То есть отдельное высказывание как грамматическая форма (предложение) в общем случае может порождать в сознании воспринимающего несколько смыслов, а исходный смысл высказывания будет воспринят только при условии полноты передачи исходного контекста.

Запрос с точки зрения способа его представления — это так же, как и в случае документа, терминологическое выражение, представляющее гипотетический объект через описание свойств (атрибутов, связей), наличие которых как признаков (зачастую уже безотносительно характера атрибутов и связей) должна проверить ИС в документах БД. То есть в итоге ПОЗ должен быть построен по типу вопроса «Есть ли?»

Вербальной форме запроса свойственно то, что она предполагает построение завершенного, логически и синтаксически правильного выражения. Такой подход «по духу» отражает стремление к точности выражения, исключающей возможность многозначности ответа, и, соответственно, в минимальной степени учитывает свойство комбинативности. В условиях, когда семантическая неопределенность отсутствует (поисковая задача первого типа), вербальная форма, безусловно, предпочтительнее, как предпочтительнее аналитический способ задания математической функции по сравнению, например, с табличным.

С другой стороны, содержание потребности частично или полностью может быть представлено уже существующими документами (как сообщениями, содержащими частные решения проблемной ситуации или имеющими с ними какую-либо смысловую общность). В этом случае можно говорить о *кластерной форме запроса*.

Эти формы являются альтернативными, но скорее взаимодополняющими, чем взаимно исключающими, что воплотилось в практике АИПС в виде двух уже привычных форм поискового запроса в диалоге «человек-система» — *запросно-ответной* и *гипертекстовой*.

С точки зрения свойства комбинативности и запрос, и документ являются моделями, представляющими средствами языка отдельные части и аспекты некоторого целостного фрагмента предметной области. Но при этом цель создания ПОД — представить изначально уникальный смысл документа компактной композицией признаков (например, в случае дескрипторных ИПЯ — ключевыми словами), по возможности не увеличивая комбинативность порождаемых ими возможных смыслов. Цель построения ПОЗ — сохраняя уникальность проблемной ситуации, увеличить комбинативность смыслов, порождаемых композицией поисковых признаков запроса, для того чтобы максимально охватить варианты представления объекта поиска.

Тот факт, что пользователь за новым знанием обращается в массив уже известного знания (хотя, возможно, и противоречивого[1]), предопределяет очевидность того, что запрос представляется в виде *гипотетического документа*, описывающего реальный, создаваемый или воображаемый объект. То есть в этом контексте задача поиска может быть сформулирована следующим образом: *найти уже существующие документы, которые являются содержательным аналогом запрашиваемого гипотетического*.

7.1.4. Обобщенная схема поиска

Резюмируя ранее приведенное, можно сказать, что эффективность информационного поиска определяется следующими факторами:

- свойством концентрации-рассеяния информации, предопределяющим априорную неполноту любого отдельного ИР — источ-

[1] Массив системы включает и документы, содержащие сведения, которые могут быть неполными, непроверенными, взаимно противоречивыми.

ника информации практически по любой теме. Любой ресурс всегда ориентирован не только тематически (по отраслям знаний) и на определенный вид информации (НТД, патенты, отчеты НИР и т. д.), но также имеет свои системы представления и средства доступа к информации;

- свойством эмерджентности информации, предполагающим множественность и комбинативность использования любого информационного сообщения;
- свойствами информационно-поисковой деятельности, зависящей как от характера задач ОД, так и от особенностей человека — его возможностей по восприятию и интерпретации найденных документов (информация может быть потенциально полезной, но актуально не воспринятой, например, по причине недостаточности знаний получившего ее потребителя);
- свойствами вычислительной среды реализации АИПС, для которой характерна жесткость процедур отбора и предопределенная ограниченность форм представления информации.

В целом процесс информационного поиска может быть представлен как итеративная цепочка операций, выполняемых в совокупной человеко-машинной среде (в сознании человека и в вычислительной машине), последовательно снимающей неопределенности, обусловленные перечисленными ранее свойствами информации, и в итоге реализующей своеобразное преобразование информационной потребности в совокупность документов, содержание которых удовлетворяет эту потребность, т. е. информация найденных документов обеспечивает решение задачи ОД.

Укрупненный алгоритм итеративного процесса поиска информации представлен на рис. 7.2.

Такой подход позволяет рассматривать процесс поиска как *последовательное*[1] изменение состояний (этапов) взаимодействующих подсистем (человека и автоматизированной информационно-поисковой системы), направленное на локализацию (снятие) неопределенностей следующих видов:

1) неопределенности соотношения «известного/неизвестного» в предмете поиска (свойственна реальной ИП);

2) неопределенности системы характеристических признаков для структуризации предмета поиска (свойственна осознанной ИП);

[1] То есть свести к последовательности однопараметрических задач.

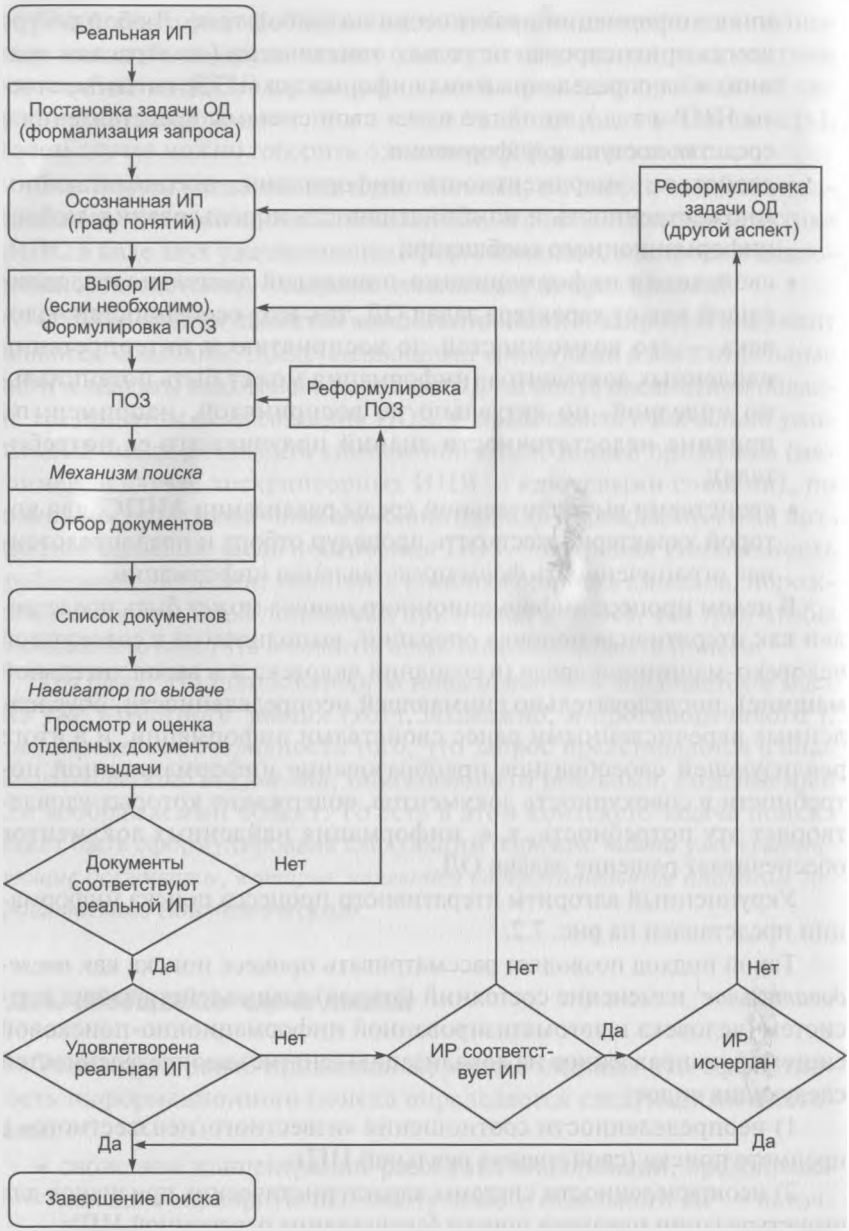


Рис. 7.2. Обобщенный алгоритм автоматизированного информационного поиска

3) лексической неопределенности, как фактора степени соответствия информационно-поискового языка естественнонаучному языку предметной области (свойственна выраженной ИП);

4) неопределенности критериев сравнения поисковых образов (адекватность формальных мер близости, реализованных в конкретных АИПС);

5) неопределенности интерпретации ПОДов (субъективность и неполнота реконструирования пользователем смысла найденных документов);

6) неопределенности тематического соответствия и степени полноты представления проблематики в данном ИР.

Причем первые четыре вида неопределенности имеют информационную природу (преобразование форм представления информации), пятая характеризует поисковый аппарат АИПС, а шестая отражает когнитивные особенности человека — приемника и генератора информации. Последняя существенна в том случае, когда в используемом ресурсе не была найдена информация, полностью обеспечивающая решение задачи ОД, и потребителю необходимо принимать одно из следующих решений:

1) продолжить поиск в этом ресурсе в надежде на то, что удастся так реформулировать запрос, что он выведет на нужный «пласт» информации;

2) перейти в другой ресурс (например, когда любая формулировка запроса дает отрицательный результат);

3) прекратить поиск и заняться непосредственно решением задачи (когда становится понятно, что легче открыть заново, чем найти описание открытия).

Не являясь практически измеримыми величинами, эти параметры, тем не менее, позволяют обозначить характер изменения состояния сторон.

Поскольку разные типы поисковых задач имеют разные типы и степени неопределенности, достаточно очевидно, что «траектория» поиска (циклы и число итераций) для каждого случая будет различной. И какой бы интеллектуальной система ни была, роль человека будет определяющей. То есть в целом ситуация все еще отвечает тезису, что «...до тех пор, пока люди не научатся адекватно выражать на естественном языке свои информационные потребности, документальная ИПС должна быть организована таким образом, чтобы *человек* мог как бы *исследовать* поисковый массив, изменяя формулировку поискового предписания в зависимости от промежуточных результатов поиска» [Михайлов, 1968].

7.2. Организация данных и процесс поиска

7.2.1. Критерии отбора документов

Как отмечалось ранее, процесс поиска всегда, так или иначе, сводится к процессу простого перебора — упорядоченного или случайного, полного или частичного. При этом достаточно очевидно, что степень соответствия («нужность») устанавливается не путем эмпирического «натурного» подбора — помещения очередного выбранного объекта непосредственно в конструкцию создаваемой системы, а путем сопоставления *образов* соотносимых объектов — параметров, свойств и т. д.

Система производит отбор ПОДов автоматически: механизм поиска включает в выдачу те документы, ПОЗы которых удовлетворяют формальному *критерию отбора*. Очевидно, что формальный (количественный) критерий соответствия может быть определен только в том случае, если соотносимые объекты имеют одинаковую природу (или приводятся к таковым) и, соответственно, сравниваемые атрибуты принадлежат одному пространству. То есть поскольку нельзя непосредственно сравнивать информационный образ с реальным объектом, для корректного соотнесения необходимо либо создать описание реального объекта, либо по образу (описанию) построить объект или его «действующий» макет — эквивалент объекта, удовлетворяющий требованиям решаемой задачи.

В отличие от логики сознания человека, где вопросы имеют форму «как», «почему», предполагающую развернутую форму ответа, машинная логика в основе своей может обрабатывать только вопросы типа «есть ли».

Для документальных систем элементом данных, задающим смысл, является термин языка (слово, словосочетание), а запрос сводится ко 2-му типу[1]: $A(?) = V$ — «Какие объекты имеют значение атрибута, равное V». А наиболее естественной формой критерия отбора будет являться предикат, построенный для выражения ПОЗ. Соответственно, документ считается *формально релевантным* (и, соответственно, включается в выдачу), если для данного документа предикат принимает значение «истина».

Простейшим вариантом критерия отбора является правило: документ считается *формально релевантным*, если количество общих слов,

[1] Типология простых запросов приведена в гл. 1.

которые он имеет с запросом, не менее заданного порогового значения. Этот критерий может быть усилен путем взвешивания, в частности, ранжирования терминов по их важности. Сущность метода заключается в назначении пользователем терминам запроса весовых коэффициентов. При поиске документ будет релевантным, если сумма весовых коэффициентов совпавших терминов будет больше заранее установленной величины.

Оценка семантической близости функцией «косинус» является своеобразной нормированной суммой «весовых» коэффициентов совпадающих терминов. Здесь запрос и документ представляются n-мерными векторами в пространстве терминов, где их i-е координаты принимают значения *«единица»* или *«ноль»* в зависимости от того, входит ли i-й термин в соответствующий поисковый образ. Документ считается *формально релевантным*, если мера принимает значение не меньше *порогового*.

В большинстве реальных ИПС критерий задается выражением алгебры логики, сформулированным пользователем на основании семантической структуры информационной потребности.

7.2.2. Организация поисковых массивов

Неявным, но основным с точки зрения реализации алгоритма поиска фактором является порядок выборки. Выборка может проводиться в «естественном» порядке, соответствующем расположению объектов в массиве, или в «искусственном», соответствующем, например, некоторой классификации предметной области.

И в том, и в другом случае мы имеем дело с *перебором* объектов, выбираемых для сравнения из хранилища. То есть рациональность построения процедуры поиска зависит от длины перебора, что в свою очередь определяется как характеристиками хранимых объектов (в первую очередь, размерами документов), так и характером запросов (в нашем примере — поиском по предмету или поиском по шифру хранения документов). Таким образом, оптимизация достигается сокращением перебора — длины *последовательно* проверяемого массива.

Есть две классические технологии обработки запросов (так называемых *режимов информационного поиска*): режим *ретроспективного поиска* и режим *избирательного распределения информации*.

При ретроспективном поиске очередной ПОЗ сравнивается со всеми ПОД. В режиме избирательного распределения информации

схема зеркально симметричная: ПОД каждого документа сравнивается со всеми поисковыми образами запросов. То есть в первом случае запросы обрабатываются после создания массива ПОД, которые, накапливаясь, формируют *ретроспективную* БД, а во втором — массив ПОЗ создается до обработки документов (при этом хранится массив ПОЗ и не обязательно — ПОД). Поэтому эти режимы иногда называют режимами обработки *разовых* и *постоянно действующих* запросов.

Технологии (алгоритмы) поиска основываются на двух типах *организации массива* объектов поиска — *прямой* и *инвертированной*. Для рассмотрения взаимосвязи алгоритма поиска и организации массива здесь и далее используем знакомый всем пример организации и поиска информации в традиционных библиотеках[1].

В случае прямой организации массива (хранилища) документы могут размещаться в последовательности, никак не связываемой с порядком какой-либо классификации или алфавита, в простейшем случае — в порядке их поступления. Но с точки зрения основного назначения АИПС — информационного обеспечения ОД, определяющим в понятии «*прямая организация*» является не характер размещения документов — единиц хранения, а размещение *содержания документов*, которое представлено изначальной «естественной» *последовательностью слов*, образующих, в том числе, и контекст их употребления. При прямой организации поиск в больших массивах будет требовать достаточно много времени, так как для сравнения с запросом надо последовательно выбирать *все*[2] документы из хранилища по той простой причине, что не обратившись к документу, мы не можем судить о его содержании.

В инвертированном массиве документы могут быть, например, разбиты на подмножества, которые, в свою очередь, упорядочены в соответствии с некоторой классификацией и, что особенно важно, обозначены идентификаторами, отражающими предметное содержание соответствующего класса (в пределе таким идентификатором мо-

---

[1] Отметим, что выбор этого примера основывается не только на его «привычности» для человека, но и на том, что с методологической и системной точек зрения применяемые в библиотеках подходы, методы и технологии являются по существу универсальными и не зависящими от степени автоматизации.

[2] Конечно, перебор можно завершить, когда будет получен документ, отвечающий в рамках уже известного подхода, однако вполне возможно предположить, что при этом мы не дойдем до документов, опубликовавших новейшие достижения, опровергающие традиционный подход.

жет быть отдельный термин). Такое упорядочение документов в хранилище сопровождается построением вспомогательной структуры — *инвертированного справочника*, в котором с каждым *индексом* (идентификатором класса) связан список ссылок на документы, отнесенные к этому классу[1]. Для случая текстовых баз данных в качестве индекса может выступать термин из текста или термин из предопределенного словаря, выступающий в качестве смыслового эквивалента фрагмента текста, словосочетания или отдельного слова текста.

В контексте приведенной в гл. 1 типологии простых запросов отметим, что запросы типа 1 выполняются поиском по «прямому» массиву: доступ к записи производится по первичному ключу. Запросы типа 2 выполняются поиском по инвертированному списку: доступ к записи производится по указателю, выбираемому из списка по значению вторичного ключа. Ответом в этих случаях будет *значение* атрибута или идентификатора. Запросы типа 3 имеют ответ — *имя* атрибута.

Запросы типа 2, 5, 6 относятся к нескольким атрибутам, и в этом случае могут быть построены несколько индексов, облегчающих поиск по этим ключам.

Для документальных систем, которым свойственны в основном запросы 2-го типа, можно выделить три следующих типа архитектур доступа.

1. *Системы с вторичными индексами.* В этих системах последовательность расположения записей соответствует последовательности значений первичного ключа. Как правило, используются один первичный индекс и несколько вторичных.

2. *Системы частично инвертированных файлов.* В этих системах записи могут располагаться в произвольной последовательности. В отличие от систем первого типа первичный индекс отсутствует. Вторичные индексы применяются для прямой адресации записей, что существенно облегчает включение в файл новых записей, так как допускается их размещение в любом свободном участке файла.

3. *Системы полностью инвертированных файлов.* В этих системах предусмотрено наличие файлов, содержащих значения отдельных элементов данных, входящих в состав записей, — допускается раздельное хранение элементов данных записи. Значения элементов

[1] Систематический каталог библиотеки имеет типично инвертированную организацию: карточки с шифром раздела представляют собой индекс, а инвертированный список — это распложенные за ней карточки с шифрами хранения соответствующих единиц.

данных, составляющих конкретную запись или кортеж, в общем случае могут размещаться в памяти произвольно. Для ускорения процесса поиска в системе используют два набора индексов: *индекс экземпляров* (значений ключей) и *индекс данных* (инвертированный список). С помощью индекса экземпляров в файле можно найти элементы данных, имеющих заданное значение. С помощью индекса данных можно найти записи, связанные с заданными значениями элементов. Такая организация характерна для организации данных *документальных информационных систем*.

Пример организации данных документальной АИПС

Примерная схема организации данных для представления и поиска информации одной из первых промышленных систем поиска документов STAIRS (Storage and Information Retrieval System), разработанной фирмой IBM в 1970-х годах, приведена на рис. 7.3. Отметим, что такая структура не только хорошо иллюстрирует принципы организации данных в документальных системах, но и составляет основу большинства современных АИПС.

Физическая структура БД рассматриваемой системы включает в себя следующие четыре файла:

- файл частотного словаря, устанавливающий соответствие между словом, встречающимся в БД, его кодом и частотой, используется при текстовом поиске;
- инверсный (инвертированный, обратный) список, содержащий для каждого слова БД список документов, его содержащих, используется при текстовом поиске;
- текстовый файл, содержащий собственно документы, используется при выдаче (просмотре) документов;
- прямой, последовательный файл, содержащий «собранные» в одну строку фиксированной длины форматные поля и список кодов слов, находящихся в тексте данного документа. При необходимости в соответствующих местах находятся разделители сегментов и/или предложений. Файл используется при наличии в запросах конструкций *SENT, SEGM, CTX,* определяющих необходимость проверки взаимного расположения поисковых терминов.

На рис. 7.4 представлена структура индексных файлов *словарь слов,* в котором содержится перечень терминов, встречающихся в до-

Рис. 7.3. Организация данных в документальной АИПС STAIRS

кументах. Словарь содержит само слово, его характеристики и указатель на списки ссылок на документы, в которых встречается это слово.

Ввиду значительных размеров словаря его организация предусматривает наличие специального индекса, представленного *матрицей пар знаков*. Каждой паре знаков поставлен в соответствие указатель на *блок словаря*, содержащий группу слов, начинающихся с этих знаков. Знаками могут быть буквы, цифры, а также специальные символы. Группы слов в словаре имеют переменную длину.

Некоторые слова в словаре могут иметь одинаковый смысл; такие слова связаны с помощью специального указателя «синоним» (на рисунке связи данного типа показаны штриховыми стрелками).

При использовании инвертированных форм представления информации на первом шаге проводится поиск в инвертированном справочнике, и если предмет запроса отождествлен с соответствующим классом, то на втором шаге для детального соотнесения содержания документа и запроса обращение будет производиться только к сравнительно небольшому числу документов — только к тем, которые отнесены к этому классу. Таким образом, за счет введения информационно-избыточной структуры и дополнительного шага поиска достигается существенный выигрыш во времени: суммарное время на поиск в инвертированном справочнике существенно меньше поиска в целом массиве документов, поскольку длина индекса, идентифицирующего содержание документа, обычно на порядки меньше длины

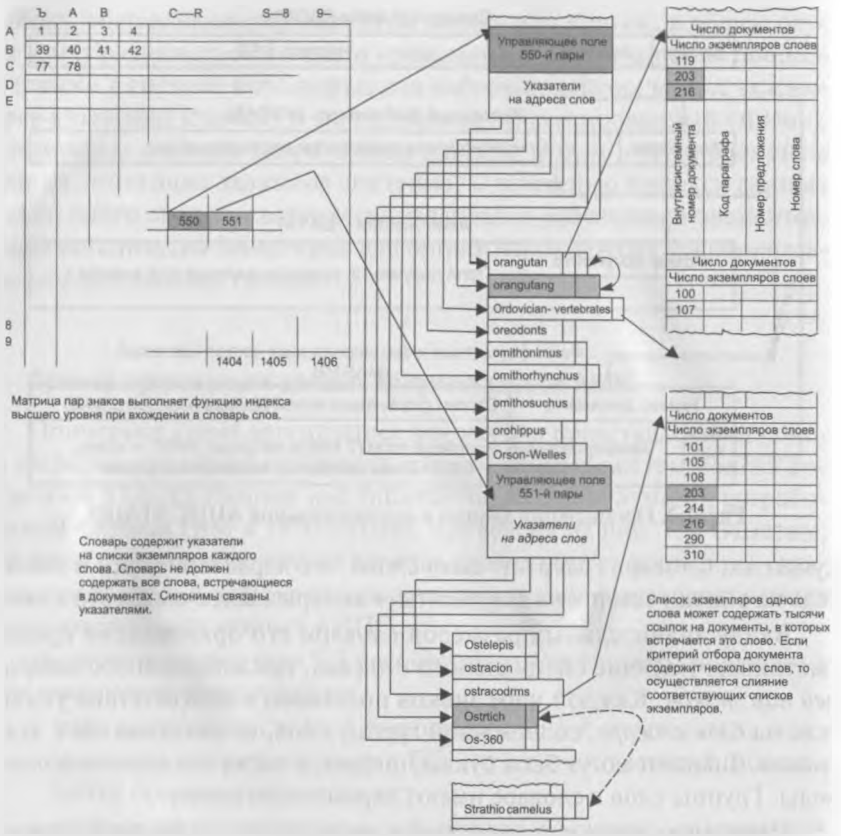


Рис. 7.4. Организация индексных файлов документов АИПС STAIRS

самого документа, и, кроме того, индексы строго упорядочены (например, по лексикографическому признаку), что позволит, например, использовать дихотомический поиск.

7.2.3. Идентификация содержания документальной информации

Идентификация содержания с помощью индексов строится по принципам искусственных информационно-поисковых языков: каждый индекс представляет то или иное множество характеристических признаков. Это позволяет сократить число просматриваемых документов: в соответствии с формулой композиции признаков (что хоро-

шо реализуется выражением алгебры логики) производится слияние относящихся к разным индексам списков ссылок на документы, т. е. выбираются только те документы, которые описываются именно этим сочетанием.

Использование технологии индексирования, тем не менее, имеет принципиальные недостатки:

1) индексационная информация, относящаяся к документу, статична: индексы, приписанные к документу, будут всегда иметь смысл, заложенный при создании языка индексирования (например, классификации конкретного поколения);

2) нельзя без дополнительных затрат реализовать управление глубиной поиска или выполнить поиск с использованием критерия «частичного» соответствия.

И все же, автоматизация поиска информации основывается именно на технологии индексирования (как способа идентификации содержания и, соответственно, инвертированных форм представления информации) документов, поскольку документальные АИПС имеют следующие принципиально важные особенности построения и использования [Солтон, 1979].

Во-первых, задачи в области документального поиска несравнимы с другими задачами обработки текстов, такими, как, например, автоматический перевод. По существу документальные ИПС создаются только для того, чтобы указать потребителю те документы, которые, *скорее всего*, имеют отношение к данному интересующему его вопросу. Поэтому здесь можно ограничиваться довольно грубым раскрытием содержания документа, указывающим лишь основные моменты, вместо фразеологического анализа, необходимого, например, при переводе.

Во-вторых, поисковые системы создаются для обслуживания больших и часто разнородных групп потребителей. Поскольку последние могут иметь различные потребности и цели, поисковые запросы варьируются от вопросов обзорного или познавательного характера до очень подробных аналитических запросов. При таких условиях слишком подробный анализ может оказаться излишне (или даже неприемлемо) специализированным для большинства пользователей.

В-третьих, в основе процесса оценки лежит некоторый критерий эффективности, обычно усредняемый по многим поисковым запросам. Это означает, что более предпочтительными оказываются такие методы, которые дают умеренно высокую общую эффективность, чем, может быть, более тонкие алгоритмы, которые могут превосход-

но обрабатывать одни запросы, но значительно хуже другие. Практически может оказаться, что для каждого вида запроса оптимальным будет некоторый специфический метод, но для среднего запроса наилучшими являются более простые методы индексирования.

Из изложенного следует, что качество поиска, в первую очередь полнота и точность могут быть достигнуты только за счет метаинформационной и/или процедурной избыточности.

Увеличение полноты достигается, например, следующими путями:

1) увеличением полноты индексирования документа (вплоть до индексирования несколькими методами каждого поля документа, включая полный текст);

2) расширением запроса за счет близкой по смыслу лексики, выбираемой пользователем или системой из дополнительных (метаинформационных) справочников, таких, как словари синонимов, тезаурусы;

3) использованием многостадийных итеративных процедур и/или нескольких механизмов поиска;

4) снижением точности запроса или порога выдачи, что позволяет, занижая требования к степени смыслового соответствия, увеличить вероятность попадания в выдачу истинно релевантных документов. Очевидно, что при этом в выдачу попадет во много раз больше нерелевантных документов, и пользователь должен будет потратить больше времени на отбор истинно релевантных. Этот путь кажется малопривлекательным, однако поскольку многие (если не большинство) промышленные ИР достаточно ограниченно используют средства, перечисленные в первых трех пунктах, в некоторых случаях этот вариант является единственно возможным для получения удовлетворительного результата.

Увеличение точности достигается, например, следующими путями:

1) использованием для индексирования словосочетаний, обычно дескрипторов ИПТ или словосочетаний, приведенных к нормализованной форме;

2) использованием при построении поискового образа документа и/или запроса статистики словоупотреблений и/или лингвистических процессоров, что позволяет «взвешивать» термины;

3) использованием сложных критериев отбора, дифференциально учитывающих роль и значимость терминов и терминологических конструкций;

4) использованием постобработки, упорядочивающей документы по релевантности, что позволяет сократить время пользователя при просмотре.

Перечисленные средства не являются взаимоисключающими. Например, во многих поисковых машинах и системах (прежде всего в тех, поисковые интерфейсы которых в явной форме не ориентированы на алгебру логики) после отбора проводится постобработка, так или иначе упорядочивающая отобранные документы. В простейшем случае это может быть сортировка по какому-либо существенному для пользователя атрибуту (дате публикации, шифру классификации и т. д.) или степени соответствия (мере близости). В более сложных случаях система может рубрицировать выданные документы в соответствии с какой-то классификационной схемой или кластеризовать их по степени взаимной близости. Существо постобработки с точки зрения сокращения объема документов, которые должен просматривать пользователь до момента фактического удовлетворения ИП, состоит в том, что достаточно большое множество всех найденных документов разделяется на сравнительно небольшие подклассы. И, обычно, пользователь удовлетворится просмотром документов одного, двух подклассов, вполне обоснованно не обращаясь к остальным.

7.3. Функциональная обработка запросов и документов в АИПС

7.3.1. Обобщенная схема обработки запросов и документов

С организационно-функциональной точки зрения в АИПС выделяются два контура: *обработки запросов* и *обработки документов*. В свою очередь, в контуре обработки документов могут выделяться (как отдельные подсистемы) контур *первичной* и *вторичной* информации. Контур первичной информации выделяется в отдельную подсистему в том случае, если массив первичных документов размещается на иных типах носителей или использует отдельную систему управления данными. На рис. 7.5 представлена обобщенная схема обработки запросов и документов в АИПС.

С точки зрения функциональности в составе АИПС можно выделить пять блоков:

- *блок предобработки* — преобразование в машинную форму документов и запросов;
- *блок формирования базы данных АИПС* — загрузка ПОДов и машинных форм документов (полных текстов) в базу данных;

- *блок поиска* — отбор по поисковому образу запроса из множества ПОД тех, которые удовлетворяют требованиям критерия смыслового соответствия;
- *блок постобработки* — упорядочение найденных документов;
- *блок выдачи* — форматирование и отображение материала найденных документов.

Как видно на рис. 7.5, изначальными являются процессы генерации информации и появление информационной потребности. Их возникновение происходит в сознании человека, однако выражение, так или иначе, связывается с конкретной предметной областью, ее структурой и терминологией. При этом в качестве средств представления могут использоваться такие лингвистические средства, как тезаурусы предметных областей, язык представления онтологий (OWL) или язык представления знаний (KWL). Для машинной формы материалов, ориентированной на передачу, используются коммуникативные форматы, как, например, ISO-2709 и ISO-8211, или XML, а для описания логической структуры ресурса, содержащего материалы, может использоваться язык описания ресурсов RDF.

Обработка поступающих в систему документов обычно включает:

- присвоение документу уникального идентификатора, необходимого для поиска, а также, возможно, для связывания ПОДа с полным текстом документа, для чего может использоваться соответствующий кодификатор или, например, система идентификации цифровых объектов (Digital Object Identifier — DOI);
- преобразование во внутрисистемный формат, когда могут использоваться XML-схемы и язык определения документов DTD;
- индексирование и, возможно, реферирование — построение поискового образа (не обязательно автоматическое или автоматизированное) в рамках лингвистических средств АИПС, для чего используются словари, рубрикаторы, классификации, тезаурусы предметных областей;
- загрузку ПОДов и, если в АИПС есть контур первичной информации, полного текста документа в базу данных. При этом используются языки определения и манипулирования данными соответствующей СУБД, а для оперативного взаимодействия с внешними ресурсами, например, XML-SQL.

При обработке запросов введенная пользователем формулировка приводится в соответствие с требованиями информационно-поискового языка (индексируется) и преобразуется во внутрисистемный

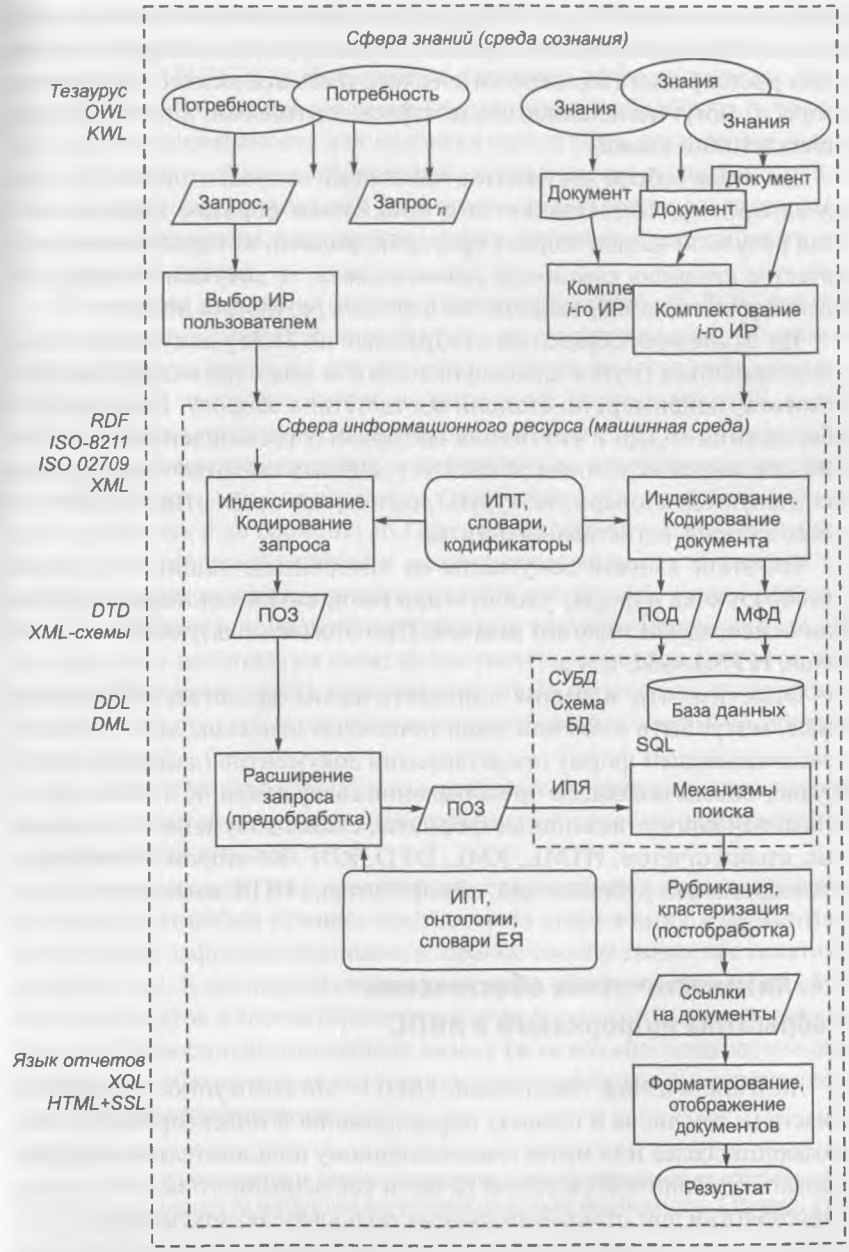


Рис. 7.5. Обобщенная схема обработки запросов и документов в АИПС

формат в соответствии с правилами информационно-поискового языка конкретной АИПС. При этом используются словари системы, а для расширения (терминологического и тематического обогащения запроса) могут использоваться тезаурусы, онтологии, а также словари естественного языка.

На этапе отбора документов поисковый запрос, по тому или иному алгоритму, сопоставляется с поисковым образом документа, и если результат удовлетворяет критерию выдачи, который выступает в качестве *критерия смыслового соответствия*, то документ (точнее, его идентификатор в БД) включается в список результата поиска.

На этапе постобработки отобранные по ПОЗу документы могут группироваться (путем классификации или кластеризации) и ранжироваться, например, по степени соответствия запросу. При этом для обогащения ПОДа и уточнения возможных (осмысленных) сочетаний лексических единиц за счет устойчивых семантических связей используются словари, тезаурусы, онтологии предметной области, а также словари естественного языка.

На этапе выдачи документы из внутренней машинной формы преобразуются в форму удобную для восприятия человеком и, более или менее, адекватную его задачам. При этом используются языки отчетов, HTML+SSL и т. д.

Отметим, что в целом лингвистические средства, упомянутые выше, могу. быть с той или иной точностью отнесены либо к группе, обеспечивающей форму представления документов (данных), либо к группе, обеспечивающей представление содержания. К первой группе относятся коммуникативные форматы, схемы документов и баз данных, языки отчетов, HTML, XML, DTD, RDF. Ко второй — тезаурусы, классификации, рубрикаторы, кодификаторы, ИПЯ, языки онтологий.

7.4. Лингвистическое обеспечение и обработка информации в АИПС

Лингвистическое обеспечение (ЛО) — это совокупность языковых средств (в том числе и правила реферирования и индексирования), позволяющих более или менее подготовленному пользователю взаимодействовать с машинной системой (в части согласованного использования терминологии при отыскании нужных пользователю документов).

Как уже отмечалось, принципиальное различие сред обработки информации (сознание человека — память ЭВМ) предопределяет, что

общий для них язык общения будет далек от естественного, и его конкретные возможности будут определяться, прежде всего, спектром задач АИПС и требованиями к уровню эффективности. Язык должен быть *безусловно приемлем* для машинной обработки, в то время как его приемлемость для человека может быть достаточно условной. Он может обладать большими возможностями, но быть сложным в освоении для человека (что ограничивает круг пользователей), но с другой стороны, упрощение[1] языка очевидно будет снижать эффективность отбора.

Если язык запросов (утилитарная поисковая часть ЛО), так или иначе, известен всем, кто проводит поиск, то другая часть, обеспечивающая индексирование документов, не столь заметна пользователю (уже потому, что пользователь не выполняет индексирования, а синтаксис языка запросов не используется в ПОДе). ЛО имеет еще одно, «системное» предназначение — согласование точек зрения на предметную область как для различных поставщиков и потребителей информации, так и во времени: ЛО имеет искусственную природу и создается в конкретных условия и в конкретное время (т. е. каждый его элемент имеет контекст, используемый в это время). Таким образом, если со временем контекст употребления термина изменился, то для адекватного восприятия смысла документа, содержащего этот термин, необходимо иметь возможность явно обратиться к «старому» контексту, который, следовательно, необходимо каким-то образом фиксировать.

В основу методов представления смысла положена та или иная знаковая система, но при этом различают классификационный и описательный подходы.

Классификация, как средство описания содержания документа, представляет собой процесс соотнесения содержания документов с понятиями, зафиксированными в заранее составленных систематических схемах. Классификационные методы обеспечивают систематизацию объектов в соответствии с некоторой заданной схемой классов, и код, присвоенный отдельному классу (или его мнемоническое обозначение), обеспечивает его полную идентификацию в рамках конкретного классификатора.

---

[1] Удобным упрощением для пользователя является применение естественного языка, для которого разработаны достаточно хорошие анализаторы. Однако, в любом случае, это не будет идеальным решением, так как программа использует априорные сведения о предметной области и не может учитывать ситуативность пользовательской задачи.

Описательные методы идентификации используются, как правило, в тех случаях, когда необходимо идентифицировать конкретный объект или группу объектов путем описания произвольного набора его характеристик. Описательный метод предполагает наряду с указанием классификационных характеристик выделение дополнительных наборов свойств, углубляющих характеристику объекта и сужающих область поиска.

7.4.1. Классификационные языки

В основе любой классификации лежит принцип деления. Каждый объект (материальный или нематериальный) с точки зрения решаемых классификацией задач характеризуется фиксированным множеством свойств, совокупность значений которых может говорить об эквивалентности (или близости) данного объекта некоторому множеству объектов, обладающих определенной общностью.

Развитие науки, как известно, характеризуется наличием двух противоположных тенденций: во-первых, дифференциацией, в результате которой каждая наука разделяется на все новые и новые ветви; во-вторых, взаимопроникновением не только смежных, но иногда очень далеких одна от другой наук, в результате чего появляются новые, ранее не существовавшие науки. Отсюда следует, что любая претендующая на научность и перспективность классификация должна учитывать особенности развития науки и иметь такую схему, которая бы позволяла адекватно отражать в классификации новые ветви уже сложившихся наук, новые науки и возникающие в результате дифференциации последних ветви новых наук.

Классификационные языки обычно строятся на базе классификации наук с ее делением на отдельные отрасли, хотя имеется множество объектов, особенно в области естествознания, медицины и техники, изучение которых не является задачей какой-либо одной науки (например, одна и та же машина или аппарат может применяться в различных отраслях техники).

Основными достоинствами классификаций являются следующие:

- весь поток научной информации индексируется в соответствии с классификациями;
- классификации отражают практически все направления в науке и технике, систематизируя объекты по основным существенным признакам;

- использование единой классификации не только облегчает поиск, но и обеспечивает платформу для единого понимания предмета рассмотрения.

Рассматривая классификацию как систематическое распределение объектов множества по классам, возникающее в результате последовательного многоступенчатого деления, можно выделить следующие два вида классификаций:

- *естественные классификации* — классификации, в основание которых кладутся существенные для выделяемых классов признаки;
- *вспомогательные классификации* — классификации, в основание которых кладутся несущественные для выделяемых классов признаки.

С точки зрения построения классификации разделяются на иерархические и фасетные.

При построении *иерархической классификации* деление на каждом уровне иерархии должно проводиться только по одному основанию. Но следствие этих правил — невозможность проведения группировки документов и информационного поиска по любому сочетанию характеристик, так как для построения иерархической классификации используется определенный ряд атрибутов (оснований деления), применяющихся только в одной последовательности. Более того, в иерархической классификации отдельные науки «разорваны» разветвлениями жесткого классификационного дерева, что предопределяет периодический пересмотр классификации. Но на разработку новых таблиц классификаций уходит много времени и труда, и новый вариант устаревает раньше, чем удается завершить работу по реклассифицированию документов. Наиболее известными иерархическими классификациями на сегодняшний день являются Десятичная классификация Дьюи, Библиографическая классификация Блисса, Классификация Библиотеки Конгресса США, библиотечно-библиографическая классификация ББК.

Фасетная система классификации в отличие от иерархической позволяет выбирать признаки классификации независимо как друг от друга, так и от семантического содержания классифицируемого объекта. Признаки классификации называются *фасетами* (facet — рамка). При создании фасетной классификации деление на классы проводится на основе всех возможных комбинаций признаков. Каждый фасет содержит совокупность значений соответствующего классификационного признака. Недостатком фасетной системы классификации является сложность ее построения, так как необходимо учитывать все многообразие классификационных признаков.

В практических задачах информационного обслуживания наиболее широко и устойчиво используются следующие системы классификации:

- *библиотечно-библиографические,* специально предназначенные для систематизации книг и других документов;
- *патентные*, служащие для индексирования объектов промышленной собственности, заявленных или признанных изобретениями;
- *классификации наук,* призванные систематизировать научную информацию (Государственный рубрикатор НТИ и отраслевые рубрикаторы, построенные на его основе, используемые при формировании всех видов информационных изданий).

Библиотечно-библиографическая классификация (ББК) предназначена для организации библиотечных фондов, систематических каталогов и картотек.

ББК имеет иерархическую структуру, позволяющую отражать содержание произведений печати, и включает следующие аспекты обобщения:

- основные таблицы;
- система типовых делений.

Основные таблицы представляют классы основных наук. Следующие (второй, третий, четвертый и т. д.) уровни классификации образуются путем деления первого уровня на подчиненные группы наук, отрасли деятельности, отдельные науки, проблемы, темы и т. п.

Система типовых делений помогает выделить и единообразно разместить однотипную литературу в систематическом каталоге. Система типовых делений представлена таблицами основных типовых делений, используемых во всех отделах классификации, и таблицами специальных типовых делений, обслуживающих отдельные отрасли наук.

К таблицам основных типовых делений относятся:

- таблицы общих типовых делений;
- таблицы территориальных типовых делений;
- таблицы типовых делений социальных систем.

Общие типовые деления используются для дополнительного тематического (история науки, научные и культурные связи и т. п.) и формального (библиографические пособия, справочные издания, сборники и т. п.) деления.

В ББК принята логически последовательная система кодирования, непосредственно связанная со структурой классификации. Система кодирования позволяет детализировать общее понятие путем

присоединения к имеющимся индексам новых знаков справа и, наоборот, при необходимости сокращать детализацию путем отбрасывания от индекса его конечных знаков.

В алфавит системы кодирования входят:

- арабские цифры;
- строчные и прописные буквы русского алфавита.
- точка («.»), двоеточие («:»), дефис («-»), круглые скобки («( )»), косая черта («/»).

Универсальная десятичная классификация (УДК) создавалась в основном еще до того, как был разработан фасетный принцип. Поэтому в УДК этот принцип получил лишь частичное воплощение, и она, по существу, является классификацией *полуфасетного* типа.

Все классы УДК сгруппированы в шесть фасетов (Общий предмет, Место, Народность, Время, Язык документа, Форма документа), каждый из которых подразделяется по иерархическому принципу на несколько уровней. Для идентификации отдельного класса на каждом уровне используются десятичные цифры, тем самым в индексах УДК каждая последующая цифра не меняет значения предыдущих, а лишь уточняет их, обозначая частное понятие. Индекс УДК представляет собой последовательность десятичных цифр, возможно, разделенных на группы знаком «точка».

Фасет «Общий предмет» (основная таблица классификации) имеет десять основных подразделений, которые называются *главными классами:* (0 — Наука в целом, информационные технологии — 004; 1 — Философия. Психология; 2 — Религия. Теология; 3 — Общественные науки; 5 — Математика. Естественные науки; 6 — Прикладные науки; Медицина. Техника; 7 — Искусство. Декоративно-прикладное искусство. Фотография. Музыка. Игры. Спорт; 8 — Языкознание. Филология. Художественная литература. Литературоведение; 9 — География. Биографии. История).

Остальные фасеты УДК представляют *вспомогательные таблицы,* предназначенные для классификации по дополнительным признакам. Вспомогательные индексы (определители) этих таблиц бывают двух видов: *общие*, которые могут соединяться с любыми индексами основной таблицы УДК, и *специальные*, присоединяемые только к понятиям текущего раздела.

Общие определители отражают категории и признаки, применяемые по всей таблице (время, место, язык, формы документов и т. д.) и служат для стандартного обозначения этих категорий и признаков.

Фрагмент УДК приведен в Приложении 2.

Международная патентная классификация[1] **(МПК)** обеспечивает достаточно полное индексирование предмета изобретения с помощью ограниченного числа рубрик за счет ориентации последних на важные с точки зрения патентоведения аспекты, такие как характеристика вещества или устройства, его функции, применение, получение. Согласно основным принципам применения Международной патентной классификации, указывающим, что изобретение, подлежащее классификации, не может рассматриваться как чистая идея в отрыве от ее технического воплощения в устройстве, способе или веществе, выделяют следующие предметы (объекты) изобретения:

1) вещество или материал;

2) устройство, прибор, конструкция;

3) процесс, способ, метод.

Изобретение рассматривается как некоторый объект, взятый относительно среды, т. е. формальное описание изобретения включает описание объекта и некоторый предикатор (аспект), отражающий его отличительную особенность. В качестве предикаторов наиболее часто используют:

- для вещества или материала: материал как таковой, его применение или получение;
- для устройства прибора или конструкции: применение, построение, функциональное назначение, принцип действия;
- для процессов: назначение, использование.

МПК охватывает все области знаний, которые могут подлежать защите охранными документами, и имеет иерархическую четырехуровневую структуру.

1. *Раздел.* Обозначается латинской заглавной буквой от A до H и снабжен *заголовком*, укрупненно отражающим его содержание.

МПК включает следующие восемь разделов:

A — удовлетворение жизненных потребностей человека;

B — различные технологические процессы, транспортирование;

C — химия, металлургия;

D — текстиль, бумага;

E — строительство, горное дело;

F — механика, освещение, отопление, двигатели и насосы, оружие, боеприпасы, взрывные работы;

G — физика;

H — электричество.

[1] Ранее МПК называлась Международная классификация изобретений (МКИ).

2. *Класс.* Каждый раздел делится на классы, обозначаемые двузначным числом. Содержание класса отражает *заголовок класса.* Некоторые классы снабжаются кратким перечнем относящейся к ним тематики — *указателем класса.*

3. *Подкласс.* Каждый класс содержит один или более подклассов, обозначаемых заглавной буквой латинского алфавита. Содержание подкласса определяет *заголовок подкласса.* Некоторые подклассы снабжаются кратким перечнем относящейся к ним тематики — *указателем содержания подкласса.*

4. *Группа, подгруппа.* Каждый подкласс разбит на подразделения, которые в дальнейшем именуются «дробными рубриками». Среди дробных рубрик различают основные группы и подгруппы.

Дробная рубрика обозначается двумя числами, разделенными наклонной чертой. Первое (максимум трехзначное, обычно нечетное) число индексирует *основную группу,* второе (минимум 2 цифры, обычно четное) — *подгруппу.* Для основной группы код подгруппы имеет значение 00. Каждую третью или четвертую цифру после наклонной черты следует понимать как дальнейшее десятичное деление предыдущей цифры. Отсюда следует, что подгруппа с индексом 5/417 должна стоять после подгруппы 5/41, но перед подгруппой 4/42.

Текст основной группы определяет область, которая считается целесообразной для проведения поиска. Текст и индексы основных групп выделены жирным шрифтом.

Текст подгруппы понимается всегда в пределах объема ее основной группы и определяет тематическую область, в которой считается целесообразным проведение поиска. Перед текстом подгруппы ставится одна точка или более, точки определяют степень ее подчиненности, т. е. указывают на то, что подгруппа является рубрикой, подчиненной ближайшей вышестоящей рубрике, напечатанной с меньшим сдвигом, т. е. имеющей на одну точку меньше.

Полный классификационный индекс МПК состоит из комбинации символов, используемых для обозначения раздела, класса, подкласса и основной группы или подгруппы.

Фрагмент МПК приведен в Приложении 3.

Государственный рубрикатор научно-технической информации (ГРНТИ) представляет собой универсальную трехуровневую иерархическую классификацию областей знания, принятую для систематизации всего потока научно-технической информации.

Рубрикатор имеет многоцелевое назначение в силу универсальности охвата тематики.

Рубрикатор имеет три уровня иерархии, при этом весь универсум знаний условно разделен на четыре подкласса:

- «Общественные науки» (значение кода первого уровня от 00 до 26);
- «Естественные и точные науки» (значение кода первого уровня от 27 до 43);
- «Технические и прикладные науки. Отрасли экономики» (значение кода первого уровня от 44 до 81);
- «Межотраслевые и комплексные проблемы» (значение кода первого уровня от 82 до 90).

Каждая рубрика состоит из кода (нотации) и наименования (описание класса), а также может иметь при себе ссылки и примечания.

На каждом уровне Рубрикатора возможно деление на 100 подклассов. Коды рубрик состоят из цепочки пар арабских цифр, разделенных точкой. Уровень рубрики, соответствующей определенной области знания, отражает не ее значимость, а только степень обобщения при логической группировке понятий.

В Рубрикаторе использовано сочетание иерархии с фасетным принципом, который проявляется в наличии совокупности рубрик, повторяющейся в разных классах в виде группы «Общие вопросы», а также в применении типовых классификационных делений в разных разделах Рубрикатора. Например, фасет «Общие вопросы» имеет одинаковую структуру в разных классах.

Наряду с иерархической классификационной структурой в Рубрикаторе с помощью *аппарата ссылок и примечаний* отражаются полииерархические связи, т. е. подчинение одного понятия двум или более классам, размещенным в разных местах иерархии. При этом могут указываться аспекты, уточняющие признаки деления понятий.

Фрагмент ГРНТИ приведен в Приложении 4.

7.4.2. Дескрипторные языки

Описательный подход в большей степени поддерживается *языками дескрипторного типа*. Они реализуют *координатное*[1] *индексирование*, которое заключается в формировании описания документа как

[1] Название обусловлено положением, предполагающим, что каждый дескриптор представляет отдельную координату в ортогональном пространстве понятий, а их совокупность — точку в этом пространстве.

совокупности дескрипторов, выбираемых из заранее созданных словарей понятий либо из текстов документов.

Метод координатного индексирования базируется на положении, что основное смысловое содержание документа и информационной потребности может быть с достаточной степенью точности и полноты выражено соответствующим списком так называемых *ключевых слов,* которые явно или в скрытом виде содержатся в тексте. Под ключевыми словами в данном случае понимаются наиболее существенные для этой цели слова и словосочетания, обладающие назывной (номинативной) функцией.

Контроль за ключевыми словами может иметь разные степени. При *отсутствии* контроля для координатного индексирования документа или информационного запроса ключевые слова выбираются непосредственно из текста документа без учета того, какие ключевые слова уже использовались ранее для индексирования таких же или близких по смыслу документов и информационных запросов. В этом случае не устраняются синонимия, полисемия и омонимия ключевых слов, а их грамматические формы даже не приводятся к нормальному виду.

При *полном* контроле за словарным составом ИПЯ для индексирования документов и информационных запросов разрешено использовать лишь *дескрипторы*, т. е. такие ключевые слова, которые содержатся в некотором нормативном списке (например, в тезаурусе). В таком списке или словаре устранены синонимия, полисемия и омонимия ключевых слов, а также обозначены определенные парадигматические связи между ними.

Контролируемый словарь предполагает ведение некоторой лексической базы данных, добавление терминов в которую проводится администратором системы (т. е. новые документы индексируются только теми терминами, которые есть в этой базе). Свободный словарь пополняется автоматически по мере появления новых терминов в новых документах.

В отличие от языка, построенного на основе классификации (УДК, рубрикаторы и др.), который позволяет потребителю легко найти свое место в информационной среде, как бы причислив себя к классу других потребителей, дескрипторный язык дает возможность «индивидуализироваться», отбирать документы по существенным только для него (потребителя) признакам. Технологически дескрипторный язык может выступать как дополнение к классификационному.

Тезаурусы

Тезаурус может быть представлен как семантическая сеть, в которой понятия связаны регулярными и устойчивыми семантическими отношениями — иерархическими (например, род-вид, целое-часть), ассоциативными, а также отношениями эквивалентности. При этом отдельное понятие определенной области знаний в тезаурусе представлено словом или словосочетанием, соотносящимся с другими словами и словосочетаниями и образующим вместе с ними замкнутую систему.

Информационно-поисковые тезаурусы позволяют решить проблему соотнесения авторской терминологии (понятий и слов естественного языка, которые автор использует для обозначения этих понятий), терминологии системы (понятий и терминов, которые используются для выражения этих понятий при вводе документов в ИПС) и терминологии потребителя (понятий и терминов, которые потребитель использует при формировании запросов).

Тезаурус состоит из контролируемого, но изменяемого словаря терминов, между которыми указаны смысловые связи. Такой словарь исчерпывающим образом покрывает некоторую специфическую область знаний и представляет собой перечень *лексических единиц*, упорядоченных по систематическому и алфавитному принципам. Кроме этого между лексическими единицами заданы смысловые отношения как иерархического (родо-видового), так и неиерархического типа (ассоциативного).

Лексические единицы тезауруса поделены на дескрипторы и *ключевые слова* — не дескрипторы.

Ключевые слова — термины естественного языка (слова или словосочетания), служащие для точного обозначения понятий определенной предметной области (предметов, явлений, свойств, отношений, процессов и т. д.). Термины, являющиеся абсолютными или условными синонимами (в рамках данной предметной области), объединяются в классы условной эквивалентности. Один из терминов класса условной эквивалентности выбирается в качестве дескриптора. Он обозначает данный класс и выражает основное значение всех слов и словосочетаний, входящих в него.

Дескрипторы — нормализованные термины естественного языка. Каждое ключевое слово, не являющееся дескриптором, но входящее в тот или иной класс условной эквивалентности, имеет отсылку к соответствующему дескриптору.

Лексические единицы тезауруса нормализованы следующим образом:

- имена существительные, обозначающие исчисляемые предметы, представлены в форме именительного падежа множественного числа;
- существительные, обозначающие неисчисляемые объекты, представлены в форме именительного падежа единственного числа;
- для всех словосочетаний-дескрипторов, включая словосочетания с именем собственным, используется естественный порядок слов.

В тезаурусе приняты три вида отношений:

- тождество (синонимия);
- подчинение (иерархические родо-видовые отношения);
- сходство (ассоциативные отношения).

Под родо-видовыми отношениями понимаются иерархические отношения между понятиями, обозначающими классы предметов, такие, что родовое понятие отражает существенные признаки всех видовых понятий, а видовое понятие содержит все признаки родового понятия, а также отражает конкретные свойства предмета или явления, выраженного этим видовым понятием.

Ассоциативные отношения — смысловые отношения дескрипторов типа часть-целое, причина-следствие, производитель-объект и т. п.

В табл. 7.1 приведены связи между дескрипторами и ключевыми словами в тезаурусе INIS с указаниями типа отношения.

Фрагмент тезауруса INIS приведен в Приложении 5.

Таблица 7.1. **Связи в тезаурусе**

Обозначение		Название	Тип отношения
Русскоязычное	Англоязычное		
см.	SEE	смотри	синонимия
см. вместо	SF	смотри вместо	синонимия
исп.	USE	используй	синонимия
исп. вместо	UF	используй вместо	синонимия
РТ	BT	вышестоящий	иерархия
ВТ	NT	нижестоящий	иерархия
АТ	RT	ассоциативный	ассоциация

7.4.3. Язык запросов документальной АИПС

Наиболее распространенным ИПЯ является язык, позволяющий составить логические выражения запроса из набора терминов. Термины могут связываться булевыми операторами AND, OR, NOT (И, ИЛИ, НЕ).

Например, запрос *((информационная AND система) OR ИПС) NOT СУБД* означает: «Найти все документы, которые содержат одновременно слова *«информационная»* и *«система»*, и/или слово *«ИПС»*, но не содержат слово *«СУБД»*.

В случае, когда система позволяет создавать запросы на «естественном языке», фраза запроса обычно разбивается на слова, из этого списка удаляются запрещенные и общие слова, производится нормализация лексики, а затем все слова связываются либо логическим AND, либо OR. Таким образом, запрос: *Программы для Unix* будет преобразован в *Unix AND Программы,* что будет означать следующее: «Найти все документы, в которых слова *«Unix»* и *«программы»* встречаются одновременно».

Кроме обычного набора логических операторов *AND, OR, NOT* большинство систем позволяет использовать контекстные операторы *NEAR, CTX, SENT, PAR*, обеспечивающие уточнение запроса требованием взаимного расположения терминов в документе.

Поскольку все документы обычно состоят из полей, в запросе можно указать, в какой части (поле) документа пользователь хочет увидеть поисковый термин (в ссылке, заголовке и т. п.). Для отдельных систем можно также задать поле, по значению которого будет упорядочена выдача.

Информационно-поисковый запрос как операционный объект представляет собой совокупность семантически связанных[1] *предложений запроса*. Понятие «Запрос» надо отличать от понятия «Поисковый образ запроса». ПОЗ — это лингвистическая конструкция, задающая условие отбора документов, в то время как запрос отражает объединенную общей тематикой последовательность поисковых действий, направленных на получение полного и логически завершенного результата.

То есть, с одной стороны, запрос — это несколько отдельных условий, позволяющих «по частям» выразить ИП, а с другой — это совокупность результатов, каждый из которых не только отвечает неко-

[1] На практике семантическая связь предложений между собой синтаксисом ИПЯ поддерживается редко.

торой части ИП, но и, возможно, получен своим методом поиска. Это позволяет избирательно использовать результаты отдельных предложений, объединять поисковые результаты, выделять общее множество релевантных документов и т. п. Пример языка запроса приведен в Приложении 6.

7.4.4. Индексирование и реферирование

Ранее понятие индексирования рассматривалось в основном с точки зрения организации физического доступа к документу (точнее, к записи в БД). В этом разделе индексирование и реферирование рассматриваются как методы концентрации, сжатия информации.

Важно понимать, что целесообразность сжатия определяется вовсе не необходимостью экономии машинной памяти, а особенностями характера самого процесса поиска. Во-первых, как отмечалось ранее, собственно отбор сводится к простейшей процедуре сопоставления машинных кодов, представляющих запрос и документ. Во-вторых, характер взаимодействия потребителя с системой имеет «телеграфный» стиль: запрос очень похож на простое перечисление основных понятий. Это происходит не по причине лености пользователя. При поиске иначе и быть не может: подробное изложение существа проблемы не только займет все время пользователя, но и заставит сосредоточиться на *отдельном* решении, в то время как задачи поиска обычно имеют обратную направленность — отыскать как можно больше вариантов и подходов к решению, среди которых пользователь сможет впоследствии (после поиска) выбрать наиболее подходящие.

В документальных системах в качестве таких смысловых сверток используют рефераты, аннотации и поисковые образы. При такой обработке полного текста основная задача состоит в возможно более полном извлечении из документа фактической информации. По существу мы переходим от непрерывного текста (точнее, непрерывного изложения содержания и, соответственно, последовательного и непрерывного процесса построения знания в сознании воспринимающего) к дискретной форме перечисления основных понятий, используемых источником и приемником информации для обозначения содержания в краткой и не избыточной перечислительной форме.

Реферирование предполагает извлечение из документа основных положений содержания и их представление в виде реферата.

Индексирование текста (в основном, автоматическое) предполагает выделение (или приписывание) слов и словосочетаний, обозначающих основные понятия, образующие содержание документа.

При индексировании часто используются списки запрещенных слов, которые не могут быть включены в ПОД (общие слова, предлоги, союзы и т. п.), а также иногда применяется нормализация лексики. Таким образом, даже то, что называется полнотекстовым индексированием, реально является выбором слов из текста документа и результатом сравнения с целым набором различных словарей, после чего термин попадает в поисковый образ документа, а потом и в индекс системы.

Например, для следующего реферативно-библиографического описания

> Trosow Samuel E. Управление библиотечным и информационным центром. Libr Quart., 2000, № 70, 153-155.
> Рецензируемая книга давно стала стандартным учебником по курсу управления в рамках библиотечной и информационной науки, охват материала в котором расширялся с каждым очередным изданием. В предисловии отмечается, что значительные изменения в окружающих условиях, вызванные внутренними и внешними факторами, требуют более систематичного подхода к обзору функций в условиях организации. Технология, политическая, экономическая и социальная среда указываются как наиболее мощные силы для изменений. Помимо отдельных глав, посвященных перечисленным вопросам, авторы уделяют большое внимание теории организации, связи социологической теории и ее применений в организационном анализе, различным уровням анализа, на которых могут изучаться организационные явления.

по содержанию реферата будет построен поисковый образ с использованием нормализованной лексики, включающий термины: библиотеки, информационные центры, управление, функции руководителей, рецензии, США.

ПОД по заголовку документа будет построен на основе свободного индексирования и будет включать отдельные слова: управление, библиотечным, информационным, центром.

7.5. Средства информационного поиска

7.5.1. Механизмы отбора документальной информации

Булев поиск. Наиболее широко используются механизмы отбора документов, в основе которых лежит какая-либо модификация вычисления *булева выражения* (правила формирования которого опреде-

ляются конкретным ИПЯ, например, представленным в приложении 6), соотносящего множество терминов запроса и множество терминов документов базы данных.

Модификацией булевого поиска является *взвешенный булев поиск*. Запрос может формулироваться на ИПЯ, описанном, например, в приложении 6, но выдача документов при этом будет ранжироваться в зависимости от степени близости запроса и документа.

Поиск «по сходству» (документы-аналоги, «Like this»). Отбор таких документов, которые имеют заданное количество общих терминов с исходным. То есть в качестве запроса используются термины (обычно отображенные на словарь системы) из документа, которому ищутся подобные.

Поиск с коррекцией запроса по релевантности. Такой поиск является уже *интерактивным итеративным процессом*. После проведения первичного поиска с использованием какого-либо метода пользователь отмечает в списке найденных документов истинно релевантные, т. е. соответствующие его информационной потребности, а не просто содержащие термины запроса. Некоторые системы имеют для этого специальное поле (область при документе), где пользователь может отметить документ как релевантный. При следующей итерации система уже сама расширяет запрос пользователя терминами из релевантных документов и снова выполняет поиск. Для построения словников на основе лексики документов, определяемых пользователем как истинно релевантные, используется «внешняя» обратная связь. Для построения реформулированного запроса используется уже «внутренняя» обратная связь, позволяющая пользователю (непосредственно) или системе (ранжированием или кластеризацией по статистическим показателям) выделить семантически значимые термины.

7.5.2. Постобработка поисковой выдачи

Основное назначение постобработки — снижение размерности пространства документов, которое необходимо «обработать» пользователю для получения уверенности в том, что полученная информация адекватна — полно и точно отражает состояние предметной области в аспекте задачи конкретного пользователя. Именно наличие конкретных особенностей содержания, формы представления или будущего использования информации позволяет предположить, что

часть найденных формальными методами документов может заранее (до просмотра) считаться малозначимой, т. е. не обладающей *существенными признаками*. Соответственно, если множество найденных документов будет упорядочено в выдаче по убыванию значения этого признака, то пользователь достаточно просто, но *обоснованно* может принять решение о завершении просмотра.

Такое упорядочение (ранжирование) документов достаточно просто реализуется, если в качестве *существенного признака* выступает, например, «свежесть» публикации: для этого система отсортирует документы по значению поля «Дата публикации». Существенно сложнее процедура в случае ранжирования по степени смыслового соответствия. Сложность предопределяется двумя факторами. Во-первых, системе достаточно трудно точно и однозначно установить действительный смысл реальной потребности пользователя по обычно очень короткому выражению запроса. Во-вторых, практически невозможно выбрать *адекватную* меру соответствия, поскольку она должна вычисляться по формальным *количественным* признакам, в то время как смысловое соответствие имеет *качественный* характер.

В технологиях информационного поиска используются две методики упорядочения. Первая — классификация, основана на сведении запроса к общепринятой (по крайней мере, в данной предметной области) системе классификации. Вторая — кластеризация, основана на предположении, что предметная область (и взгляды на нее) обладает свойством структурированности, и выданные документы могут быть разделены в соответствии с аспектами, один или несколько из которых, вероятно, будут соответствовать реальной потребности.

Классификация и кластеризация представляют собой две противоположные технологии: классификация заключается в автоматическом отнесении (определении тематики) документа к одному из классов, определенных на известном множестве признаков, в то время как задачей кластеризации является автоматическое построение классов семантически подобных документов. То есть в случае классификации на первом этапе задается система признаков, определяющих фиксированное количество классов, а на втором — документы распределяются между этими классами. В случае кластеризации множество документов сразу разбивается на кластеры по степени близости признаков, которыми они обладают сами, т. е. происходит не столько выявление, облает ли документ признаками, заранее объявленными в классификационной системе, столько выявление признаков, которые могут быть классификационными.

7.6. Поисковый интерфейс

Поисковый интерфейс (интерфейс пользователя) — это программа, обеспечивающая формирование запросов, просмотр найденных документов и управление поисковой сессией, включая, например, выбор ресурсов, установление параметров отбора, использование ранее полученных результатов и т. д.

Наибольшее разнообразие форм и возможностей (и проблем для пользователя) демонстрируют средства, которые пользователь должен применять при подготовке запроса для его обработки системой. Тем не менее, здесь можно выделить два класса решений. Первые (в том числе и в действиях пользователя) — «вербальные» — представлены средствами подготовки выражения поискового условия. Вторую группу составляют средства, позволяющие представить запрос в так называемой «кластерной» форме, например, документом (поиск аналогов), множеством документов или терминов.

7.6.1. Средства представления запроса

Средства подготовки запроса в вербальной форме. Как отмечалось ранее, все ИПС явно или неявно используют поисковые условия, основанные на алгебре логики.

По форме (структуре) диалога и, соответственно, по способу задания условия отбора средства такого рода можно разделить на две категории: *рубрикационного типа* и *структурно-логические*. Первые реализуются в виде иерархических, последовательно раскрывающихся списков, через которые обеспечивается доступ к тематически связанным группам документов. Раскрывая очередную рубрику и перемещаясь, таким образом, по тематической иерархии, пользователь уточняет предметную область и увеличивает степень соответствия выдаваемых документов своей информационной потребности. При таком решении недостатки предопределенности, а иногда и неточности соотнесения документов с отдельными рубриками компенсируются логичностью естественно-научной классификационной схемы, заменяющей пользователю путеводитель.

Структурно-логические средства подготовки запроса — это меню-подобные схемы и процедуры, имитирующие посредничество и позволяющие в интерактивном режиме «сконструировать» булево-подобное выражение, удовлетворяющее требованиям ИПЯ системы.

Кластерные формы представления запроса. В этом случае для спецификации потребности используются документы как образцы для поиска подобных. Инициирование поиска осуществляется пользователем в момент просмотра отдельного документа, который им признан *истинно релевантным*.

Функция поиска документов-аналогов использует наиболее информативные термины текущего документа, при этом пользователь в диалоге может указать поля, термины которых будут использованы системой для построения ПОЗ и, возможно, логику и пороги критерия соответствия.

Интерфеис поиска по обратнои связи реализуется последовательностью шагов:

- отбор и выделение множества релевантных документов;
- построение ранжированного словника релевантных документов;
- выбор терминов из словника для формирования документального пространства;
- разбиение построенного информационного пространства на кластеры и предоставление пользователю документов каждого кластера.

7.6.2. Интерфейсные средства обработки результатов и технология поиска

Интерфейсные средства обработки результатов поиска позволяют работать как с отдельными документами, так и с их коллекциями.

Интерфеисная форма «Документ» помимо традиционных средств управления просмотром и выводом может включать средства изменения формы представления материала, а также средства инициирования процедур поиска, использующих лексику документа. В ряде случаев (обычно для БД вторичной информации) интерфейсная форма позволяет обращаться к текстам соответствующих первоисточников, размещенных в локальной среде или во внешних ресурсах.

Интерфейсная форма «Протокол запросов» позволяет работать с результатами поисков, обеспечивает разделение результата на подмножества по уровню релевантности, сортировку документов по значениям указанных реквизитов, а также сохранение запросов и результатов всех или выделенных пользователем предложений для последующей обработки.

7.6.3. Обобщенная технологическая схема поиска

С точки зрения взаимодействия «пользователь — система», где роль системы — не более чем информационно-технологическая поддержка, процесс информационного поиска в общем случае может быть представлен как *навигация* — целенаправленное и управляемое перемещение в документальном и лексическом пространстве базы данных, обеспечивающее оцениваемый уровень удовлетворения информационной потребности или объективно подтверждающее отсутствие информации.

Целенаправленность здесь предполагает наличие некоторой цели, обычно в сфере основной деятельности, которая в свою очередь может быть представлена как комплекс локальных (информационных) целей тематического (многоаспектный поиск) и/или технологического типа.

Управляемость — это, с одной стороны, возможность выбора средств и/или параметров работы, а с другой — выборочное, в том числе повторное, обращение к результатам и их обработка (например, статистическая или структурно-форматная). Кроме того, поскольку выделенная последовательность результатов (физически соответствующая отдельным шагам поиска, а логически — отдельной цели) образует ряд, то это позволяет использовать статистически вычисляемые показатели (разностного типа), характеризующие сходимость процесса поиска и, в частности, обеспечивающие некоторую обоснованность принятия решения об окончании процесса совершенствования запроса.

Для приведенной на рис. 7.6 обобщенной технологической схемы поиска ИАС xIRBIS [xIRBIS] классическая схема выдачи документов «по запросу-выражению» расширена до динамически управляемого процесса кластеризации пространства документов и терминов.

При этом процесс поиска может развиваться по принципу «расходящихся кругов», обеспечивая выявление «центров активации» искомого образа в семантической сети базы данных, т. е. построение множеств или цепочек документов, которые в свою очередь могут служить мостом к понятиям (документам), возможно, не содержащим терминов исходного запроса. Пользователь может продвигаться по пути (реализовать навигацию), предлагаемому системой, или же изменять его, в том числе и выбирая из сформированных системой альтернатив либо инициируя новый путь через процедуру поиска или прямого отбора.

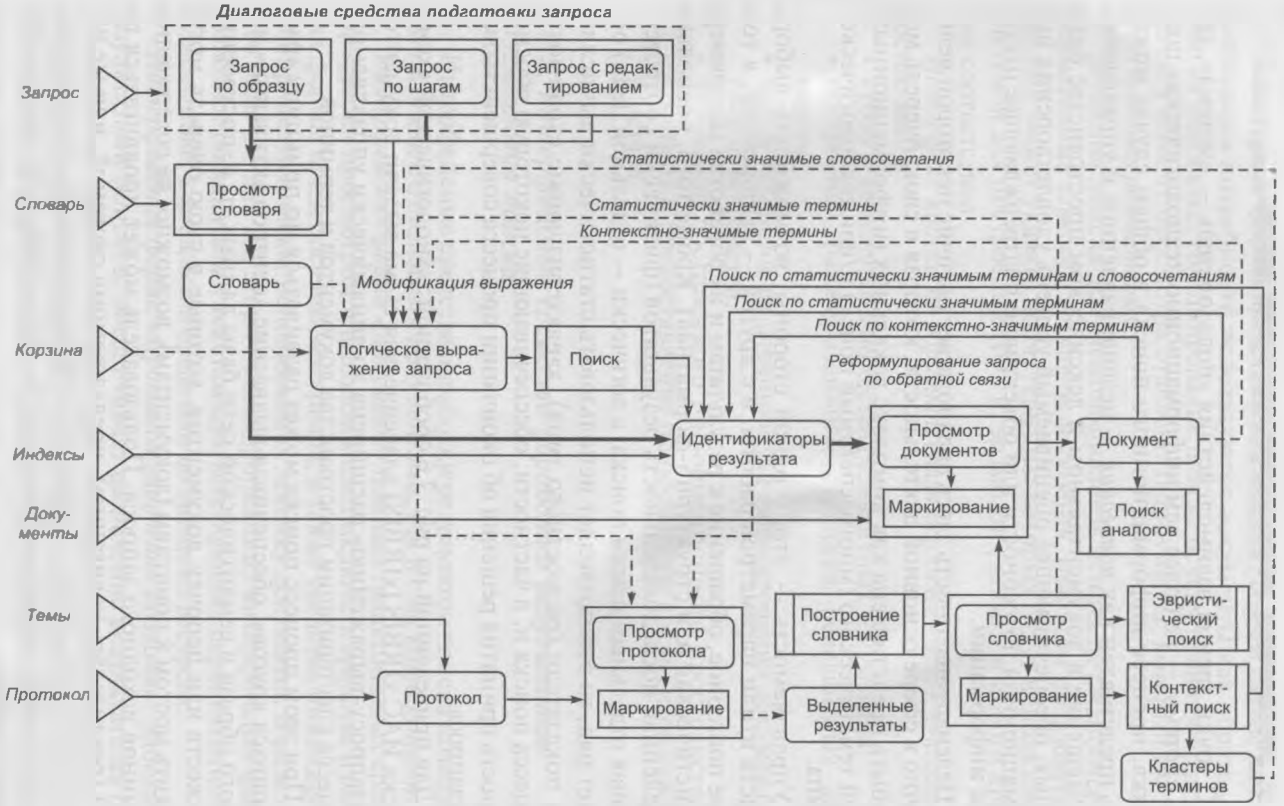


Рис. 7.6. Обобщенная технологическая схема поиска

Несмотря на то что основной задачей поискового интерфейса является проложение путей к документу и в итоге — получение текста, в контексте двойственности цели поискового процесса рассмотренный процедурный интерфейс построен симметрично: при подготовке ПОЗа можно формировать документальную часть результата, а при формировании результата (поиске, отборе, просмотре, реформулировке запроса) — строить запрос.

Доказательство полноты, которая реально не может быть формально вычислена (по причине принципиальной невозможности полного знания о существующих или создаваемых решениях), компенсируется подтверждаемостью — получением результата другим путем, например, вхождением в информационное пространство БД через информационные объекты разной природы и/или использованием поисковых механизмов разного типа.

В заключение отметим, что интерфейсные средства собственно и являются для пользователя теми инструментами, которые он выбирает и использует для реализации конкретной поисковой траектории (навигации в базе данных).

Обобщенная технологическая схема поиска, приведенная на рис. 7.6, наглядно демонстрирует, можно сказать, парадоксальную особенность поискового процесса: схема подобна лабиринту, имеющему много явно заметных входов, но выход не указан — его должен определять пользователь индивидуально в зависимости от собственных целей, наличных ресурсов и способностей.

Контрольные вопросы

1. Перечислите основные операции процесса поиска информации.
2. Охарактеризуйте технологические составляющие информационного поиска.
3. Приведите типологию поисковых задач и примеры поисковых задач каждого типа.
4. Охарактеризуйте типы информационной потребности.
5. Определите условия установления соответствия информационной потребности и содержания документа БД.
6. Охарактеризуйте основные этапы процесса информационного поиска.
7. Перечислите основные и технологические объекты, используемые при поиске.

8. Перечислите типы информационной потребности пользователя и определите их связь с уровнями информационных объектов.
9. Дайте сравнительную оценку характера деятельности человека и компьютерной системы.
10. Дайте определение понятия «интерфейс пользователя».
11. Охарактеризуйте влияние интерфейсных средств на адаптацию пользователя.
12. Приведите примеры диалоговых интерфейсных средств обучения пользователя работе с АИПС и базой данных.
13. Проведите сравнительный анализ вербальной и кластерной стратегий поиска.
14. Определите зависимость методов построения запроса и стратегий поиска.
15. Определите назначение «обратной связи» в процессе информационного поиска.
16. Перечислите информационные объекты, используемые для реализации технологии «обратной связи» в процессе информационного поиска.

Глава 8
РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ИНФОРМАЦИИ

Исходя из введенного в первой главе положения, что информация — это данные и контекст их обработки, будем рассматривать архитектуры, реализующие совместную обработку информации, выделяя вопросы, относящиеся к собственно обработке данных, и вопросы, связанные с контекстом — функциональностью обработки метаинформации.

8.1. Распределенные вычисления

Под распределенными вычислениями понимают такие вычисления, которые производятся на оборудовании, состоящем из более чем одного процессора или блока оперативной памяти.

Обычно при распределенных вычислениях программы разбиваются на части, которые исполняются на множестве компьютеров, объединенных сетью. Распределенные вычисления — это вид параллельных вычислений. Однако следует заметить, что в параллельных вычислениях обычно используются разделенные на части программы, которые обрабатываются на множестве процессоров одного компьютера.

Программы, разработанные для распределенных вычислений, отличаются от программ для параллельных вычислений тем, что их части между собой связаны слабее, предназначены для вычислений на различном оборудовании и приспособлены к межсетевому обмену данными. Кроме того, они должны быть устойчивы к проблемам связи и сбоям вычислительной техники.

При разработке программного обеспечения распределенных вычислений чрезвычайно важно учитывать следующие факторы: возможность организации межпрограммного взаимодействия через разнородные компьютерные сети и возможность транспортировки самих программ по сети для их исполнения на удаленных компьютерах с возможностью взаимодействия между собой.

Главной целью распределенных вычислений является предоставление пользователям имеющихся в наличии распределенных вычислительных мощностей. При этом для пользователей использование этих мощностей должно быть прозрачным, открытым и легко масштабируемым.

При этом под открытостью понимается способность вычислительной системы взаимодействовать с другой системой в стиле и по стандартам Открытых Систем. Часто под такими стандартами понимают стандарты Web Services, хотя это и сужает понятие открытости.

Прозрачность подразумевает свободное перемещение кода и его исполнение на любой системе, а также взаимодействие с любой другой системой, исполняющей другие части кода.

Масштабируемость подразумевает возможность использования любых дополнительных ресурсов по мере надобности в процессе вычислений.

К основным проблемам распределенных вычислений следует отнести: надежность, сложность диагностики отказов, невозможность распараллеливания многих вычислительных задач.

8.1.1. Архитектура распределенных вычислений

Для организации систем распределенных вычислений используются самые разные модели межпрограммного обмена данными и транспортировки программного кода. Остановимся на базовых моделях.

Модель «клиент—сервер». В этой модели клиент отправляет запросы на сервер, который на эти запросы отвечает. Взаимодействие инициирует клиент. Сервер постоянно находится в ожидании возможных запросов.

Модель с посредником между клиентом и сервером. В этой модели основная нагрузка по взаимодействию с сервером ложится на посредника, который собирает запросы от «легких» клиентов. В протоколе взаимодействия клиента и посредника по сравнению со схемой «клиент—сервер», как правило, используется множество допущений, призванных существенно упростить разработку приложений-клиентов.

Модель с серверами приложений. В этой модели между клиентом и системой управления информационными ресурсами, которая выполняет функции универсального сервера, появляется цепочка серверов-посредников, которые реализуют различные функции по агрегированию запросов и ответов от групп клиентов и различных серверных подсистем. Такая модель характерна для больших корпоративных информационных систем и систем управления.

Модель кластеров. Кластерная модель применяется при организации параллельных вычислений на группах совместно работающих компьютеров, которые называются кластерами. Такие комплексы имеют специальную архитектуру, оптимизированную для решения задач, допускающих высокую степень параллелизма отдельных операций.

Модель «точка—точка». Данная модель предполагает отсутствие единого управляющего центра и единого центрального информационного ресурса. Все компьютеры в такой модели равноправны и могут выступать как в качестве клиентов, так и в качестве серверов.

Модель единого пространства. В этой модели вся вычислительная архитектура предоставляется конечному потребителю в виде единого вычислительного пространства. Все операции по распараллеливанию вычислений, использованию распределенного адресного пространства, обмену данными между частями программы скрыты от пользователя. Ресурсы наращиваются по мере необходимости, как если бы для вычислений использовался только один компьютер.

8.1.2. Виды параллелизма в распределенных вычислениях

Несмотря на развитие в последнее время различных способов виртуализации представления о реальной инфраструктуре вычислений, следует признать, что во многом распределенные вычисления ориентируются на архитектуру современных вычислительных комплексов. С другой стороны, и сами эти комплексы долгое время создавались под определенные вычислительные задачи. В настоящее время выделяют несколько системных архитектур, ориентированных на распределенные вычисления.

Многопроцессорные системы. Как правило, это компьютеры, которые имеют более одного процессора, а операционная система способна распределять задачи или потоки (нити) задач по различным

процессорам, добиваясь одновременного использования всех процессоров.

Многоядерные системы. Данные архитектуры специально предназначены для исполнения легковесных процессов или потоков, на которые разбивается задача при ее исполнении. При этом предполагается, что различные задачи могут использовать одни и те же потоки, но с различными данными.

Мультикомпьютерные системы. Данная архитектура предполагает согласованное использование нескольких компьютеров для решения одной задачи или комплекса взаимосвязанных задач. При этом связность компонентов задачи гораздо ниже, нежели при решении задач с использованием кластеров.

Вычислительные кластеры. Кластеры — это многомашинные вычислительные комплексы, компьютеры которых объединены локальными высокоскоростными линиями связи. Довольно часто кластерные вычисления отличают от прочих распределенных вычислений. Кластеры решают групповые задачи, требующие высокой согласованности вычислений, что, в свою очередь, требует компактного совместного размещения компьютеров, объединенных в кластер. При «традиционных» распределенных вычислениях таких тесных связей между компьютерами не требуется. Для решения задач распределенных вычислений часто применяются компьютеры, размещенные на площадках, существенно удаленных друг от друга.

Грид-вычисления. Грид-вычисления (GRID-Computing) — это один из видов распределенных вычислений, когда принято оперировать понятием виртуального суперкомпьютера, который на самом деле состоит из множества слабосвязанных вычислительных машин. При этом виде вычислений, как правило, используются вычислительные ресурсы, которые в данный момент времени простаивают. Это могут быть суперкомпьютеры или даже множество персональных компьютеров. Обычно в качестве связи используют Интернет.

Грид-архитектуру от кластерной отличает слабая связность компонентов системы, разнородность оборудования и системного программного обеспечения, а также географическая удаленность компьютеров друг от друга. Грид-вычисления принято рассматривать как самостоятельный вид вычислений, а не только как подвид распределенных вычислений.

Суть Грид — это возможность, используя набор стандартов и протоколов, предоставить доступ к приложениям и данным, вычисли-

тельным мощностям, памяти и множеству других вычислительных ресурсов через Интернет. Основные преимущества Грид-вычислений — это возможность за низкую стоимость получить высокопроизводительную масштабируемую систему.

Однако существуют и значительные ограничения по типам задач, которые могут быть решены на основе Грид-архитектур. Во-первых, из-за разнородности компьютеров, которые объединяются в Грид, разработка программного обеспечения превращается в сложную и нетривиальную задачу. Стоимость такой программы может существенно превышать стоимость обычного ПО. Во-вторых, слабая связность компьютеров и ненадежность коммуникаций заставляют уделять значительное внимание протоколам обмена информацией и согласованию расчетов в ненадежной вычислительной среде, что также повышает стоимость ПО. В-третьих, существует проблема стоимости каналов обмена информацией и самих вычислительных мощностей. В-четвертых, разрешение исполнения «чужих» программ на своих вычислительных мощностях чревато проблемами с безопасностью.

Интернет-вычисления. Этот тип вычислений предполагает использование интернет-технологий для организации распределенных вычислений. В основе интернет-вычислений (Cloud computing — «облачные вычисления») лежит концепция «программное обеспечение как сервис». На основе этой модели развиваются многие современные технологии, такие, например, как Web 2.0.

Интернет-вычисления — это, в некотором смысле, развитие идеи Грид-вычислений.

Во-первых, в интернет-вычислениях была успешна решена проблема биллинга. Это значит, что была найдена жизнеспособная коммерческая схема использования распределенных вычислительных мощностей.

Во-вторых, во многом благодаря использованию простаивающих мощностей интернет-компаний, например, Google, Amazon, Microsoft, удалось создать достаточно надежную распределенную децентрализованную инфраструктуру вычислений.

В-третьих, интернет-вычисления опираются не на набор библиотек для сбора программ, которые могли бы исполняться на разнородной архитектуре, а на программные интерфейсы приложений (API), которые позволяют представлять вычислительную среду в качестве виртуального компьютера и разрабатывать ПО для этого компьютера,

не заботясь о конкретной архитектуре вычислительных систем, задействованных в интернет-вычислениях.

При интернет-вычислениях пользователь арендует вычислительные мощности, память и каналы обмена данными. При этом пользователь платит только за те объемы, которые реально использует, а не за возможность использования ресурсов.

8.2. Распределенные базы данных

8.2.1. Основные условия и требования к распределенной обработке данных

Такая отличительная особенность БД, как многоцелевое параллельное использование данных, предопределяет наличие средств, обеспечивающих практически одновременный и независимый доступ к одним и тем же данным. Причем сама база может быть размещена на одном или нескольких компьютерах.

В [Дейт, 2001] приводятся следующие сформулированные ведущими поставщиками СУБД свойства «идеальной» системы управления распределенными базами данных:

- *прозрачность относительно расположения данных*: все данные должны представляться так, как если бы они были локальными;
- *гетерогенность системы*: должна быть обеспечена работа с данными, которые хранятся в системах с различной архитектурой и производительностью (независимость от СУБД на отдельной ЭВМ);
- *прозрачность относительно сети*: должна быть обеспечена одинаково эффективная работа в условиях разнородных сетей;
- *поддержка распределенных запросов*: пользователь должен иметь возможность объединять данные из любых баз, даже если они размещены в разных вычислительных системах;
- *поддержка распределенных изменений*: пользователь должен иметь возможность изменять данные в любых базах, на доступ к которым у него есть права, даже если эти базы размещены в разных вычислительных системах;
- *поддержка распределенных транзакций*: должны выполняться транзакции, выходящие за рамки одной вычислительной системы, и поддерживаться целостность распределенной БД даже

при возникновении отказов как в отдельных вычислительных системах, так и в сети;

- *безопасность*: должна быть обеспечена защита всей распределенной БД от несанкционированного доступа;
- *универсальность доступа*: должна использоваться единая методика доступа ко всем данным.

Однако ни одна из существующих СУБД не достигает этого идеала вследствие следующих практических проблем:

- Низкая и несбалансированная производительность сетей передачи данных, что в распределенных транзакциях сильно снижает общую производительность обработки.
- Обеспечение целостности данных в распределенных транзакциях базируется на принципе «все или ничего» и требует специального протокола двухфазного завершения транзакций, что приводит к длительной блокировке изменяемых данных.
- Необходимость обеспечения совместимости данных стандартного типа, для хранения которых в разных системах используются разные физические форматы и кодировки.
- Выбор схемы размещения системных каталогов. Если каталог будет храниться централизованно, то удаленный доступ будет замедлен. Если будет размножен — то изменения придется распространять и синхронизировать.
- Необходимо обеспечить совместимость СУБД разных типов и поставщиков.
- Увеличение потребностей в ресурсах для координации работы приложений с целью обнаружения и устранения тупиковых ситуаций в распределенных транзакциях.

Именно указанные причины определили на практике частичность и «этапность» введения в СУБД тех или иных возможностей распределенной обработки данных. В простейшем случае пользователь имеет возможность обращаться по сети к записям в БД, размещенным на других компьютерах. В других случаях СУБД сама производит аутентификацию удаленного клиента и устанавливает сетевые соединения.

В общем случае режимы работы с БД можно классифицировать по следующим признакам:

- *количество одновременно выполняемых задач* — однопользовательский или многопользовательский;
- *правило обслуживания запросов* — последовательное или параллельное;

- *схема размещение данных* — централизованная или распределенная БД.

Следует отметить, что общая тенденция развития технологий обработки данных вполне соответствует этапам развития средств вычислительной техники и информационных технологий, и в первую очередь — сетевых. В этом смысле следует выделить два класса: *системы распределенной обработки данных* и *системы распределенных баз данных*.

Системы распределенной обработки данных в основном отражают структуру и свойства многопользовательских операционных систем с базой данных, размещенной на большом центральном компьютере (мэйнфрейме). Еще до недавнего времени это был единственно возможный вариант вычислительной среды для реализации больших баз данных. Клиентские места в этом случае реализовывались в виде терминалов или мини-ЭВМ, обеспечивающих в основном ввод-вывод данных и не имеющих собственных вычислительных ресурсов для функционально-ориентированной обработки.

Развитие сетевых технологий в сочетании с широким распространением персональных ЭВМ и внедрением стандартов открытых систем привело к появлению систем баз данных, размещенных в сети разнотипных компьютеров. Такие *системы распределенных баз данных* обеспечивают обработку распределенных запросов, когда при обработке одного запроса используются ресурсы базы, размещенные на различных ЭВМ сети. Система распределенных баз данных состоит из узлов, каждый из которых является СУБД, а узлы взаимодействуют между собой так, что база данных любого узла будет доступна пользователю, как если бы она была локальной.

Соответственно, программы, обеспечивающие целевую (функциональную) обработку данных, могут быть организованы таким образом, чтобы обеспечить более эффективное использование совокупных вычислительных ресурсов за счет специализированного разделения функций обработки между центральным процессом СУБД и клиентскими функционально-ориентированными процедурами.

Для «типового» приложения обработки данных можно выделить следующие группы (уровни) функций:

- ввод и отображение данных: внешний (пользовательский) уровень реализации целевой функциональной обработки и представления (PL — Presentation Logic);
- функциональная обработка, реализующая алгоритм решения задач пользователя; соответствующие «*бизнес-правила*» реализу-

ются обычно средствами высокоуровневого языка программирования или расширенного языка манипулирования данными типа ADABAS Natural или 4-GL (BL — Business Logic);

- манипулирование данными БД в рамках приложения, которое обычно реализуется средствами SQL (DBL — Database Logic). Кроме того, средствами SQL, помимо операций манипулирования данными (Data Management Logic — извлечения, изменения и т. д.) реализуются общие для БД функции (CDBL — Common DB Logic), например, правила целостности, типовые представления, которые, по существу, являются *общими «бизнес-правилами»* на уровне данных;
- управление ресурсами БД, реализуемое специализированными средствами конкретной СУБД (RL — Resource Logic);
- управление процессами обработки: связывание и синхронизация процессов обработки данных разного уровня.

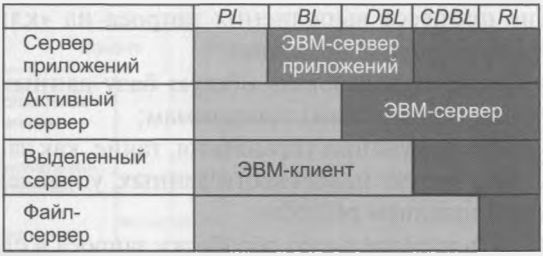


Рис. 8.1. Разделение функций в базовых архитектурах распределенной обработки

Рассматриваемые ниже архитектуры распределенной обработки в целом могут характеризоваться диаграммой, представленной на рис. 8.1.

8.2.2. Архитектура распределенной обработки данных

Почти все модели организации взаимодействия пользователя с базой данных построены на основе модели «клиент—сервер». То есть предполагается, что каждое такое приложение отличается способом распределения функций ранее приведенных групп обработки данных между как минимум двумя частями:

- клиентской, которая отвечает за целевую обработку данных и организацию взаимодействия с пользователем;

- серверной, которая обеспечивает хранение данных, обрабатывает запросы и посылает результаты клиенту для специальной обработки.

В общем случае предполагается, что эти части приложения функционируют на отдельных компьютерах, т. е. к серверу БД с помощью сети подключены компьютеры пользователей (клиенты).

Сервер — это программа, реализующая функции собственно СУБД: определение данных, запись-чтение данных, поддержка схем внешнего, концептуального и внутреннего уровней, диспетчеризация и оптимизация выполнения запросов, защита данных.

Клиент — это различные программы, написанные как пользователями, так и поставщиками СУБД, внешние или «встроенные» по отношению к СУБД. Программа-клиент организована в виде приложения, работающего «поверх» СУБД и обращающегося для выполнения операций над данными к компонентам СУБД через интерфейс внешнего уровня[1].

Разделение процесса выполнения запроса на «клиентскую» и «серверную» составляющую позволяет:

- одновременно использовать общую базу данных различным прикладным (клиентским) программам;
- централизовать функции управления, такие, как защита информации, обеспечение целостности данных, управление совместным использованием ресурсов;
- обеспечивать параллельную обработку запроса в случае распределенных БД;
- высвобождать ресурсы рабочих станций и сети;
- повышать эффективность управления данными за счет использования ЭВМ, специально разработанных для работы СУБД (серверы баз данных и машины баз данных).

8.2.3. Базовые архитектуры распределенной обработки

Учитывая, что одним из основных показателей эффективности сетевой обработки данных является время обслуживания запроса, рассмотрим различные модели архитектуры распределенной обработки на примере, когда прикладная программа работы с базой данных,

[1] Инструментальные средства, в том числе и утилиты, не отнесены к серверной части очень условно. Являясь не менее важной составляющей, чем ядро СУБД, они выполняются самостоятельно, как пользовательское приложение.

расположенной на сервере, загружена на рабочую станцию, и пользователю необходимо получить все записи, удовлетворяющие некоторым поисковым условиям.

Архитектура «файл-сервер»

В архитектуре «файл-сервер», схема которой представлена на рис. 8.2, средства организации и управления базой данных (в том числе и СУБД) целиком располагаются на машине клиента, а база данных, представляющая собой обычно набор специализированных структурированных файлов — на машине-сервере. В этом случае серверный компонент представлен даже не средствами СУБД, а сетевыми составляющими операционной системы, обеспечивающими удаленный разделяемый доступ к файлам. Таким образом, «файл-сервер» представляет собой вырожденный случай клиент-серверной архитектуры.

Рис. 8.2. Архитектура «файл-сервер»

Взаимодействие между клиентом и сервером происходит на уровне команд ввода-вывода файловой системы, которая возвращает запись или блок данных. Запрос к базе, сформулированный на языке манипулирования данными, преобразуется самой СУБД в последовательность команд ввода-вывода, которые обрабатываются операционной системой машины-сервера.

Достоинство — возможность обслуживания запросов нескольких клиентов.

Недостатки:

- высокая загрузка сети и машин-клиентов, так как обмен идет на уровне единиц информации файловой системы — физических

записей, блоков или даже файлов, из которых на машине клиента будут выбраны и представлены необходимые для приложения элементы данных;
- низкий уровень защиты данных, так как доступ к файлам БД управляется общими средствами ОС сервера;
- низкий уровень управления целостностью и непротиворечивостью информации, так как бизнес-правила функциональной обработки, сосредоточенные на клиентской части, могут быть противоречивыми и несихронизированными.

В среде файлового сервера программа управления данными, которая выполняется на машине-клиенте, должна осуществить запрос каждой записи базы, после чего она может определить, удовлетворяет ли запись поисковым условиям, лишь после этого передать для функциональной обработки. Очевидно, что для этой схемы характерно наибольшее суммарное время обработки информации.

Архитектура «выделенный сервер базы данных»

В архитектуре сервера базы данных, схема которой представлена на рис. 8.3, средства управления базой данных и база данных размещены на машине-сервере.

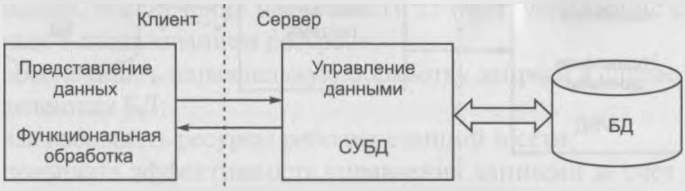


Рис. 8.3. Архитектура с выделенным сервером базы данных

Взаимодействие между клиентом и сервером происходит на уровне команд языка манипулирования данными СУБД (обычно SQL), которые обрабатываются СУБД на машине-сервере. Сервер базы данных осуществляет поиск записей и анализирует их. Записи, удовлетворяющие условиям, могут накапливаться на сервере, и после того как запрос будет целиком обработан, пользователю на клиентскую машину передаются все логические записи (запрашиваемые элементы данных), удовлетворяющие поисковым условиям.

Достоинства:
- возможность обслуживания запросов нескольких клиентов;
- снижение нагрузки на сеть и машины сервера и клиентов;

- защита данных осуществляется средствами СУБД, что позволяет блокировать неразрешенные пользователю действия;
- сервер реализует управление транзакциями и может блокировать попытки одновременного изменения одних и тех же записей.

Недостатки:

- бизнес-логика функциональной обработки и представление данных могут быть одинаковыми для нескольких клиентских приложений, и это увеличит совокупные потребности в ресурсах при исполнении — повторение части кода программ и запросов;
- низкий уровень управления непротиворечивостью информации, так как бизнес-правила функциональной обработки, сосредоточенные на клиентской части, могут быть противоречивыми.

Данная технология позволяет снизить сетевой трафик и повысить общую эффективность обработки за счет оптимизации и буферизации ввода-вывода: сервер может осуществить поиск и обрабатывать запросы даже быстрее, чем если бы они обрабатывались на рабочей станции.

Архитектура «активный сервер баз данных»

Для того чтобы устранить недостатки, свойственные архитектуре сервера базы данных, необходимо, чтобы непротиворечивость бизнес-логики и изменения базы данных контролировались на стороне сервера. Причем некоторые, заранее специфицированные состояния могли бы изменять последовательность взаимодействия приложения с базой данных.

Для этого функции бизнес-логики разделяются между клиентской и серверной частями. Общие или критически значимые функции оформляются в виде *хранимых процедур*, включаемых в состав базы данных. Кроме этого, вводится механизм отслеживания событий БД — *триггеров*, также включаемых в состав базы. При возникновении соответствующего события (обычно изменения данных) СУБД вызывает для выполнения хранимую процедуру, связанную с триггером, что позволяет эффективно контролировать изменение базы данных.

Хранимые процедуры и триггеры могут быть использованы любыми клиентскими приложениями, работающими с базой данных. Это снижает дублирование программных кодов и исключает необходимость компиляции каждого запроса (рис. 8.4).

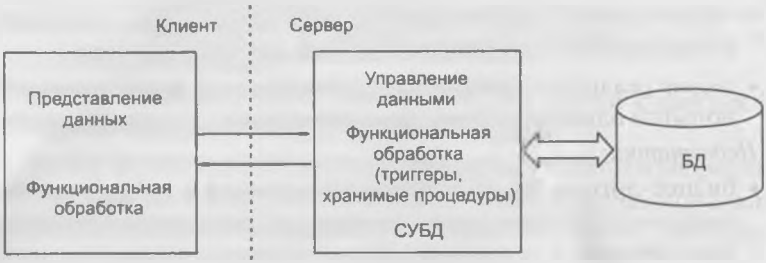


Рис. 8.4. Архитектура «активный сервер баз данных»

Недостатком такой архитектуры становится существенно возрастающая загрузка сервера за счет необходимости отслеживания событий и выполнения части бизнес-правил.

Такую архитектуру организации взаимодействия (а также рассматриваемый далее сервер приложений) иногда называют *моделью с «тонким клиентом»*, в отличие от предыдущих архитектур, называемых *моделью с «толстым клиентом»*, где на стороне клиента выполняется большинство функций.

Архитектура «сервер приложений»

Рассмотренные выше архитектуры являются *двухзвенными*: здесь все функции доступа и обработки распределены между программой клиента и сервером БД.

Дальнейшее снижение требований к ресурсам клиента достигается за счет введения промежуточного звена — *сервера приложений*, на который переносится значительная часть программных компонентов управления данными и большая часть бизнес-логики. При этом серверы баз данных обеспечивают исключительно функции СУБД по ведению и обслуживанию базы данных. Схема трехзвенной архитектуры сервера приложений приведена на рис. 8.5.

Рис. 8.5. Архитектура сервера приложений

К другим (организационно-технологическим) достоинствам трехзвенной архитектуры можно отнести:

- централизованное ведение бизнес-логики и в случае их изменения отсутствие необходимости их тиражирования в клиентских приложениях;
- отсутствие необходимости устанавливать на клиентских машинах компонент программного обеспечения управления доступом к данным;
- возможность отложенного обновления БД в случае изменения данных, запрошенных с сервера, в автономном режиме. Данные будут обновлены в базе после следующего соединения клиентской программы с сервером приложений.

Архитектура сервера баз данных

Повышение эффективности и оперативности обслуживания большого числа клиентских запросов, помимо простого увеличения ресурсов и вычислительной мощности серверной машины, может быть достигнуто двумя путями:

- снижением суммарного расхода памяти и вычислительных ресурсов за счет буферизации (кэширования) и совместного использования наиболее часто запрашиваемых данных и процедур (разделяемые ресурсы);
- распараллеливанием процесса обработки запроса — использованием разных процессоров для параллельной обработки изолированных подзапросов и/или для одновременного обращения к частям базы данных, размещенным на отдельных физических носителях.

Рассмотрим архитектуры, реализующие следующие модели совместной обработки клиентских запросов.

Архитектура «один к одному». В этом случае (рис. 8.6) для обслуживания каждого запроса запускается отдельный серверный процесс.

Таким образом, даже если от клиентов поступят совершенно одинаковые запросы, для обработки каждого из них будет запущен отдельный процесс, каждый из которых будет выполнять одинаковые для всех запросов действия и использовать одни и те же ресурсы.

Многопотоковая односерверная архитектура. Обработку всех клиентских запросов выполняет один серверный процесс (использующий один процессор), взаимодействующий со всеми клиентами и монопольно управляющий ресурсами (рис. 8.7). При этом для отдельно-

Рис. 8.6. Архитектура сервера «один к одному»

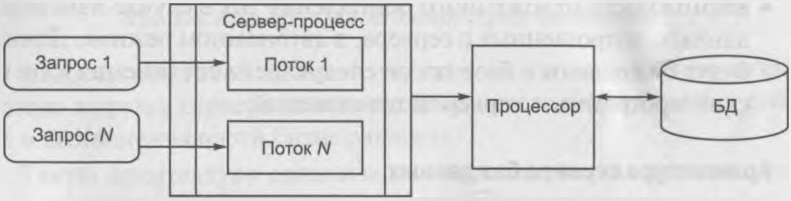


Рис. 8.7. Многопотоковая односерверная архитектура

го клиентского процесса создается поток (thread), в рамках которого локализуется обработка запроса.

Мультисерверная архитектура. В том случае, когда для работы СУБД используются многопроцессорные платформы, обслуживание запросов может быть физически распределено для параллельной обработки между процессорами. Такое решение (рис. 8.8) требует введения дополнительного звена, в задачи которого входит диспетчеризация запросов для обеспечения сбалансированной загрузки процессоров.

Если серверный процесс реализуется как многопоточное приложение, говорят, что СУБД имеет *мультисерверную многопотоковую архитектуру*.

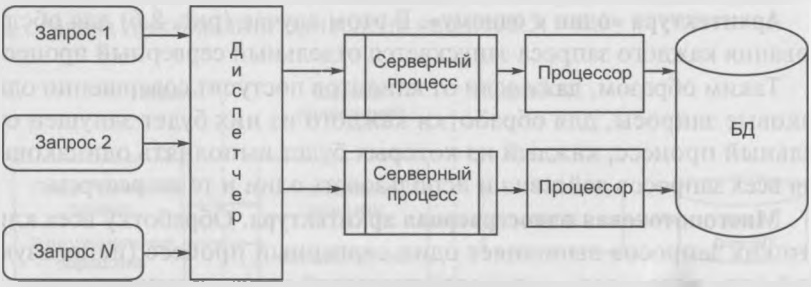


Рис. 8.8. Многопотоковая односерверная архитектура

Следует отметить, что характер распределения запросов в значительной степени зависит от того, поддерживает ли операционная система потоковую обработку, а также от возможностей средств управления приоритетами задач.

Серверные архитектуры с параллельной обработкой запроса. Для повышения оперативности за счет распараллеливания процесса обработки отдельного клиентского запроса в мультисерверной архитектуре можно использовать следующие подходы:

1) размещение хранимых данных БД на нескольких физических носителях (сегментирование базы). Для обработки запроса в этом случае запускаются несколько серверных процессов (использующих обычно отдельные процессоры), каждый из которых независимо от других выполняет одинаковую последовательность действий, определяемую существом запроса, но с данными, принадлежащими разным сегментам базы. Полученные таким образом результаты объединяются и передаются клиенту. Такой тип распараллеливания называют *моделью горизонтального параллелизма*;

2) запрос обрабатывается по конвейерной технологии. Для этого запрос разбивается на взаимосвязанные по результатам подзапросы, каждый из которых может быть обслужен отдельным серверным процессом независимо от обработки других подзапросов. Получаемые результаты объединяются согласно схеме декомпозиции запроса и передаются клиенту. Такой тип распараллеливания называют *моделью вертикального параллелизма*.

Примерная схема обработки клиентского запроса, построенная с использованием обеих моделей параллелизма (гибридная модель), приведена на рис. 8.9.

Использование моделей параллельной обработки позволяет существенно сократить общее время обслуживания запроса, что особенно важно в случае работы с большими базами данных и аналитической обработки (OLAP-приложений).

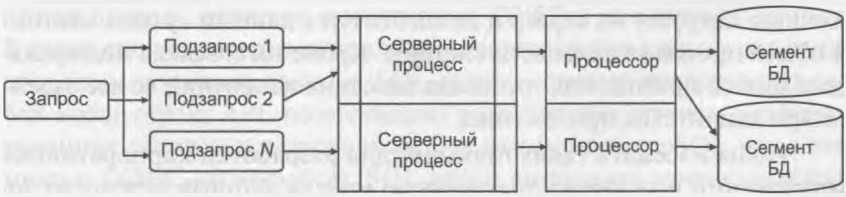


Рис. 8.9. Архитектура сервера обработки запроса при гибридном параллелизме

8.2.4. Технологии и средства доступа к удаленным БД

Программное обеспечение распределенных приложений

Распределенные корпоративные приложения все более усложняются, интегрируя в себя унаследованные приложения, разрабатываемые и вновь приобретаемые готовые программные средства. Кроме того, разные подсистемы решают разные бизнес-задачи, однако одна из главных целей создания корпоративной системы — получить «единый образ» общего состояния системы, что обеспечит пользователям доступ к нужным операциям и ресурсам.

Основа такой инфраструктуры — так называемое *промежуточное программное обеспечение*, позволяющее, не вникая в тонкости сетевых реализаций, создавать и эксплуатировать взаимодействующие между собой приложения с разными требованиями к межмодульным коммуникациям.

Промежуточное ПО эволюционировало вместе с архитектурой клиент—сервер. Ранние, но достаточно эффективные как с точки зрения разработки, так и в эксплуатации, частные решения предназначались для упрощения доступа к базам данных в двухзвенной модели, где «толстый» клиент реализует всю логику обработки информации, предоставляемой сервером базы данных. Такие системы вполне удовлетворяли потребностям небольших корпоративных подразделений с ограниченным числом пользователей и невысокой интенсивностью обмена.

Однако по мере того, как клиент-серверная архитектура стала проникать в сферу высококритичных корпоративных приложений, обслуживающих уже не десятки, а сотни пользователей и работающих со значительными массивами данных, стали очевидны недостатки двухзвенного подхода. Этот способ реализации клиент-серверной схемы доступа ограничивал возможности масштабирования, поскольку рост числа обращений к одной базе данных непомерно увеличивал нагрузку на сервер и делал доступ к данным «узким местом» в общей производительности системы. Кроме того, всякая модификация логики приложения требовала внесения изменений во все экземпляры клиентских приложений.

Чтобы избежать таких проблем, для разработки корпоративных приложений используют трехзвенную модель, которая переносит логику приложения на отдельный уровень сервера приложений. В результате клиентская часть приложения становится «тоньше» и в ос-

новном отвечает за предоставление удобного пользовательского интерфейса. Как правило, сервер баз данных также освобождается от необходимости поддерживать бизнес-логику, которая в двухзвенной модели реализуется с помощью специальных расширений СУБД, например, хранимых процедур.

Таким образом, в распределенной неоднородной среде программное обеспечение промежуточного уровня играет роль «информационной шины», надстроенной над сетевым уровнем и обеспечивающей доступ приложения к разнородным ресурсам, а также независимую от платформ взаимосвязь различных прикладных компонентов, изолирующую логику приложений от уровня сетевого взаимодействия и ОС.

ПО промежуточного уровня можно разделить на две категории:

1) ПО доступа к базам данных (например, ODBC-интерфейсы и SQL-шлюзы);

2) ПО межмодульного взаимодействия — системы, реализующие вызов удаленных процедур (RPC — Remote Procedure Call); мониторы обработки транзакций (TP-мониторы); средства интеграции распределенных объектов.

При этом следует отметить, что различия прикладных задач не позволяют построить универсальное ПО, реализовав в одном продукте все необходимые возможности.

Доступ к базам данных в двухзвенных моделях «клиент—сервер»

В двухзвенных моделях клиент–сервер, где несколько баз данных обслуживают ограниченное число пользователей, в роли встроенного ПО доступа к данным могут выступать обычные OLE- или ODBC-драйверы.

Необходимость в более сложных решениях возникает в больших, разнородных многозвенных системах, где множество приложений в параллельном режиме осуществляет доступ к разнообразным источникам данных, включая разнотипные СУБД и хранилища данных. В таких системах между клиентами и серверами баз данных размещается промежуточное звено — SQL-шлюз, который представляет собой набор общих API, позволяющих разработчику строить унифицированные запросы к разнородным данным (в формате SQL или с помощью ODBC-интерфейса). SQL-шлюз выполняет синтаксический разбор такого запроса, анализирует и оптимизирует его и в конце концов выполняет преобразование в SQL-диалект нужной СУБД. ПО

этого типа реализует синхронный механизм связи, когда выполнение приложения, сделавшего запрос, блокируется до момента получения данных.

Создается такое приложение обычно с использованием средств языков высокого уровня (например, C++, Pascal, Visual Basic), позволяющих реализовать эффективную целевую обработку данных и дружественный пользовательский интерфейс. В исходный текст программы включаются SQL-выражения, специфицирующие условия выборки или изменения данных в базе. Во время исполнения приложения эти выражения передаются серверу, который собственно и манипулирует данными. Данные, полученные в результате выполнения сервером SQL-запросов, возвращаются прикладной программе и размещаются в заранее определенных структурах для дальнейшей обработки, в том числе корректировки записей.

Рассмотрим различные способы организации доступа прикладной программы к серверу базы данных в двухзвенной архитектуре.

Использование библиотек доступа и встраиваемого SQL

Каждая СУБД помимо интерактивной SQL-утилиты обязательно имеет библиотеку процедур доступа и набор драйверов СУБД для различных операционных систем. Схема взаимодействия клиентского прило- ения с сервером базы данных в этом случае представлена на рис. 8.10.

Библиотека доступа содержит набор функций, позволяющих клиентскому приложению соединяться с базой данных, передавать запросы серверу и получать данные — результаты обработки запроса. Типичный набор функций такой библиотеки включает:

- соединение с базой данной;
- запрос на добавление данных;
- запрос на извлечение данных;
- запрос на изменение данных;
- закрытие соединения с базой данных.

Обычно в библиотеке присутствуют также функции, позволяющие определить характеристики структуры набора результата (число, порядок и имена столбцов, число строк, номер текущей строки), передвигаться по этой структуре не только вперед, но и назад и т. д.

Библиотечные вызовы преобразуются драйвером базы данных в сетевые вызовы и передаются сетевым программным обеспечением на сервер. На сервере происходит обратный процесс преобразования

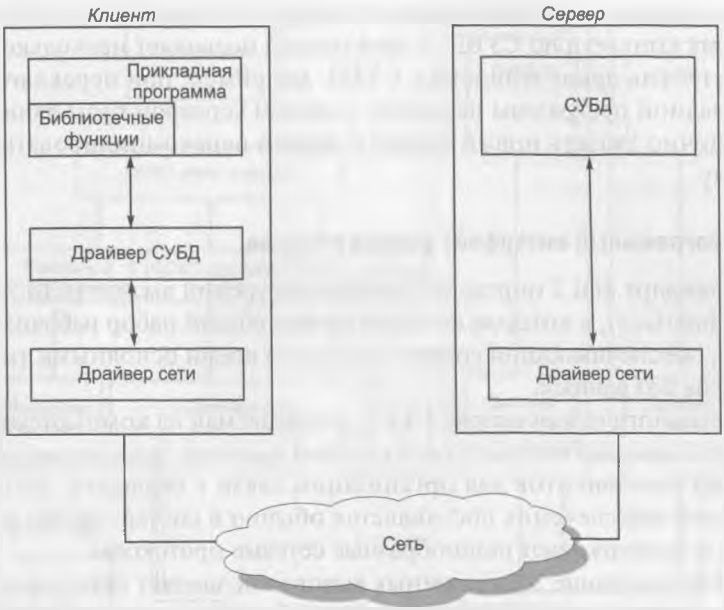


Рис. 8.10. Схема взаимодействия с использованием библиотек процедур доступа

сетевых пакетов в SQL-запросы, которые обрабатываются СУБД. Результаты обработки передаются клиенту.

Такой способ создания приложений достаточно гибок и позволяет реализовать практически любое приложение, однако имеет и недостатки:

- разработка клиентской программы возможна только для той операционной системы и на том языке программирования, в которых поддерживается библиотека;
- драйвер базы данных определяет допустимые типы сетевых интерфейсов;
- библиотечные функции обычно не унифицированы.

Некоторой модификацией данного способа является использование «встроенного» языка SQL. В этом случае текст программы на языке третьего поколения вместо вызовов функций библиотеки включает непосредственно предложения SQL, которые предваряются выражением «EXEC SQL». Перед компиляцией в машинный код такая программа обрабатывается препроцессором, который транслирует смесь операторов «собственного» языка СУБД и SQL-предложений в промежуточный «чистый» исходный код, а затем коды SQL замеща-

ются вызовами соответствующих процедур из библиотек, поддерживающих конкретную СУБД. Такой подход позволяет несколько снизить степень привязанности к СУБД, например, при переключении прикладной программы на работу с другим сервером базы данных — достаточно указать новый сервер и заново перекомпилировать программу.

Программный интерфейс уровня вызовов

Стандарт SQL2 определил интерфейс уровня вызова (CLI — Call Level Interface), в котором стандартизован общий набор рабочих процедур, обеспечивающий совместимость со всеми основными типами серверов баз данных.

Технологическая основа CLI — размещаемая на компьютере клиента специальная библиотека, в которой хранятся вызовы процедур и сетевых компонентов для организации связи с сервером. Это программное обеспечение поставляется обычно в составе среды разработки и поддерживает разнообразные сетевые протоколы.

Использование программных вызовов позволяет свести к минимуму операции на компьютере-клиенте. В общем случае клиент формирует оператор языка SQL в виде строки и пересылает ее на сервер посредством процедуры исполнения (execute). Когда же сервер в качестве ответа возвращает несколько строк данных, клиент считывает результат последовательным вызовом процедуры выборки данных. Далее данные из столбцов полученной таблицы могут быть связаны с соответствующими переменными приложения. Вызов специальной процедуры позволяет клиенту определить число полученых строк, столбцов и типы данных в каждом столбце.

Открытый интерфейс доступа к базам данных

Спецификация открытого интерфейса баз данных (ODBC — Open Database Connectivity) предназначена для унификации доступа к данным, размещенным на удаленных серверах. ODBC опирается на спецификации CLI.

ODBC представляет собой программный слой, унифицирующий интерфейс взаимодействия приложений с базами данных. За реализацию особенностей доступа к каждой отдельной СУБД отвечает соответствующий специальный ODBC-драйвер (рис. 8.11). Пользовательское приложение этих особенностей не видит, так как взаимодействует с универсальным программным слоем более высокого уровня.

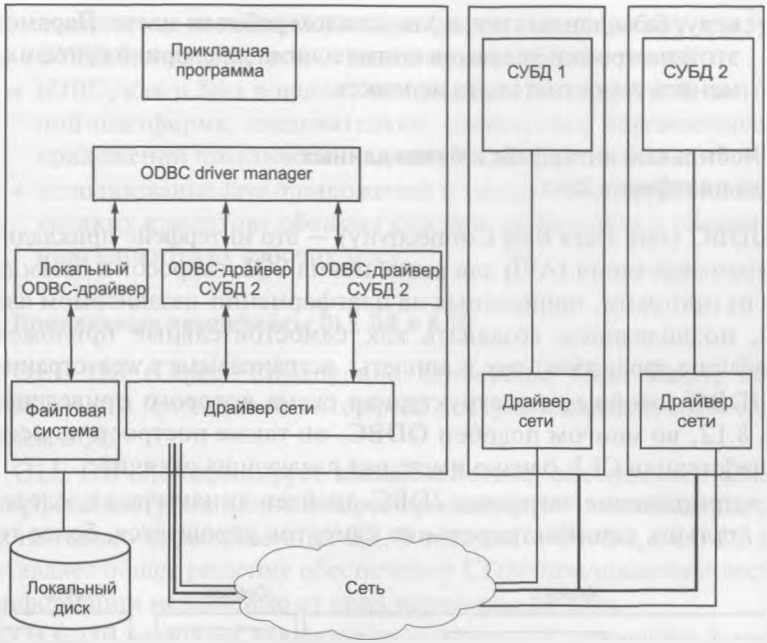


Рис. 8.11. Структурная схема доступа к данным с использованием ODBC

Таким образом, приложение становится в значительной степени независимым от СУБД. Вместо создания в каждом отдельном случае СУБД-приложения с обращениями через «родной», но быстро устаревающий интерфейс, можно использовать один общий стандартизированный программный интерфейс.

В архитектуре ODBC используется один ODBC Driver Manager и несколько ODBC-драйверов, обеспечивающих доступ к конкретным СУБД. Driver Manager связывает приложение и интерфейсные объекты, которые выполняют обработку SQL-запросов к конкретной СУБД.

Такой подход является достаточно универсальным, стандартизируемым, что и позволяет использовать ODBC-механизмы для работы практически с любой системой.

Однако этот способ также не лишен недостатков:

- увеличивается время обработки запросов (как следствие введения дополнительного программного слоя);
- необходимы предварительная инсталляция и настройка ODBC-драйвера (указание драйвера СУБД, сетевого пути к сер-

веру, базы данных и т. д.) на каждом рабочем месте. Параметры этой настройки являются статическими, т. е. приложение их изменить самостоятельно не может.

Мобильный интерфейс к базам данных на платформе Java

JDBC (Java Data Base Connectivity) — это интерфейс прикладного программирования (API) для выполнения SQL-запросов к базам данных из программ, написанных на платформенно-независимом языке Java, позволяющем создавать как самостоятельные приложения (standalone application), так и апплеты, встраиваемые в web-страницы.

JDBC, обобщенная структурная схема которого приведена на рис. 8.12, во многом подобен ODBC, он также построен на основе спецификации CLI, однако имеет ряд следующих отличий:

- приложение загружает JDBC-драйвер динамически, следовательно, администрирование клиентов упрощается, более того,

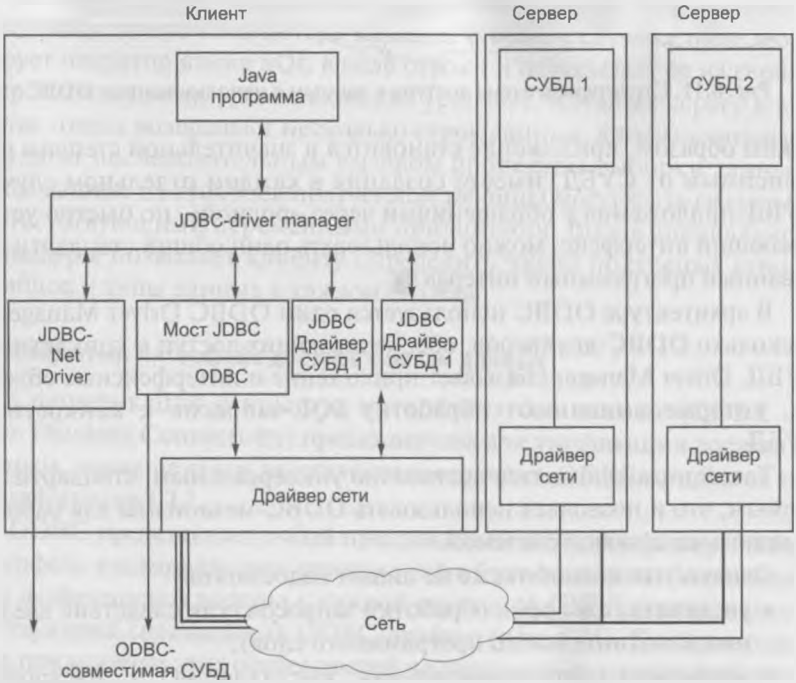


Рис. 8.12. Структурная схема доступа к данным с использованием JDBC

появляется возможность переключаться на работу с другой СУБД без перенастройки клиентского рабочего места;
- JDBC, как и Java в целом, не привязан к конкретной аппаратной платформе, следовательно, проблемы с переносимостью приложений практически снимаются;
- использование Java-приложений и связанной с ними идеологии «тонких клиентов» обещает снизить требования к оборудованию клиентских рабочих мест.

Прикладные интерфейсы OLE DB и ADO

OLE DB (Object Linking and Embedding Data Base), как и ODBC, — это прикладной интерфейс доступа к данным с использованием SQL.

OLE DB специфицирует взаимодействие, обеспечивая единый интерфейс доступа к данным через провайдеров — поставщиков данных не только из реляционных БД. В отличие от ODBC, OLE DB предоставляет общее решение обеспечения COM-приложениям доступа к информации независимо от типа источника данных.

OLE DB включает два базовых компонента: *провайдер данных* и *потребитель данных*. Потребитель (клиент) — это приложение или COM-компонент, обращающийся посредством API-вызовов к OLE DB. Провайдер (сервер) — это приложение, отвечающее на вызовы OLE DB и возвращающее запрашиваемый объект (обычно это данные в табличном виде).

ADO (Active Data Object) — это универсальный интерфейс высокого уровня к OLE DB. Модель объекта ADO не содержит таблиц, среды или машины БД. Здесь основными объектами являются следующие: объект *Соединение*, создающий связь с провайдером данных; объект *Набор данных* и объект *Команда* — выполнение процедуры или SQL-строки.

В общем случае ADO можно рассматривать как язык программирования операций с БД, позволяющий выбирать, модифицировать и удалять записи. И поскольку он опирается на универсальный OLE DB, то может использоваться практически в любых приложениях Microsoft.

Рассмотренные технологии построения приложения ориентированы на извлечение данных непосредственно из статического источника (хранилища данных) и не могут обращаться за данными к другому прикладному модулю.

8.2.5. Корпоративные серверы приложений

Появление серверов приложений как отдельных готовых решений связано и с бурным вторжением Web-технологий в сферу корпоративных высококритичных систем. Однако возможности протокола HTTP ограничены функциями связи без каких-либо средств сохранения информации о состоянии, поэтому он не подходит для поддержки мощных корпоративных систем.

На рис. 8.13 приведен «идеальный» состав сервера приложений с максимальным набором необходимых служб и средств связи с клиентскими системами и информационными ресурсами.

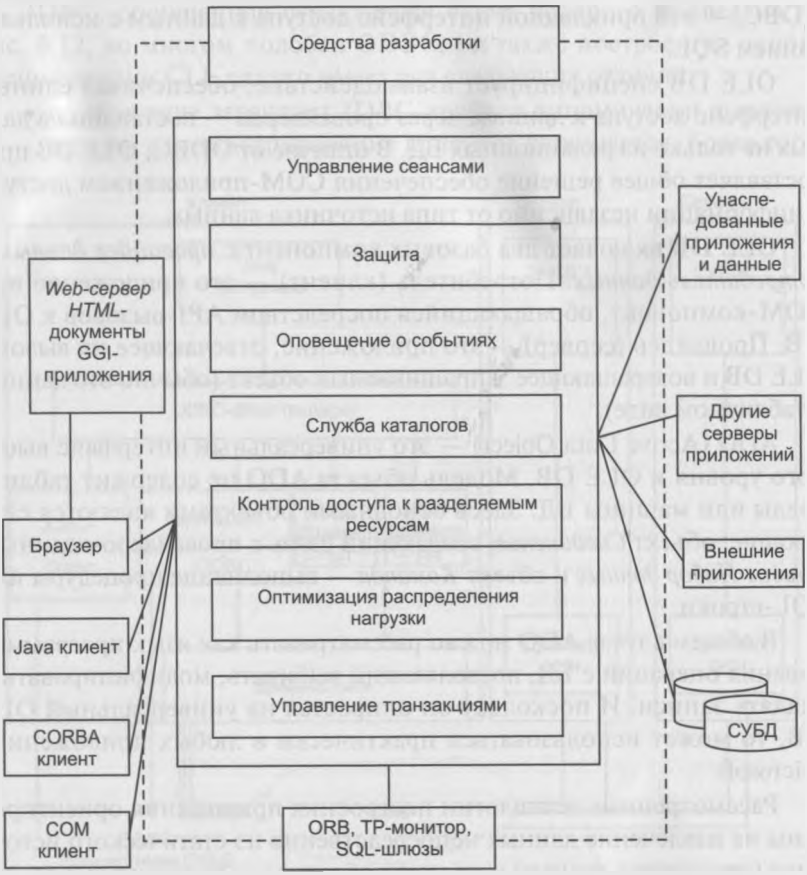


Рис. 8.13. Обобщенная структура сервера приложений

Сегодня прикладные разработки базируются на одной из двух компонентных моделей — MTS/DCOM и CORBA, способных интегрировать объекты на удаленных платформах.

Обе модели распространяют принципы вызова удаленных процедур на объектные распределенные приложения и обеспечивают прозрачность реализации и физического размещения серверного объекта для клиентской части приложения; поддерживают возможность взаимодействия объектов, созданных на различных объектно-ориентированных языках, и скрывают от приложения детали сетевого взаимодействия.

В DCOM взаимодействие удаленных объектов, представленное на рис. 8.14, базируется на спецификации DCE RPC, а CORBA использует брокер объектных запросов (ORB), синхронный механизм которого во многом схож с RPC.

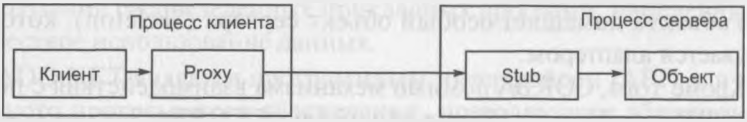


Рис. 8.14. DCOM-технология взаимодействия «клиент—сервер»

В DCOM-технологии взаимодействие между клиентом и сервером осуществляется через двух посредников. Клиент помещает параметры вызова в стек и обращается к методу интерфейса объекта. Это обращение перехватывает посредник Proxy, упаковывает параметры вызова в COM-пакет и адресует его в Stub, который в свою очередь распаковывает параметры в стек и инициирует выполнение метода объекта в пространстве сервера.

CORBA-технология также использует интерфейс объекта, но в этом случае схема взаимодействия объектов (рис. 8.15) включает промежуточное звено (*Smart agent*), реализующее доступ к удаленным объектам. Smart agent, установленный на машинах сетевого окружения (сервере локальной сети или Internet-узле), моделирует сетевой

Рис. 8.15. CORBA-технология взаимодействия «клиент—сервер»

каталог известных ему серверов объектов. При создании сервера происходит автоматическая регистрация его объектов в каталоге одного или нескольких Smart agent.

Связи между брокерами осуществляются в соответствии с требованиями специального протокола General Inter ORB Protocol, определяющего низкоуровневое представление данных и множество форматов сообщений.

На машине клиента создаются два объекта-посредника: Stab и ORB (Object Required Broker — брокер вызываемого объекта). Так же, как и в DCOM-технологии, Stub передает перехваченный вызов брокеру, который посылает широковещательное сообщение в сеть. Smart agent, получив сообщение, отыскивает сетевой адрес сервера и передает запрос брокеру, размещенному на машине сервера. Вызов требуемого объекта производится через специальный базовый объектный адаптер (BOA). При этом данные в стек пространства вызываемого объекта помещает особый объект сервера (Skeleton), который вызывается адаптером.

Кроме того, CORBA помимо механизма взаимодействия с помощью ORB включает в себя ряд общих служб CORBA Services (служба каталогов, защиты, оповещения о событиях, поддержки транзакций и ряд других), а также реализаций объектов для разных прикладных областей.

Ключевым компонентом архитектуры CORBA является язык описания интерфейсов IDL, на уровне которого поддерживаются «контрактные» отношения между клиентом и сервером и обеспечивается независимость от конкретного объектно-ориентированного языка. CORBA IDL поддерживает основные понятия объектно-ориентированной парадигмы (инкапсуляцию, полиморфизм и наследование).

В модели DCOM также может использоваться разработанный Microsoft язык IDL, который, однако, играет вспомогательную роль и используется в основном для удобства описания объектов. Реальная интеграция объектов в DCOM происходит не на уровне абстрактных интерфейсов, а на уровне бинарных кодов, и это одно из основных различий этих двух объектных моделей.

И DCOM, и CORBA дают возможность динамического связывания удаленных объектов: клиент может обратиться к серверу-объекту во время выполнения, не имея информации об этом объекте на этапе компиляции. В CORBA для этого существует специальный интерфейс динамического вызова DII, а COM использует механизм OLE-Automation. Информацию о доступных объектах сервера на эта-

пе выполнения клиентская часть программы получает из специального хранилища метаданных об объектах — репозитария интерфейсов Interface Repositary в случае CORBA или библиотеки типов (Type Library) в модели DCOM. Эта возможность очень важна для больших распределенных приложений, поскольку позволяет менять и расширять функциональность серверов, не внося существенных изменений в код клиентских компонентов программы. Например, банковское приложение, основная бизнес-логика которого поддерживается сервером в центральном офисе, а клиентские системы разбросаны по филиалам в разных городах.

Доступ к данным с помощью ADO.NET

ADO.NET является преемником Microsoft ActiveX Data Objects (ADO). Это W3C-стандартизированная модель программирования для создания распределенных прикладных программ, нацеленных на совместное использование данных.

ADO.NET является программным интерфейсом (API) для прикладного программного обеспечения, позволяющим обращаться к данным и другой информации. ADO.NET поддерживает такие современные требования, как создание клиентского интерфейса к базам данных на фронтальном уровне и на уровне промежуточного слоя объектов клиентских приложений, инструментальных средств, языков программирования или Internet-браузера.

ADO.NET, подобно ADO, обеспечивает интерфейс доступа к OLE DB-совместимым источникам данных. Прикладные программы, позволяющие пользователям совместно использовать данные, могут использовать ADO.NET для подключения к источникам данных, а также для поиска, и модификации этих данных. Прикладные программы также могут использовать OLE DB для управления данными, хранящимися в форматах, отличных от форматов БД.

В решениях, требующих автономного или удаленного доступа к данным, ADO.NET использует XML для обмена данными между программами или с Web страницами. Любой компонент, который обслуживает XML, также может использовать и компоненты ADO.NET. Если передача пакетов компонентом ADO.NET подразумевает поставку набора данных в файле XML, то компонентом, способным обеспечить его получение, может быть только компонент ADO.NET. Передача данных в XML-формате дает возможность легко отделить обработку данных от компонент пользовательского интерфейса.

Для распределенных приложений использование наборов данных XML в ADO.NET обеспечивает лучшую эффективность, чем использование COM для офлайнового обслуживания данных в ADO. Поскольку передача наборов данных происходит через файлы XML, описанные в достаточно простом стандартном языке и являющиеся обычными текстовыми файлами, компоненты ADO.NET не имеют архитектурных ограничений, свойственных COM. Фактически, любые два компонента могут совместно использовать наборы XML-данных при условии, что они оба используют ту же самую XML-схему форматирования.

ADO.NET обладает масштабируемостью, что удобно для совместно использующих данные Web-приложений. Кроме того, ADO.NET не использует длительные блокировки баз данных и активные подключения, которые на долгое время монополизируют ресурсы сервера, являющиеся, как правило, весьма ограниченными. Это позволяет увеличивать число пользователей без значительного увеличения загрузки ресурсов системы.

8.2.6. Схемы размещения и доступа к данным в распределенных БД

Размещение данных в распределенных БД характеризуется следующими понятиями.

1. Фрагментация. Любая логическая единица БД (например, отношение в случае реляционных моделей данных) может быть разделена на некоторое количество частей, называемых фрагментами, которые затем могут распределяться по различным узлам. Существуют два основных типа фрагментации: горизонтальная и вертикальная. В первом случае фрагменты представляют собой подмножества записей, а во втором — подмножества атрибутов.

2. Размещение. Каждый фрагмент сохраняется на узле, выбранном с учетом оптимальной схемы доступа.

3. Репликация. Распределенная СУБД может поддерживать актуальную копию некоторого фрагмента на нескольких различных узлах.

Определение и размещение фрагментов должно проводиться с учетом особенностей использования базы данных (в частности, на основе анализа транзакций). Существуют четыре стратегии размещения данных в системе.

1. Централизованное размещение. Данная стратегия предусматривает создание на одном из узлов единственной базы данных под

управлением СУБД, доступ к которой будут иметь все пользователи сети.

2. Фрагментированное размещение. В этом случае база данных разбивается на непересекающиеся фрагменты, каждый из которых размещается на одном из узлов системы.

3. Размещение с полной репликацией. Эта стратегия предусматривает размещение полной копии всей базы данных на каждом из узлов системы. Снимок представляет собой копию базы данных в определенный момент времени. Эти копии обновляются через некоторый установленный интервал времени, например, один раз в час или в сутки.

4. Размещение с избирательной репликацией. Данная стратегия представляет собой комбинацию методов фрагментации, репликации и централизации. Одни массивы данных разделяются на фрагменты, что позволяет добиться для них высокой локализации ссылок, тогда как другие, используемые на многих узлах, но не подверженные частым обновлениям, подвергаются репликации. Все остальные данные хранятся централизованно.

Управление параллельной обработкой в распределенных БД

При *параллельной обработке данных* (совместной работе нескольких пользователей с общими данными) СУБД должна гарантировать, что пользователи не будут мешать друг другу (сюда относятся, например, проблемы потерянного обновления, зависимости от промежуточных результатов, несогласованности обработки, несогласованности многих копий данных) и их действия будут изолированы. Такими единицами изолированности являются транзакции — неделимые (с точки зрения воздействия на состояние целостности БД) последовательности операторов управления данными.

Решения по организации управления параллельным выполнением в распределенной среде основаны на подходах с использованием механизмов *блокировок и временных отметок*.

Механизм блокировки обеспечивает эквивалентность графика параллельного выполнения транзакций некоторому (но, в общем случае, непредсказуемому) варианту последовательного выполнения этих транзакций.

Механизм обработки временных отметок гарантирует, что график параллельного выполнения транзакций будет эквивалентен конкретному варианту последовательного выполнения этих транзакций в соответствии с их временными отметками.

Многоуровневая модель предметной области для распределенных БД

Архитектурная модель ANSI/SPARC (см. главу 5), представляющая собой классическое решение для локальных СУБД, может быть расширена на случай распределенных СУБД. На рис. 8.16 приведена обобщенная схема архитектуры распределенной БД, где в верхней части присутствуют схемы глобального уровня, обеспечивающие целостное представление БД, как если бы она была локальной.

Здесь на внешнем уровне может поддерживаться совсем иная модель данных (или даже несколько моделей), чем на концептуальном уровне. Поддержка разнообразных возможностей абстрагирования в такой системе достигается благодаря средствам определения и поддержки межуровневого отображения моделей данных. Схемы фрагментации и распределения определяют размещение логических сегментов данных по физическим разделам (локальным БД), а локальная схема преобразования обеспечивает отображение фрагментов во внешние схемы. При этом локальные концептуальные и внутренние схемы определяют в соответствии с концепцией представление данных с учетом особенностей конкретной СУБД.

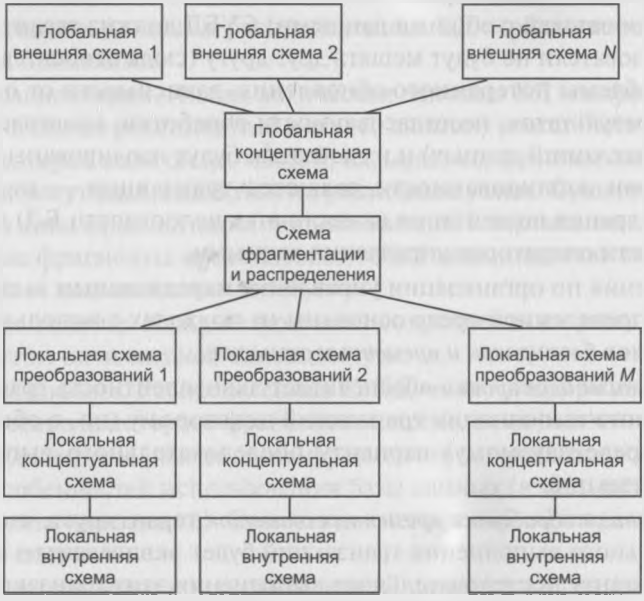


Рис. 8.16. Обобщенная схема распределенной БД

8.3. Распределенные документальные информационные ресурсы

8.3.1. Типологии распределенных информационных ресурсов

Рассматривая в главе 7 технологии и средства информационного поиска, мы не акцентировали внимание на организационно-физических аспектах ИР и доступа к ним (показывая тем самым, что принципы информационного поиска и архитектура ИС имеют универсальный характер).

Реализации АИПС по своим возможностям различаются достаточно существенно. Это связано не столько с решениями тех или иных разработчиков, сколько с многочисленными факторами, в той или иной степени ограничивающими или даже исключающими некоторые функциональные возможности. Такими факторами являются, например, назначение и область применения, масштабность и интенсивность использования ИР, характер распределения информационных компонентов (для сетевых систем), тип информации, источники и режим пополнения, предполагаемый профессиональный уровень потребителей и т. д.

По архитектуре, определяющей физическую доступность ресурсов, информационно-поисковые системы (в том числе и Internet-машины) могут быть разделены на следующие классы:

- локальные системы — локализованные данные и их обработка;
- частично-распределенные — локальная обработка распределенных данных;
- полностью распределенные системы, где реализуются принципы распределенных вычислений и распределенного хранения данных.

Локальные системы обеспечивают доступ удаленных пользователей к ресурсам, сосредоточенным на поисковом сервере. Эти системы в большинстве случаев функционально эквивалентны локальным системам, например, на CD-ROM-носителях.

Ко второму типу относятся системы, использующие данные, находящиеся на Web-серверах в качестве распределенных *первичных* ИР; вторичные и индексные данные сосредоточены на поисковом сервере, осуществляющем обслуживание пользователей. Это такие системы, как AltraVista, Google, Yandex и пр.

К третьему типу относятся системы, в которых процесс поиска реализуется на совокупности серверов, распределенных по сети, ко-

торые при обработке запроса опрашивают друг друга, причем исходные и промежуточные данные поиска также имеют распределенный характер.

По тематическому и видовому спектрам ИР могут быть однородными (иметь четко выраженную тематику и работать с документами определенного типа и состава) и гетерогенными (политематическими и не имеющими требований к составу и форме документов).

По способу формирования ИР подразделяются на те, которые используют предопределенные источники, например, публикации издательств, рецензирующих материалы, и те, которые используют все свободно доступные материалы. Примерами здесь, соответственно, являются базы данных научной информации и поисковые машины Internet, индексирующие открытые HTML-страницы.

Одним из наиболее значимых примеров являются электронные библиотеки (ЭБ), где в качестве компонентов выступают электронные каталоги (библиографические и реферативные базы данных), полнотекстовые массивы (электронные журналы, фактографические базы данных, хранилища электронных копий источников в том или ином виде и т. д.), справочно-нормативные файлы (рубрикаторы, тезаурусы, авторские, предметные, географические и другие указатели), возможно, связанные между собой ссылками, указателями хранения или условиями поиска.

С точки зрения характера и формы представления информации (и, соответственно, логики организации поиска) архитектура ИР включает три уровня: уровень собственно документов (полных текстов), уровень поисковых образов и метаинформационный уровень. Характер и логика взаимосвязей информационных элементов отдельных уровней отражены схемой на рис. 8.17, где в скобках даны примеры, характерные для автоматизированных или традиционных информационных систем.

Взаимосвязь между компонентами разных уровней может быть реализована как для компонентов в целом, так и для их элементов. Иллюстрацией служит, например, такая связь, как «библиографическая запись электронного каталога — запись полнотекстовой базы данных» или «библиографическая запись — оцифрованная копия источника (изображение)». К другому типу — на уровне элементов — могут относиться такие связи, как «пристатейная ссылка — библиографическая запись» или «фрагмент библиографической записи — запись нормативной базы данных» и т. д. Однако вербально или алгоритмически выразить эту взаимосвязь иногда весьма сложно.

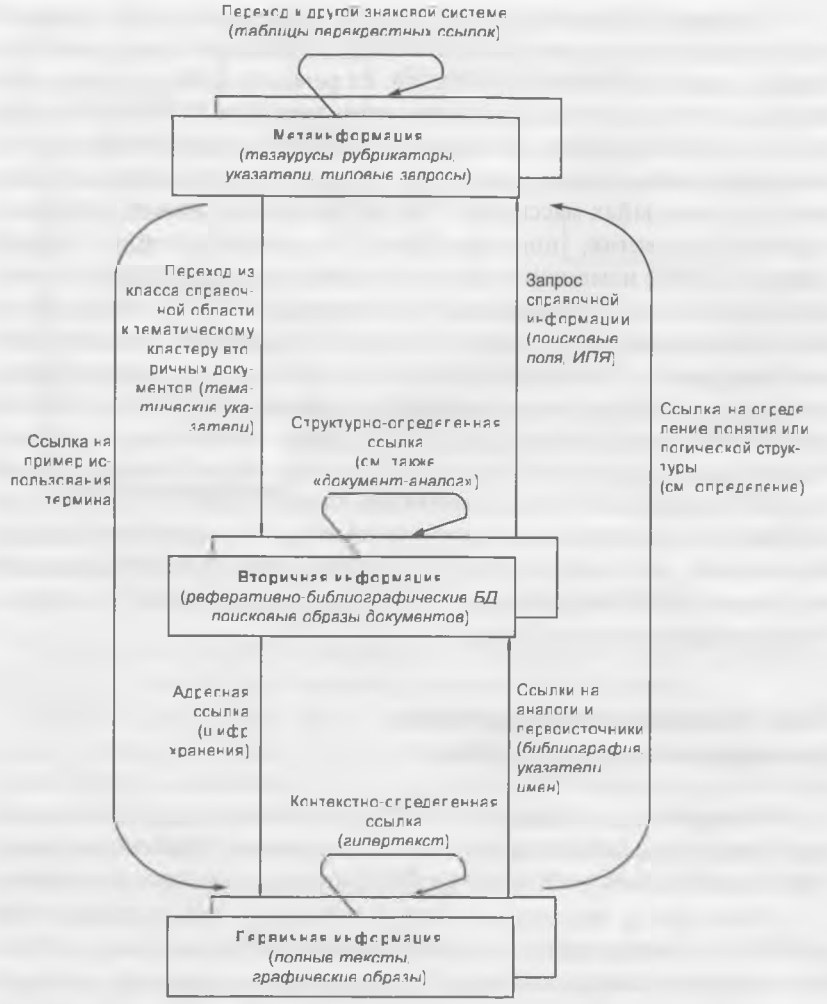


Рис. 8.17. Характер взаимосвязей информационных элементов

Рассматривая эту схему как «технологию» поиска, можно видеть, что ссылки вполне узнаваемы и представляют собой традиционные правила и приемы отыскания информации в условиях «бумажной» библиотеки, когда поиск начинается с классификационной схемы или указателя, далее через библиографические карточки к первоисточнику, где с помощью пристатейных ссылок и указателей снова продолжается с метаинформационного уровня.

С широким внедрением телекоммуникационных сетей и некоторой стандартизации представления данных в Internet задача взаимосвязи становится еще более сложной. Ее решение путем создания статичных связей практически невозможно, даже если бы все компоненты имели свои уникальные идентификаторы и незыблемое место в информационном пространстве (чего зачастую невозможно добиться даже для локальных массивов). Что уж говорить о, скажем, информационных объектах, появляющихся на многочисленных сайтах Internet. Любое изменение местоположения информационного объекта влечет за собой возникновение «ложных» связей в распределенных электронных библиотеках. И число этих связей с течением времени возрастает. Поэтому на смену статичным связям приходят динамические, генерируемые программно во время обращения к объекту. Связи могут быть построены на таких идентификаторах, как давно применяемые ISBN и ISSN или DOI (Digital Object Identifier). В тех случаях, когда такие идентификаторы отсутствуют (а таких случаев большинство), одним из решений может быть генерация динамических связей, где в качестве основы для построения идентификаторов могут выступать уникальные элементы записи, например, элементы библиографического описания.

8.3.2. Распределенная обработка в поисковых машинах Internet

Особенностью современных поисковых систем является фактическая разработка собственных операционных сред. Типичным примером является среда, на базе которой функционируют серверы Google.

Особенность файловой системы Google, представленной на рис. 8.18, — ориентация на очень большие, мало изменяющиеся файлы, которые должны храниться на множестве ненадежных вычислительных систем.

GFS состоит из множества кластеров. Каждый кластер состоит из отдельного Master-компьютера (мастера) и множества подчиненных ему серверов (chunkservers). Все файлы в системе состоят из кусочков фиксированной длины — 64 Мб. Каждый кусочек помечен 64-битовым идентификатором.

GFS — это файловая система с журналом. Master, собственно, и ведет журнал изменений и обращений. Он же и восстанавливает систему при сбоях.

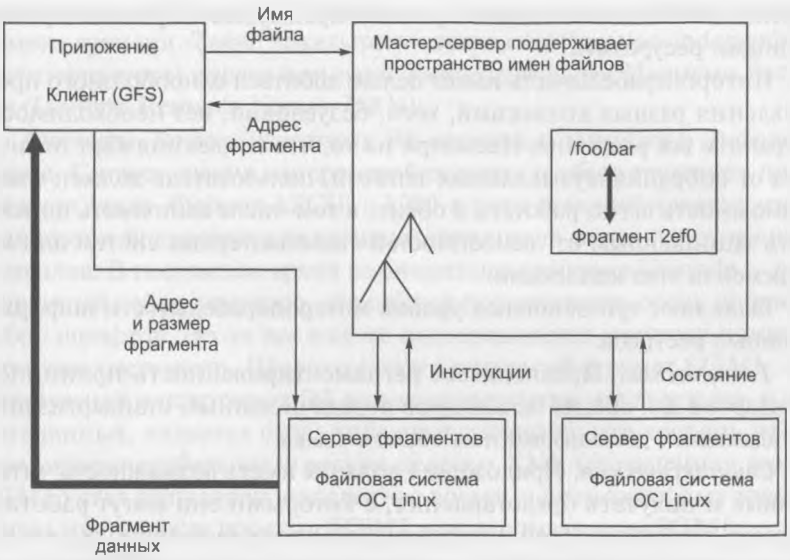


Рис. 8.18. Архитектура файловой системы Google (GFS)

Система постоянно реплицируется. Кроме того, постоянно ведется контроль целостности данных и их непротиворечивости.

В GFS хранятся и индексы, и сами страницы, которые индексируются роботами Google. Собственно, GFS — это только часть технологии Google, предназначенной для работы с собранными данными. Кроме GFS, для поиска и записи данных в Google применяют технологии MapReduce (распараллеливание процедуры поиска и агрегирование результатов поиска) и BigTable (распределенное хранение больших объемов структурированных данных).

Технологии Google предназначены для параллельной обработки запросов. При этом процедуры записи в GFS и поиска данных в GFS симметричны, что дает возможность использовать MapReduce и BigTable как при поиске, так и при записи данных.

8.3.3. Интероперабельность в распределенных ИР

Как уже отмечалось, эффективность использования накопленной человечеством информации в значительной степени сегодня связывается с обеспечением интероперабельности (совместимостью) информационных ресурсов, так как отсутствие унифицированности и стан-

дартизации способно свети к нулю все преимущества работы с разнородными ресурсами.

Интероперабельность имеет целью добиться однообразного представления разных коллекций, хотя, безусловно, нет необходимости устранять все различия. Несмотря на то, что коллекция карт отличается от собрания музыкальных записей, пользователь должен иметь возможность легко работать в обеих, в том числе выполнять поиск и быть защищенным от «особенностей» компьютерных систем или менеджмента этих коллекций.

Выделяют три основных уровня интероперабельности информационных ресурсов.

Техническая. Предполагает регламентированность принципов, стандартов для общих механизмов передачи данных, стандартизацию метаданных с использованием общего языка.

Синтаксическая. Приложения должны иметь возможность читать данные и получать представления, с которыми они могут работать. Уровень синтаксической интероперабельности достаточно высок, когда можно без труда построить синтаксический анализатор и API-интерфейсы, необходимые для манипулирования данными.

Семантическая. Одно из наиболее важных требований к формату обмена состоит в том, что данные должны быть понятными. Если синтаксическая интероперабельность неразрывна с синтаксическим разбором данных, то семантическая связана с установлением соответствия между терминами, используемыми в данных, что требует анализа содержимого.

Интероперабельность и стандартизация взаимосвязаны. К сожалению, формальный процесс создания международных стандартов зачастую противоречит тому, что требуется для реальной интероперабельности распределенных ИР. И не только потому, что официальный процесс стандартизации слишком медлителен для быстроменяющегося мира: получающиеся стандарты еще и исключительно сложны. Многие международные стандарты никогда не испытывались в реальной жизни, а на практике работают только те стандарты, которые часто используются: иногда стандарт принимается де-факто за счет того, что его использует ведущая исследовательская группа.

Приведем основные аспекты интероперабельности ЭБ.

Наименование и идентификация. Необходимо несколько способов для идентификации материалов ЭБ. Каждый компьютер в Интернете имеет IP-адрес и доменное имя. Однако этих возможностей недостаточно. Библиотечные объекты нуждаются в идентификаторах, кото-

рые обозначают именно материал, а не его расположение в данный момент времени. Такие локально-независимые (location-independent) идентификаторы иногда называют Универсальными Именами Ресурсов (Uniform Resource Names, URN).

Форматы. Во всех известных ЭБ материалы хранятся в цифровой форме. С точки зрения интероперабельности особую трудность представляет текст. Формат ASCII, в 1980-е годы ставший практическим стандартом кодировки для компьютеров, имеет ограниченное число символов. В настоящее время развивается кодировка Unicode — расширенный набор символов, способный поддерживать очень широкий набор шрифтов. Но он все еще не поддерживается многими компьютерными системами. Широко поддерживаемый формат SGML, используемый в некоторых ЭБ как язык разметки и для текста, и для метаданных, является столь гибким и сложным, что достичь на его базе интероперабельности весьма нелегко. XML (упрощенная версия SGML) стал популярен в последнее время, и возможно, ему удастся занять нишу между простым HTML и исчерпывающим SGML.

Метаданные. Как уже отмечалось, метаданные часто делят на три категории: описательные (используются для библиографических целей, поиска и обработки), структурные (которые связывают различные объекты или части объектов между собой) и административные (для управления коллекцией и доступом к ней). Для интероперабельности необходим обмен этими метаданными. Это требует отдельного соглашения об именах полей метаданных, форматах кодировки и — по крайней мере — соглашение по семантике. Очевидный пример важности семантики: если в одной коллекции поле «Дата» используется для даты создания, а в другой — для даты включения в коллекцию, то это поле представляет относительную ценность.

Распределенный поиск. Пользователи часто хотят найти информацию из нескольких независимых коллекций одновременно. Они могут быть организованы логично и единообразно, но может существовать различие в описательных метаданных, используемых для поиска. Традиционный подход к поиску информации в различных коллекциях одновременно состоит в том, чтобы во всех них был стандартный набор метаданных и поддерживался один и тот же протокол. Все большее число специалистов осознает нереалистичность таких требований. Должна быть возможность поиска в различных коллекциях, даже если материал в них организован по-разному.

Сетевые протоколы. Перемещение материалов с одного компьютера на другой требует интероперабельности на сетевом уровне.

Практически универсальный набор интернет-протоколов в значительной степени решает эту проблему, но не всегда. К примеру, интернет-протоколы не слишком подходят для передачи непрерывных потоков данных, как видео- или аудиоматериалы, которые должны быть целостным потоком в предсказуемые интервалы времени.

Поисковые протоколы. В процессе выполнения операций компьютер посылает запрос другому с целью получить некоторые данные. Этот запрос должен быть передан по определенному протоколу, который может быть как простой (вроде HTTP), так и сложный (например, Z39.50). Идеальный протокол должен поддерживать идентификацию обоих компьютеров (с целью обеспечения безопасности), высокоуровневые запросы для определения доступных ресурсов, варианты поиска и возможности обработки, методы хранения и модификации промежуточных результатов, а также интерфейс для различных многочисленных форматов и процедур (наиболее близок к достижению этих целей Z39.50, но он малоприменим из-за собственной сложности и не отвечает всем критериям).

Идентификация и безопасность. Ряд наиболее сложных проблем интероперабельности между ЭБ включает идентификацию. Здесь есть три категории проблем. Первая — это идентификация пользователей. ЭБ вынуждены присваивать каждому пользователю «пользовательский ID» и пароль. Вторая — идентификация компьютеров. Системы, содержащие ценную информацию, особенно финансовую или конфиденциальную информацию, должны знать, с каким компьютером они соединяются. Можно полагаться на IP-адреса, но это крайне ненадежно. Третья — идентификация материалов. Люди должны быть уверены, что они получают аутентичную версию, которая не была изменена (случайно или намеренно).

Контрольные вопросы

1. Каковы особенности организации распределенной файловой системы?
2. В чем состоит главная проблема разделения файлов?
3. Каковы способы решения конфликтов при разделении файлов?
4. Перечислите основные виды параллелелизма при распределенной обработке данных.
5. Обоснуйте целесообразность разделения «клиентских» и «серверных» функций.

6. Обоснуйте назначение механизма неделимых транзакций.
7. Проведите сравнительный анализ распределения функций для различных базовых архитектур распределенной обработки.
8. Определите основные принципы и примерные структурные схемы распределенной обработки.
9. В чем отличие модели кластеров от модели единого пространства?
10. Чем отличаются мультикомпьютерные системы от кластеров?
11. Проведите сравнительный анализ видов параллелелизма.
12. Приведите примеры интернет-вычислений.
13. Сформулируйте основные требования к системам управления распределенными базами данных.
14. Перечислите основные условия и предпосылки появления систем управления распределенными базами данных.
15. Перечислите основные различия системы распределенной обработки данных и системы распределенных баз данных.
16. Обоснуйте целесообразность разделения «клиентских» и «серверных» функций.
17. Проведите сравнительный анализ распределения функций для различных базовых архитектур.
18. Определите основные принципы и примерные структурные схемы сервера распределенной обработки.
19. Перечислите основные решения распределенной обработки на основе межмодульного взаимодействия.
20. Назовите основные классы распределенных информационных ресурсов.
21. Определите алгоритм поиска с использованием информационных объектов разного уровня (первичная, вторичная, метаинформация).
22. Почему использование протокола HTTP для реализации ИПС вызывает трудности?
23. Перечислите основные факторы, определяющие интероперабельность ИР.

Список литературы

Нормативная и справочная литература

1. Об информации, информационных технологиях и о защите информации. Федеральный закон от 27 июля 2006 г. № 149-ФЗ.

2. ГОСТ ИСО/МЭК 2382-1—99. Информационная технология. Словарь. Часть 1. Основные термины.

3. ГОСТ Р ИСО 15489-1—2007. Система стандартов по информации, библиотечному и издательскому делу. Управление документами. Общие требования.

4. ГОСТ Р ИСО/МЭК 9126—93. Информационная технология. Оценка программной продукции. Характеристики качества и руководство по их применению. Государственный стандарт Российской федерации. Издание официальное. М.: Госстандарт России, 1994.

5. Информационно-библиотечная деятельность, библиография. Термины и определения / Межгосударственный стандарт ГОСТ 7.0—99 / Система стандартов по информации, библиотечному и издательскому делу. Минск, 2000.

6. Тезаурус информационно-поисковый одноязычный. Правила разработки, структура, состав и форма представления. Государственный стандарт Союза ССР. ГОСТ 7.25—80. (СТ СЭВ 174—85). М.: Государственный комитет СССР по стандартам, 1988.

7. CODASYL DBTG Report. New York: ACM, 1969.

8. ANSI/X3/SPARC Study Group on Data Base Management Systems. Interim Report. FDT Bull. ASM-SIGMOD. 1975. Vol. 7. No 2.

Дополнительная литература

1. Бинарное изображение. URL: http://ru.wikipedia.org/wiki

2. *Винер Н.* Кибернетика и общество. М.: Наука, 1968.

3. *Голицына О.Л., Максимов Н.В., Попов И.И.* Базы данных: учеб. пособие. 3-е изд. М.: ФОРУМ: ИНФРА-М, 2012.

4. *Голицына О.Л., Максимов Н.В., Попов И.И.* Информационные системы: учеб. пособие. М.: ФОРУМ, 2007.

5. *Дейт К. Дж.* Введение в системы баз данных / пер. с англ. 7-е изд. М.: Вильямс, 2001.

6. Зарождение печати. URL: http://www.gosreglament.ru/article/history.shtml

7. Информатика. Базовый курс: учебник для вузов / С.В. Симонович и др. СПб.: Питер, 1999.

8. *Криницкий Н.А., Миронов Г.Д., Фролов Г.Д.* Автоматизированные информационные системы / под ред. А.А. Дородницына М.: Наука, 1982.

9. *Колин К.К.* Информационные проблемы социально-экономического развития общества // Проблемы социальной информатики. Вып. 1. М., 1995.

10. *Лейнер Б., Среф В., Кларк Д.* и др. Краткий курс истории Интернет // JetInfo. 1997. № 14(45).

11. *Мазур М.* Качественная теория информация. М.: Мир, 1974.

12. *Максимов Н.В., Попов И.И.* Компьютерные сети: учеб. пособие. М.: ФОРУМ: ИНФРА-М, 2003.

13. *Мартин Дж.* Организация баз данных в вычислительных системах. М.: Мир, 1980.

14. Математический словарь. М., 1988.

15. *Михайлов А.М., Черный А.И., Гиляревский Р.С.* Основы информатики. М.: Наука, 1968.

16. *Муранивский Т.В.* Теоретические основы научно-технической информации. М.: МГИАИ, 1982.

17. *Ожегов С.И.* Словарь русского языка. М., 1989.

18. *Смирнов В.А., Финн В.К.* Предисловие к книге: *Белнап Н., Стил Т.* Логика вопросов и ответов. М.: Прогресс, 1981.

19. *Солтон Дж.* Динамические библиотечно-информационные системы / пер. В.Р. Хисамутдинов. М.: Мир, 1979.

20. *Самарин Ю.Н., Сапошников Н.П., Синяк М.А.* Допечатное оборудование: учеб. пособие. URL: http://www.hi-edu.ru/e-books/xbook341/01/part-002.htm

21. *Mizzaro S.* How many relevances in information retrieval? // Interacting With Computers, 10(3). P. 305—322, June 1998.

22. *Tailor R.S.* Question-negotiation and information seeking in libraries // College and Research Libraries, 1968, 29. P. 178—194.

23. Документальная информационно-аналитическая система xIRBIS: программа для ЭВМ / Н.В. Максимов, Е.Н. Васина, О.Л. Голицына и др. / Свидетельство о гос. регистрации № 2008611511 от 25.03.2008.

Приложения

Приложение 1

Глоссарий

AIIM (Association for Information and Image Management International) — международная ассоциация по информации и обработке изображений. Аккредитована ANSI как организация по развитию стандартов. AIIM представляет США в Международной организации по стандартизации и является представительной организацией для промышленных коалиций продавцов и конечных пользователей. Занимается созданием способных к взаимодействию стандартов для технологий управления документами во всем мире.

Anchors — гипертекстовые ссылки, внедренные в Web-документ. Позволяют пользователю переходить от одного фрагмента информации к другому независимо от места ее хранения в Internet.

ANSI (American National Standards Institute) — американский национальный институт стандартов — неправительственная организация, устанавливающая стандарты. Развивает и издает стандарты для «добровольного» использования в Соединенных Штатах. Набор стандартов принимается национальными организациями через поставщиков данной страны.

API (Application Program Interface) — интерфейс прикладной программы. Функциональный интерфейс, поддерживаемый операционной системой (ОС) или специальной программой, который позволяет прикладной программе использовать специфические данные или функции ОС или программы.

APRP (Adaptive Pattern Recognition) — адаптивное распознавание образов.

ASCII (American Standard Code for Information Interchange) — американский стандартный код для обмена информацией — соглашение для представления символьной информации; код для представления английской текстовой информации, используемый с отдельными модификациями в большинстве вычислительных систем.

Asynchronous Transaction — 1. Возврат управления пользователю после посылки запроса на сервер (станцию) для выполнения операции, в то время как

станция выполняет операцию. Это позволяет пользователю решать другие задачи, в то время пока станция завершает операцию. 2. Операция, в которой управление возвращается вызывающей программе до завершения запрошенной операции. Значение, возвращаемое при вызове асинхронной транзакции, обычно указывает результат попытки начать операцию. Абонент должен иметь некоторый дополнительный механизм для определения состояния завершения операции.

Authentication — установление личности пользователя, делающего попытку доступа к системе.

Authorization — определение набора привилегий, которыми обладает пользователь.

Backup — резервное копирование. Процесс (регулярный или разовый) копирования данных на другие носители, обычно оптические или ленточные. Все файлы или только недавно измененные маркируются для последующего копирования.

BLOB (Binary Large Object) — тип данных СУБД, используется для хранения произвольной информации, которая может быть представлена в двоичном виде. Тип данных BLOB является частью структуры базы данных, которая обеспечивает полную функциональность СУБД для манипулирования BLOB-элементами. То есть BLOB-элементы могут создаваться, удаляться, проверяться или копироваться. Но чаще всего отсутствует возможность работы внутри BLOB. Например, невозможно извлечение частей текста, индексирование и перемещение по BLOB.

CASE-средства (технологии) — программные средства, поддерживающие процессы создания и сопровождения ИС, включая анализ и формулировку требований, проектирование прикладного ПО (приложений) и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы.

CCITT (Consultative Committee on International Telegraphique et Telephonique) — Международный консультативный комитет по телеграфии и телефонии, МККТТ, в настоящее время ITU-T.

CD-ROM (Compact Disc Read Only Memory) — постоянная память на компакт-дисках.

COM (Component Object Model) — составляющая программного обеспечения, поддерживающая OLE.

Content — содержательная часть данных документа, в противоположность атрибутам. Может включать текст, изображения, видео, звук, программы или любой другой материал, содержащийся на бумаге, дискете, компакт-диске (CD-ROM) и др. Отметим, что некоторые системы управления документами расценивают данные как один из атрибутов.

Conversion — изменение формата документа или его части. Может быть классифицировано по типам преобразования — преобразование символьных

наборов, преобразование форматов текстовых процессоров или преобразование языков описания страницы. Преобразования, изменяющие логическую структуру документа, считаются трансформацией документа.

CP 866 — распространенная в РФ кодировка символьной информации на базе кода ASCII с расширением его до 256 символов: кодовая страница 866 для IBM PC, в части кириллицы отсортирована по алфавиту, используется для работы с немодифицируемыми (нерусскоязычными) программами в ОС типа MS-DOS, сохраняет наиболее часто используемые в программах псевдографические знаки.

Data mining — «добыча» данных. Набор методов, позволяющих извлекать из сырых данных ранее неизвестные знания о зависимостях и закономерностях поведения рассматриваемого объекта. При этом все результаты формулируются в текстовых и графических формах, удобных для восприятия человеком.

Data model — модель данных. Описание содержания базы данных на более детализированном уровне, чем требуется непосредственно системе управления базы данных.

Data transformation — преобразование данных. Процесс изменения данных при начальной загрузке или при выполнении перемещения данных. Данные могут быть преобразованы для улучшения удобочитаемости при объединении данных из различных источников, для улучшения качества данных при их суммировании и т. д.

Data warehouse — хранилище (склад, кладовая) данных. База данных, разработанная для решения прикладных задач, в основном, из области принятия решений. Данные извлекаются из файловых систем операционных систем из всевозможных СУБД и т. п. Затем они преобразуются и объединяются, чтобы обеспечить анализ по разным разрезам.

Datamart — небольшое хранилище (витрина) данных. Термин появился как расширение аналогии продажи в розницу, которая произвела термин «хранилище данных». Если хранилище содержит большие объемы данных, то витрина — та часть содержимого, которая не содержит данных, не представляет интереса.

DDE (Dynamic Data Exchange) — динамический обмен данными.

Digital Video — видео, фиксируемое в цифровом формате.

DirectX — предложенная Microsoft система команд управления позиционированием виртуального звукового источника.

DMS (Document Management System) — система управления документами.

Document Content Model — структура составного документа.

Document Interchange Format — правила представления документов с целью обмена.

DOS — дисковая, однозадачная, однопользовательская операционная система с интерфейсом командной строки.

dpi (dot per inch) — плотность печати или разрешение сканирования в точках на дюйм.

dpi (dots per inch — точек на дюйм) — единица измерения разрешения, в частности, оптического разрешения сканера.

DTD (Document Type Definition) — определение типа документа — начало (преамбула) SGML-документа, где определяются компоненты документа и его структура. Описание типа (шаблона) документа.

EAX (Environmental Audio Extensions) — модель добавления реверберации с учетом звуковых препятствий и поглощения звуков.

EDI (Electronic Data Interchange) — обмен данными и документами между различными пользователями согласно стандартным (ANSI X.12, EDIFACT) правилам.

EDIFACT (Electronic Data Interchange For Administration,Commerce And Transport) — электронный обмен данными в управлении, торговле и на транспорте (ISO 9735—1987).

Embedding — размещение (вложение, внедрение) данных в составном документе, при котором данные и связанные с ними управляющие приложения физически размещены внутри документа.

FTP (File Transfer Protocol) — протокол передачи файлов — Internet-протокол для передачи файлов между компьютерами.

GIF (Graphics Interchange Format) — формат хранения и распространения файлов изображений.

GUI (Graphical User Interface) — графический интерфейс пользователя.

HTML (Hypertext Markup Language) — язык высокого уровня для определения структуры документов.

HTTP (Hyper Text Transfer Protocol) — Internet-протокол передачи (получения) документов HTML.

ICR — аббревиатура слов Intelligent Character Recognition, «интеллектуальное распознавание символов». Так называют технологии или системы, предназначенные для массовой обработки документов, заполненных печатными буквами и цифрами от руки, т. е. для распознавания рукописных символов. Если OCR-система должна построить точную электронную модель исходного документа, то от ICR-системы требуется найти на изображении документа информацию, извлечь ее и передать во внешнюю базу данных. Извлеченные данные упорядочиваются по заранее заданным правилам, а как выглядит и какую структуру имеет исходный документ, при этом несущественно.

Internet — сеть сетей, объединяющая множество компьютерных сетей во всем мире и предоставляющая доступ к мировым информационным ресурсам.

Interoperability — интероперабельность. Информационная, техническая или функциональная совместимость.

Intranet — корпоративная сеть, использующая протоколы и стандарты Internet.

IPA, принципы IPA (Integrity, Purposefulness, Adaptability) — принципы целостности, целенаправленности, адаптивности. На этих принципах базируется восприятие животных и людей.

ISO (International Organization for Standardization) — Международная организация по стандартизации (ВОС).

JPEG — 1. **Joint Photographic Experts Group**. — объединенная экспертная группа по фотографии, разработавшая алгоритм сжатия изображения. 2. Стандартный алгоритм сжатия неподвижного изображения, разработанный группой JPEG. Сжатие по этому алгоритму основано на психовизуальном восприятии изображений человеком и ведет к потере информации за счет исключения мелких деталей.

LAN (Local Area Network) — локальная компьютерная сеть.

Linking — объединение (связывание) объектов в составной документ, вследствие чего ссылка связи, вставленная в документ, указывает на фактические данные, которые физически находятся в другом месте документа или в каком-то другом документе.

Localization — локализация. Адаптация программного продукта к национальным особенностям страны или географического региона, в котором он используется. Например, разработчики программ обработки текстов должны локализовать алгоритмы сортировки списков для различных алфавитов.

MDA (Multilevel Document Analysis) — «многоуровневый анализ документа».

Metadata — метаданные.

Middleware — программное обеспечение, обеспечивающее интерфейс высокого уровня, освобождающий разработчика прикладных программ от знания сложностей аппаратных средств, операционной системы и сетевой семантики.

MIDI (Musical Instrument Digital Interface) — протокол передачи и интерпретации команд управления воспроизведением звука. Применяется в звуковых картах и определяет основные средства для управления расположением инструментов, голосов, а также для деления на инструментальные группы (клавишные, ударные и т. д.).

MIDI секвенсор — устройство, которое записывает и воспроизводит команды MIDI, а не аудиосигналы.

MIME (Multipurpose Internet Mail Extention) — многоцелевое расширение электронной почты Internet. Официально предложенный стандарт электронной почты в Internet. MIME-формат позволяет включать в сообщение электронной почты помимо текста также изображения, звук, видео.

NNTP (Network News Transfer Protocol) — сетевой протокол передачи новостей. Служит для помещения и извлечения статей в телеконференциях.

OCR (Optical Character Recognition) — распознающая программа для ввода документов с использованием оптического сканера.

ODA (Office Data/Document Architecture) — архитектура деловых документов (стандарт ISO 8613).

ODBC (Open Database Connectivity) — интерфейс вызова данных в гетерогенной среде реляционных и нереляционных систем управления базами данных. ODBC предназначен обеспечить универсальный набор команд интерфейса для доступа к данным, что обеспечивает доступ к множественным различным базам данных.

ODIF (Office Document Interchange Format) — формат обмена документами в делопроизводстве (ISO 8613).

ODMA (Open Document Management API) — API для связи прикладных программ с системой управления документами и другим групповым ПО.

OLAP (On-line analytical processing) — аналитическая обработка данных в оперативном режиме. Прикладное ПО для анализа информации, хранящейся в базе данных.

On-line — 1. Режим работы с компьютером или каким-либо другим устройством, при котором подразумевается постоянное с ним взаимодействие. Синонимы: интерактивный, диалоговый, оперативный. 2. Постоянно включенное устройство; неавтономный режим работы.

OSI (Open System Interconnection) — взамосвязь открытых систем. Иной (не Internet) набор сетевых протоколов, предложенный ISO. Этот стандарт сетевого и межсетевого взаимодействия определяет семь уровней взаимодействия компонентов сети: физический, канальный, сетевой, транспортный, сеансовый, уровень представления данных и прикладной. Для каждого уровня разработан один или несколько протоколов, которые обеспечивают сетевое взаимодействие широкого класса устройств.

PDF — аббревиатура слов Portable Document Format, «универсальный формат документов». Термин введен корпорацией Adobe, которой был разработан данный формат.

Plug-and-play — «вставляй и работай». Способ, реализуемый в устройствах для массового непрофессионального пользователя.

Point-and-click — «укажи и щелкни». В GUI способ запуска различных приложений.

Postscript — распространенный формат электронных документов — язык описания страниц печатных документов для лазерных принтеров и других устройств вывода. Разработан фирмой Adobe.

RAID (Redundant Array of Independent Disc) — дисковый массив, обеспечивающий резервирование и дублирование данных.

Recovery — восстановление, возобновление, возврат, возврат в исходное состояние.

Replication — процесс физического дублирования данных из одной базы данных в другую. Некоторые репликаторы позволяют двунаправленное копирование, где любая копируемая база данных может модифицироваться, тогда изменения автоматически распространяются в другую.

Repository — корпоративный информационный ресурс, содержащий всю разработку, предоставленную от анализа до кодов программ, и способный к сохранению версий и конфигураций.

Router — маршрутизатор, устройство для передачи сетевых пакетов из одной сети в другую на основе информации, содержащейся в передаваемом пакете. Сетевой шлюз является наиболее типичным представителем маршрутизаторов.

RPC (Remote Procedure Call) — вызов удаленной процедуры, дистанционный вызов процедуры. Используется в серверной части приложения. Механизм RPC скрывает от программиста детали сетевых протоколов нижележащих уровней.

Scanning — сканирование — процесс преобразования информации, находящейся на твердом носителе, в цифровой формат.

SGML (Standard Generalized Markup Language) — язык разметки высокого уровня для представления документов сложной структуры, обычно используемых в технических приложениях.

SMTP (Simple Mail Transfer Protocol) — простой (упрощенный) протокол электронной почты. Прикладная служба в сетях TCP/IP для передачи текстовых сообщений.

SNA (Systems Network Architecture) — сетевая архитектура систем. Разработана корпорацией IBM для организации сети своих хост-машин и терминалов. Состоит из семи уровней протоколов, которые подобны уровням модели OSI. Определяет способы передачи информации: иерархический (связь между хост-машиной и терминалами) и одноранговый (равноправный).

SQL (Structured Query Language) — структурированный язык запросов. Стандартный язык запросов, используемый для обращения к реляционным базам данных.

TCP/IP (Transmission Control Protocol/Internet Protocol) — набор протоколов для коммуникации в локальной сети или во взаимосвязанном наборе сетей. Основной протокол Internet/Intranet.

TIFF (Tagged Image File Format) — теговый формат файла изображений.

URL (Universal Resource Locator) — последовательность символов, обозначающая адрес документа (или его части) на сервере Паутины. Типичный URL содержит 3 части: используемый протокол при извлечении документа (ftp, http и др.); доменное имя компьютера, где хранится документ; путь к документу (pathname) в локальной файловой системе; синтаксис URL — protocol://server_name/path.

WAN (Wide Area Network) — глобальная вычислительная сеть.

Web-site — место в Паутине (буквально) — первоначально Web-сервер или совокупность серверов Internet, которые представляли компании, университеты и другие организации во Всемирной паутине. По сути дела это логически обособленная совокупность гипермедиа-информационных объектов, объединенная общей темой. Следует отличать «сайт» от сервера. Сервер — объект сетевого пространства, в то время как сайт — объект информационного web-пространства. На сервере может располагаться множество сайтов.

WWW (World Wide Web) — Всемирная паутина.

X.400 — наборы протокольных стандартов для международной пересылки электронной почты. Этот стандарт для систем работы с сообщениями электронной почты позволяет включать в сообщения не только текстовую, но и другую информацию, например, факсы и графические изображения.

XML — eXtensible Markup Language, «расширяемый язык разметки». Современный инструмент для создания и обработки документов; его возможности используются многими программами.

Автоматизированная информационно-поисковая система (АИПС) — совокупное название как для программных систем и оболочек, ориентированных на ввод, хранение, поиск и выходное представление документов.

Агрегат данных — именованная совокупность элементов данных, представленных простой (векторной) или иерархической (группы или повторяющиеся группы) структурой. Примеры — массивы, записи, комплексные числа и пр.

Адаптивная бинаризация, adaptive binarization, AB — способ обработки изображения; алгоритм, выбирающий порог бинаризации в зависимости от контрастности данного участка изображения. Дает возможность точно распознавать текст со сложных оригиналов, например, ветхих, истертых страниц.

Администратор базы данных (АБД) — лицо или группа, уполномоченные для ведения БД (модификация структуры и содержания БД, активизация доступа пользователей, выполнение других административных функций, которые затрагивают всех пользователей).

Анализ документа — процедура обработки изображения, в ходе которой OCR-программа создает электронную редактируемую копию документа. Собственно распознавание текста — одна из составных частей анализа документа.

Архитектура документа — структурное описание документа, включающее в себя все входящие в него виды информации (текст, векторная и растровая графика, таблицы).

Атрибут — поле данных, содержащее информацию об объекте.

База данных (БД) — именованная совокупность взаимосвязанных данных, отображающая состояние объектов и их отношений в некоторой пред-

метной области, используемых несколькими пользователями и хранящихся с минимальной избыточностью.

Байт — 1. Единица количества информации, равная обычно восьми битам. 2. Ячейка памяти, соответствующая одному байту.

Бинаризация — перевод изображения в бинарный формат, когда каждая точка может быть либо белого, либо черного цвета.

Бит — 1. Двоичная единица количества информации. 2. Единица объема памяти, соответствующая одному биту информации.

Битрейт (bitrate) — ширина потока (битовая скорость). Для звукового сигнала термин обозначает общую ширину потока, безразлично к тому, монофонический или стереофонический сигнал он содержит.

Браузер — прикладная программа клиента, которая позволяет просматривать, извлекать и показывать содержание документов, находящихся на серверах Всемирной паутины.

Валидация — автоматическая проверка данных на соответствие заданным правилам.

Верификация — проверка данных оператором. Производится путем сличения результатов распознавания с исходным изображением части документа.

Вид документа — элемент в классификации множества документов.

Видеоадаптер — электронная плата, генерирующая видеосигнал, посылаемый видео дисплею по кабелю.

Всемирная паутина (WWW) — Internet-обслуживание, которое дает возможность пользователям читать и выбирать документы со всего мира.

Вторичный документ — документ, являющийся результатом аналитико-синтетической переработки одного или нескольких первичных документов.

Гипертекст — система представления информации, состоящая из узлов данных и смысловых связей между ними.

Глобальная вычислительная сеть — сеть передачи данных, охватывающая значительное географическое пространство (регион, страну, ряд стран, континенты).

Данные — представление информации в некотором формализованном виде, пригодном для передачи, интерпретации или обработки. Данные образуются при взаимодействии сигналов с физическими телами, когда в последних возникают определенные изменения свойств, т. е. происходит регистрация сигналов.

Дескриптор — нормативное ключевое слово, предназначенное для координатного индексирования документов и информационных запросов.

Дескрипторный язык — информационно-поисковый язык, словарный состав которого состоит из дескрипторов, а использование основано на принципе координатного индексирования.

Документ — агрегат данных в документальных системах (АИПС), имеющий иерархическую структуру и, кроме форматных полей (элементы или агрегаты данных фиксированной длины), обычно содержащий текстовые поля или символьные последовательности неопределенной длины.

Естественный язык — язык, словарь и грамматические правила которого обусловлены практикой применения и не всегда формально зафиксированы.

Запись логическая — идентифицируемая (именованная) совокупность элементов или агрегатов данных, воспринимаемая прикладной программой как единое целое при обмене информацией с внешней памятью. Запись — это упорядоченная в соответствии с характером взаимосвязей совокупность полей (элементов) данных, размещаемых в памяти в соответствии с их типом.

Запись физическая — совокупность данных, которая может быть считана или записана как единое целое одной командой ввода-вывода.

Запрос (информационный) — сообщение, обычно неформатированное, информационно-поисковой системе со стороны абонента, содержащее его информационную потребность и подвергающееся автоматическому индексированию.

Иерархическая модель данных — использует представление предметной области БД в форме иерархического дерева, узлы которого связаны по вертикали отношением «предок-потомок». Навигация в БД представляет собой перемещение по вертикали и горизонтали в данной структуре.

Импорт (загрузка, download) — утилита (функция, команда), служащая для чтения файлов операционной системы, которые содержат данные, представленные обычно в некотором коммуникативном формате.

Инвертированный файл (список) — файл, предназначенный для быстрого произвольного поиска записей по значениям ключей, организованный в виде независимых упорядоченных списков (индексов) ключей — значений определенных полей записей основного файла.

Индекс — таблица ссылок на объекты, используемая для определения адреса записи.

Индексирование — формирование описания документа как совокупности дескрипторов, выбираемых из заранее созданных словарей понятий либо из текстов документов.

Интеллектуальная фильтрация фона, intellectual background filtering, IBF — прогрессивный способ обработки изображения; удаление фоновых текстур или картинок перед распознаванием текста. Применяется для повышения точности распознавания. Технология IBF реализована в OCR-системе FineReader.

Интерфейс — разделительная граница, определенная с помощью характеристик этой границы.

Информационная система — система, предназначенная для хранения, обработки, поиска, распространения, передачи и предоставления информации.

Информационная технология — совокупность методов, производственных процессов и программно-технических средств, объединенных в технологический комплекс, обеспечивающий сбор, создание, хранение, накопление, обработку, поиск, вывод, копирование, передачу и распространение информации.

Информационно-поисковая система (ИПС) — программная система для хранения и поиска данных по неформатированным запросам. Для общения пользователя с ИПС разработчики системы стремятся применять упрощенный естественный язык.

Информационно-поисковый язык (ИПЯ) — искусственный язык, обеспечивающий компактную, строго алгоритмизированную запись содержания документов и запросов в ИПС. ИПЯ можно определить как специализированную семантическую систему, состоящую из алфавита, правил образования (грамматики) и правил интерпретации (семантики).

Информационные ресурсы — совокупность накопленной информации, зафиксированной на материальных носителях в любой форме, обеспечивающей ее передачу во времени и пространстве. В контексте автоматизированных информационных систем под информационными ресурсами обычно подразумевают информационные массивы и базы данных, рассматриваемые совместно с информационными технологиями, обеспечивающими их доступность.

Информационный запрос — записанный на естественном языке текст, выражающий некоторую информационную потребность.

Информационный поиск — процесс отыскания в поисковом массиве таких записей, которые соответствуют признакам, указанным в информационном запросе.

Информация — сведения, воспринимаемые человеком и (или) специальными устройствами как отражение фактов материального или духовного мира в процессе коммуникации.

Искусственный язык — язык, специально созданный и регулируемый на основе согласованных принципов.

Классификация — процесс соотнесения содержания документов с понятиями, зафиксированными в заранее составленных систематических (классификационных) схемах.

Клиент/сервер — технология (архитектура) взаимодействия клиента и сервера. Клиент — программа, запрашивающая у сервера информацию или выполнение какого-либо задания на сервере от имени клиента. Сервер — прикладная программа, исполняющая запросы клиента. Клиент и сервер взаимодействуют по определенному протоколу. Программа клиента и программа сервера могут располагаться как на одной машине, так и на совершенно различных компьютерах произвольной сети.

Ключ — значение (элемент данных), используемые для идентификации или определения адреса записи.

Ключевое поле — поле в структуре записи. Поле определяют как ключевое (или индексированное) для убыстрения или упрощения операций поиска и/или для модификации операций обработки данных.

Ключевое слово — предметное слово, выбираемое из некоторого текста (документа) и используемое для координатного индексирования этого текста (документа).

Код — система представления информации в виде данных, состоящая из набора условных знаков и правил присвоения им значений.

КОИ (ГОСТ 19768—74) — используемая в СССР и РФ кодировка символьной информации на базе кода ASCII с расширением его до 256 символов: используется в ряде систем типа UNIX; в части кириллицы эта кодировка не отсортирована по алфавиту и, следовательно, не позволяет использовать большинство зарубежных программ без соответствующих модификаций.

Коммуникативные (обменные) форматы данных — соглашения о представлении агрегатов информации при передаче.

Контрастность — параметр, показывающий, насколько самый темный участок изображения отличается от самого светлого. Влияет на качество распознавания.

Координатное индексирование — индексирование, при котором основное содержание документа представляется в виде сочетания ключевых слов, или дескрипторов.

Лемматизация — нахождение начальной формы слова по любой его словоформе.

Логическая структура БД — определение БД на физически независимом уровне.

Логический файл — файл в представлении прикладной задачи, состоящий из логических записей, структура которых может отличаться от структуры физических записей, представляющих информацию в памяти.

Локальная вычислительная сеть (ЛВС) — коммуникационная система, поддерживающая в пределах одного здания или некоторой ограниченной территории один или несколько высокоскоростных каналов передачи цифровой информации, предоставляемых подключаемым устройствам для кратковременного монопольного использования.

Метаданные — информация, которая описывает другие данные с помощью таких атрибутов, как их структура, ассоциации, типы и диапазоны.

Методы поиска — совокупность моделей и алгоритмов реализации отдельных технологических этапов, таких, как построение поискового образа запроса, отбор документов (сопоставление поисковых образов запросов и документов), расширение и реформулирование запроса, локализация и оценка выдачи.

Механизмы поиска — реализованные в системе модели и алгоритмы процесса формирования выдачи документов в ответ на поисковый запрос.

Модель данных — базовый инструментарий, обеспечивающий на формальном абстрактном уровне конкретные способы представления объектов и связей.

Модель физическая — определяющая размещение и способы поиска данных на внешних запоминающих устройствах СУБД.

Морфологический поиск — поиск с учетом морфологии (всех возможных форм слова).

Мультимедиа — среда, материал, состоящий из комбинации текста, графики, видео, мультипликации и звука, представляющий таким образом информацию в более понятном и удобообрабатываемом виде.

Мэйнфрейм — компьютер высокой мощности, разработанный для решения наиболее интенсивных вычислительных задач. Обычно используется одновременно многими пользователями.

Навигатор (browser) — программа навигации и просмотра, размещающаяся на рабочем месте пользователя, клиентская программа в сети Всемирной паутины.

Навигация — целенаправленная, определяемая стратегией последовательность использования методов, средств и технологий конкретной АИПС для получения и оценки результата поиска.

Независимость данных логическая (физическая) — свойство системы, обеспечивающее возможность изменять логическую (физическую) структуру данных без изменения физической (логической).

Носитель информации (данных) — средства регистрации, хранения, передачи информ. ции (данных).

Операционная система (ОС) — общее название программ и программных комплексов, расширяющих функциональные возможности аппаратуры вычислительных машин, повышающих эффективность использования вычислительных средств и облегчающих взаимодействие пользователя с машиной.

Открытая система — 1. Система, имеющая возможность расширения за счет средств среды, в которой она функционирует. 2. Система, независимая от изготовителей ИС, удовлетворяющая требованиям ряда международных стандартов.

Открытость — свойство информационных технологий и систем, предполагающее способность объединять разные информационные системы, аппаратуру и программные продукты различных производителей, что делает возможным обмен между ними данными, распределенный доступ к информационным ресурсам.

Отношение (relation) — агрегат данных, хранящийся в одной из таблиц (строка таблицы) табличной, реляционной БД или создаваемый виртуально в процессе выполнения операции над базой данных при выполнении запросов к данным.

Поисковый образ документа (ПОД) — описание документа, выраженное средствами ИПЯ и характеризующее основное смысловое содержание или какие-либо другие признаки этого документа, необходимые для его поиска по запросу.

Поисковый образ запроса (ПОЗ) — записанный на ИПЯ текст, выражающий смысловое содержание информационного запроса и содержащий указания, необходимые для наиболее эффективного осуществления информационного поиска.

Полнотекстовые документы (записи) — полный (или почти) исходный текст журнальной статьи или другого документа.

Пользователь БД — программа или человек, обращающийся к базе данных с помощью средств управления данными СУБД.

Предметная область (ПрО) — набор объектов, представляющих интерес для актуальных или предполагаемых пользователей, когда реальный мир отображается совокупностью конкретных и абстрактных понятий, между которыми фиксируются определенные связи.

Проектирование базы данных — упорядоченный формализованный процесс создания системы взаимосвязанных описаний — моделей предметной области, которые связывают хранимые в базе данные с объектами предметной области, описываемыми этими данными.

Протокол — совокупность определений (соглашений, правил), регламентирующих формат и процедуры обмена информацией между двумя или несколькими независимыми устройствами или процессами, т. е. описание того, как программы, компьютеры или иные устройства должны действовать, когда они взаимодействуют друг с другом.

Профиль документа — в ODA набор свойств документа, которые относятся к документу в целом.

Рабочая станция — 1. Комбинация устройств ввода-вывода и вычислительных аппаратных средств, используемых отдельным пользователем. 2. Автономный компьютер для выполнения отдельных прикладных программ (функций).

Разметка — дополнительная информация, включаемая в документ и выполняющая функции выделения логических элементов данного документа и задания процедур обработки выделенных элементов.

Разрешение оптическое — параметр сканера, характеризующий предельно достижимую детальность считывания информации с оригинала, указывается в точках на дюйм (dpi).

Распознавание документа — построение редактируемой электронной копии бумажного документа. Как правило, проводится в два этапа; сначала с помощью сканера получают электронную «фотографию» страницы, затем обрабатывают ее специальной OCR-программой. Результатом работы OCR-про-

граммы становится точная электронная копия документа, которую можно редактировать, сохранять в различных форматах, распечатывать и т. д.

Распределенная база данных — совокупность баз данных, которые обрабатываются и управляются по отдельности, а также могут разделять информацию.

Релевантность — свойство некоторой информации (документ, факт, и пр.) удовлетворять информационную потребность пользователя АИС (relevant — относящийся к делу).

Реляционная алгебра — алгебра (язык), включающая набор операций для манипулирования отношениями.

Реляционная база данных — база данных, состоящая из отношений. Здесь вся информация, доступная пользователю, организована в виде таблиц, обычно имеющих уникальные имена, состоящих из строк и столбцов, на пересечении которых содержатся значения данных, а операции над данными сводятся к операциям над этими таблицами.

Реляционная СУБД (РСУБД) — система управления базами данных, поддерживающая реляционную модель данных (РМД).

Репозитарий — архив, склад, кладовая.

Сетевой сервер — сетевой (хост-) компьютер, выполняющий системные функции отработки сетевых протоколов для связанных с сервером других сетевых компьютеров, обычно рабочих станций. Сетевой сервер обычно выполняет одну или несколько обслуживающих функций, таких, как файловый сервер, FTP-сервер, Web-сервер и др.

Система управления базами данных (СУБД) — совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Словарь данных — исчерпывающий набор таблиц или файлов, представляющий собой каталог всех описаний данных (имен, типов). Может содержать также информацию о пользователях, привилегиях и т. д., доступную только администратору базы данных. Является центральным источником информации для СУБД, АБД и всех пользователей.

Содержание документа — в ODA представляет собственно информацию документа: текст, рисунки и т. п.

Стратегия поиска — общий план (концепция, предпочтение, предрасположенность, установка) поведения пользователя для выражения и удовлетворения информационной потребности, обусловленный характером цели и типом поиска, архитектурой БД, а также методами и средствами поиска конкретной АИПС.

Структура данных — атрибутивная форма представления свойств и связей предметной области, ориентированная на выражение описания данных средствами формальных языков и таким образом учитывающая возможности и ограничения конкретных средств с целью сведения описаний к стандартным ти-

пам и регулярным связям. Структура данных с точки зрения программирования — это способ отображения значений в памяти — размер области и порядок ее выделения (который и определит характер процедуры адресации-выборки).

Таблица — основная единица информации в системе управления реляционной базой данных. Состоит из одной или более единиц информации (строк), каждая из которых содержит значения некоторого вида (столбцы).

Тег — признак. Часть элемента данных (обычно один или несколько разрядов), определяющих его тип.

Тезаурус — семантическая сеть, в которой понятия связаны регулярными и устойчивыми семантическими отношениями — иерархическими (например, род-вид, целое-часть), ассоциативными, а также отношениями эквивалентности.

Текстовый слой (PDF-документа) — часть документа, сохраненного в формате PDF, которая содержит часть текста или даже весь текст документа.

Терминал — устройство, содержащее видеоадаптер, дисплей и клавиатуру. Адаптер и дисплей (иногда и клавиатура) обычно скомпонованы в одном устройстве.

Технологии поисковые — унифицированные (оптимизированные в рамках конкретной АИПС) последовательности эффективного использования в процессе взаимодействия пользователя с системой отдельных средств поиска для устойчивого получения конечного и, возможно, промежуточных результатов.

Типы данных — совокупность соглашений о программно-аппаратурной форме представления и обработки, а также ввода, контроля и вывода элементарных данных, к типам данных прежде всего относятся классические типы — целое число, действительное число, булевское значение.

Топология БД — схема распределения компонент базы данных по физическим носителям, в том числе различным узлам вычислительной сети.

Точка сохранения — момент времени, когда в БД записывается вся работа в транзакции. В транзакции могут применяться ряд точек сохранения, выступающих в роли промежуточных точек для работы.

Точность распознавания — основной параметр, характеризующий качество работы OCR-программы. Численно равен отношению количества правильно распознанных символов к общему количеству символов в документе и выражается в процентах.

Транзакция — последовательность операций над данными базы, переводящая БД из одного непротиворечивого состояния в другое, которая может быть представлена как одно «событие».

Уровни представления данных — концептуальный, внутренний и внешний. Внутренний уровень — глобальное представление БД, определяет необходимые условия в первую очередь для организации хранения данных на внешних запоминающих устройствах. Представление на концептуальном уровне является обобщенным взглядом на данные с позиций предметной об-

ласти. Внешний уровень представляет потребности пользователей и прикладных программ.

Файл — именуемая единица информации, поддерживаемая операционной системой. Доступ к данным реализуется либо в рамках ОС, либо пользовательскими программами, либо в рамках СУБД, либо комбинированно. Обычно ОС может предоставить пользовательским программам не более двух типов файлов: записе-ориентированные, когда при обращении к файлу из пользовательской программы считывается или выводится в файл запись (агрегат или элемент данных — логическая единица информации) и потоко-ориентированные, когда пользовательской программе предоставляется для записи или чтения физический элемент файла (очередной бит или байт данных).

Файл бинарный — файл, содержащий произвольную двоичную информацию (текст с бинарной разметкой, программа, графика, архивный файл).

Файл графический — бинарный файл, содержащий данные, обычно полученные с помощью растрового сканера и соответствующие двумерному изображению объекта.

Файл-сервер — установленное в сети устройство хранения файлов, доступное всем пользователям сети. Не только хранит файлы, но и управляет ими, поддерживает порядок при запросе файлов пользователями сети и вносит в них изменения.

Файл текстовый — файл, содержащий символьную информацию в одном из соответствующих кодов, и коды, управляющие режимом отображения символов на печать и экранные устройства.

Формат — способ расположения и представления данных на носителе информации.

Форматы файлов — представление информации на уровне взаимодействия операционной системы с прикладными программами.

Целостность — свойство БД, при котором она удовлетворяет некоторым определенным ограничениям значений данных и сохраняет это свойство при всех модификациях (замена, добавление или удаление) данных.

Централизованное управление данными — осуществляется средствами, входящими в состав СУБД, обеспечивает: сокращение избыточности в хранимых данных; устранение несовместимости в хранимых данных многими приложениями; совместное использование хранимых данных, что достигается необходимой интеграцией данных; целостность данных, которая достигается с помощью процедур, предотвращающих внесение в БД неверных данных и ее восстановление после отказов системы; лучший учет противоречивых требований к использованию БД в различных приложениях посредством соответствующего структурирования БД.

Шлюз — устройство для соединения разнотипных сетей, работающих по разным протоколам связи в целях обеспечения передачи информации из одной сети в другую.

Экспорт (выгрузка, upload) — утилита (функция, команда), служащая для вывода информации из системы в файл(ы) в некотором коммуникативном формате.

Электронная почта — передача сообщений по компьютерной сети. Электронная почта представляет собой вариант почтовой службы, который предназначен для взаимодействия компьютеров (или терминалов). Дает пользователю возможность отправлять и принимать сообщения и (в некоторых случаях) изображения или речевые послания, предназначенные как индивидуальным адресатам, так и группам пользователей (конференции).

Электронно-цифровая подпись (ЭЦП) — аналог личной подписи сотрудника, который служит для заверения электронных документов. Гарантией однозначной авторизации подписанного электронного документа и невозможности подделки такой подписи является специальная криптографическая функция, лежащая в основе алгоритма выработки ЭЦП.

Электронный документ — документ, носителем которого является электронная среда — МД, МЛ, компакт-диск и т. д.

Элемент данных (элементарное данное) — неделимое именованное данное, характеризующееся типом (например, символьный, числовой, логический и пр.), длиной (в байтах) и обычно рассчитанное на размещение в одном машинном слове соответствующей разрядности. Это минимальная адресуемая (идентифицируемая) часть памяти — единица данных, на которую можно ссылаться при обращении к данным.

Элемент текста — часть текста, ограниченная начальной и конечной метками.

Язык манипулирования данными (ЯМД). ЯМД обычно включает в себя средства запросов к базе данных и поддержания базы данных (добавление, удаление, обновление данных, создание и уничтожение БД, изменение определений БД, обеспечение запросов к справочнику БД).

Язык описания данных (ЯОД) — средство внутрисистемного определения данных, представляющего обобщение внешних взглядов. Описание представляет собой модель данных и их отношений, т. е. структур, из которых образуется БД.

Язык структурированных запросов (SQL) — основной интерфейс пользователя и АБД для запоминания и поиска информации в базе данных для ряда СУБД.

Фрагмент таблицы УДК

004 Информационные технологии. Вычислительная техника. Теория, технология и применения вычислительных машин и систем.

Все подклассы 004 подразумевают цифровую обработку данных, кроме тех, которые включают индексы 004.386 или 004.387.

=> 621.3.049.77 Микроэлектроника. Интегральные схемы

=> 621.39 Электросвязь

Основные деления

004.2 Архитектура вычислительных машин
004.3 Аппаратные средства. Техническое обеспечение
004.4 Программные средства
004.5 Человеко-машинное взаимодействие. Пользовательский интерфейс
004.6 Данные
004.7 Связь компьютеров. Сети ЭВМ. Вычислительные сети
004.8 Искусственный интеллект
004.9 Прикладные информационные (компьютерные) технологии

Специальные определители

004.01 Документация
004.02 Методы решения задач
004.021 Алгоритмы
=> 004.421 Алгоритмы составления программ
004.023 Эвристические методы
004.03 Типы и характеристики систем
004.031 Типы систем
004.031.2 Автономные системы. Системы с пакетной обработкой
004.031.4 Неавтономные системы. Онлайновые системы
004.031.42 Интерактивные системы
004.031.43 Системы реального времени. Системы обработки транзакций
004.031.6 Встроенные системы
004.032 Характеристики систем
004.032.2 Режим обработки данных
004.032.22 последовательный
004.032.24 параллельный, одновременный

=> 004.272 Архитектуры параллельной обработки
004.032.26 Нейронные сети
004.032.3 Согласование по времени. Задание времени цикла
=> 004.074.34 Время цикла памяти
004.032.32 Синхронные процессы
=> 004.451.23 Синхронизация
004.032.322 Синхронизирующая частота. Тактовая частота
004.032.324 Период синхронизирующих импульсов. Такт. Машинный цикл. Цикл внутреннего тактирования
004.032.34 Асинхронные процессы
004.032.6 Мультимедиа
=> 004.357 Акустическая и мультимедийная периферия. Устройства ввода данных с голоса
=> 004.427 Средства разработки мультимедиа
004.032.8 Поколения компьютеров
Подразделять путем добавления к индексу номера поколения, например
004.032.84 Компьютеры четвертого поколения
004.04 Ориентация процесса обработки данных
004.041 процедурная
004.042 на поток данных
004.043 на структуру данных
004.045 объектная
004.046 функциональная
004.047 логическая
004.048 на реализацию искусственного интеллекта
=> 004.8 Искусственный интеллект

004.07 Характеристики памяти
004.072 Функционирование памяти
004.072.2 Чтение
004.072.3 Запись
004.072.4 Доступ
004.072.5 Адресация
004.072.6 Поблочная передача
004.074 Эффективность памяти
004.074.2 Плотность записи
004.074.3 Время доступа
004.074.32 Время позиционирования. Время установки головок
004.074.34 Время цикла памяти
=> 004.032.3 Согласование по времени. Задание времени цикла
004.076 Энергозависимость
004.076.2 Энергозависимая память

004.076.4 Энергонезависимая память
004.08 Носители вводимых и выводимых данных. Запоминающие среды
=> 621.377.6 Цифровые накопители, резисторы и запоминающие устройства

Основной ряд индексов

004.2 Архитектура вычислительных машин
004.22 Представление данных
=> 621.3.037.3 Виды представления информации
004.222 Численные данные
004.222.2 Представление чисел с фиксированной запятой (точкой)
004.222.3 Представление чисел с плавающей запятой (точкой)
004.222.5 Переполнение. Потеря значимости
004.223 Символьные и другие подобные данные
004.223.2 Алфавитно-цифровые данные
Отдельные алфавиты обозначать при помощи .003.33..., например
004.223.2:003.332.5 Представление арабской письменности
004.223.3 Графические знаки. Включая: Двухбайтовое и трехбайтовое представление
Отдельные письменности обозначать при помощи :003.32..., например
004.223.3:003.324.1 Представление китайской письменности
004.223.5 Специальные символы
004.223.6 Управляющие символы
004.223.7 Переключающие символы. Ключи перехода
004.23 Структура системы команд
004.231 Виды систем команд
004.231.2 Вычислительные машины с полной системой команд (CISC)
004.231.3 Вычислительные машины с сокращенным набором команд (RISC)
004.232 Формат команд
004.233 Виды команд
004.233.2 Команды ветвления
004.233.3 Команды обработки данных
004.233.5 Команды ввода-вывода
004.234 Регистры
004.235 Схемы адресации
004.236 Подпрограммы в системе команд
004.237 Прерывания
004.238 Состояние процесса
004.239 Защита памяти
004.25 Системы памяти
004.252 Иерархия памяти
004.254 Кэш-память
004.255 Виртуальная память

004.258 Система управления памятью
004.27 Перспективные архитектуры. Нефоннеймановские архитектуры
004.272 Архитектуры параллельной обработки
=> 004.032.24 Параллельный, одновременный режим обработки данных
004.272.2 Методы параллельной обработки
004.272.22 Конвейерное управление
004.272.23 Использование присоединенных вспомогательных процессоров
004.272.25 Векторная обработка
004.272.26 Многопроцессорная обработка
004.272.3 Архитектурные решения для параллельной обработки
004.272.32 ОКМД-архитектура (одиночный поток команд и множественный поток данных, SIMD)
004.272.33 МКОД-архитектура (множественный поток команд и одиночный поток данных, MISD)
004.272.34 МКМД-архитектура (множественный поток команд и множественный поток данных, MIMD)
004.272.4 Виды процессорных систем для параллельной обработки
004.272.42 Системы матричных процессоров
004.272.43 Многопроцессорные системы
004.272.44 Потоковые процессоры. Системы, управляемые потоком данных
004.272.45 Архитектура сети взаимодействующих процессоров
004.273 Программно-ориентированная архитектура
004.274 Динамическая архитектура
004.3 Аппаратные средства. Техническое обеспечение

Специальные определители

Здесь применяются специальные определители -1/-8 из класса 62 Инженерное дело. Техника в целом и определитель -9 из класса 66 Химическая технология, а также нижеследующие:
004.3'1 Производство вычислительных устройств
004.3'12 Принципы конструирования. Проектные соображения
004.3'122 Проектирование снижения шума. Проектирование снижения помех
004.3'124 Проектирование теплового режима. Технология охлаждения
004.3'14 Технология сборки ЭВМ. Компоновка компьютеров
004.3'142 Корпусы и конструктивное оформление
004.3'142.2 Компоновка на уровне устройств
004.3'142.22 Объединение компонентов в блоки. Крепление кристаллов на подложке
Включая: Проводное соединение. Автоматическая сборка на ленточном носителе
=> 621.792 Соединение материалов с помощью адгезии
004.3'142.23 Перевернутые кристаллы (связывание за одну операцию с металлизированными межсоединениями подложки)

004.3'142.24 Корпусы с однорядным расположением выводов. Однорядные корпусы
004.3'142.25 Корпусы с двухрядным расположением выводов. Двухрядные корпусы
004.3'142.26 Корпусы с матричным расположением выводов. Матричные корпусы
004.3'142.27 Плоские корпусы
004.3'142.4 Компоновка на уровне плат. Установка блоков на панелях
Включая: Штырьковый монтаж. Поверхностный (планарный) монтаж. Гибридная компоновка
004.3'142.6 Компоновка на уровне стойки
004.3'144 Компоненты компьютеров
Подразделять при помощи : (знак отношения), например
004.3'144:621.3.049.75 Печатные схемы компьютеров
004.3'144:621.3.049.771.15 Сверхбольшие интегральные схемы компьютеров
004.3'144:621.314 Блоки питания компьютеров
004.3'144:621.316.54 Переключатели
004.3'144:621.318.5 Реле
004.3'2 Компьютерные установки. Установка компьютеров
Например
004.3'2:692.5 Конструкция пола для установки компьютеров
004.3'2:697.9 Вентиляция и кондиционирование воздуха для компьютерных установок

Фрагмент МПК

Раздел G — ФИЗИКА

Примечание :

(1) В данном разделе термин «переменная» (как существительное) используется для определения признаков или свойств объектов (например, размера, физических условий, таких как температура, качества, такого как плотность или цвет и т. д.), которые характеризуют собой данный объект (например, предмет, качество материала, светового луча) и подлежат измерению в определенный момент времени. Переменная может изменяться по величине в зависимости от времени или других условий ее измерения, но может быть в определенных условиях или для практических целей и неизменяемой (например, длина стержня может быть принята постоянной во многих практических случаях).

(2) Следует обратить внимание на значение терминов или выражений, используемых в примечаниях к нескольким классам этого раздела, например, значение термина «измерение» в классе G01, «управление» и «регулирование» в классе G05.

(3) Классификация в данном разделе часто вызывает значительные трудности при определении сущности и функциональных признаков объекта. Эти трудности возникают в связи с возможностью использования одного и того же объекта в различных областях техники, т. е. в тех случаях, когда имеет место различие между целевыми назначениями объекта и способом его использования, кроме того, часто бывает, что объект, отнесенный к данному разделу, входит составной частью в какую-либо систему, отличающуюся иными признаками, чем перечисленные в описании объекта. Например, любая информация (в частности в форме последовательности цифр) может воспроизводиться для целей обучения и рекламы (G09), для индикации результатов измерений (G01), для дистанционной передачи или приема информации средствами сигнализации (G08). Описание измерений (G01) для дистанционной передачи или приема информации средствами сигнализации (G08). Описание цели в этом случае определяется характерными признаками, не связанными с формой каких-либо устройств. С другой стороны, устройства, реагирующие на изменение окружающих условий, например, на изменение давления текучей среды, могут быть использованы без изменения конструкции самого устройства для получения информации о давлении (G01L) или о других величинах, функционально связанных с изменением давления (другие

подклассы класса G01, например G01K для определения температуры), для регистрации наличия или отсутствия давления (G07C), подачи сигналов тревоги (G08B), управления каким-либо другим прибором (G05).

ПРИБОРЫ

G01 Измерение (счет G06M); испытание

Примечание:

(1) Кроме простых измерительных приборов в этот класс включены и другие реагирующие и записывающие устройства, а также сигнальные и управляющие устройства, поскольку они связаны с процессами измерения и не предназначены для конкретных устройств сигнализации или управления.

(2) В этом классе термин «измерение» используется в различных аспектах. В своем первоначальном значении он соответствует цифровому выражению значения переменной величины по отношению к выбранной системе измерения или по отношению к заданной переменной величине той же природы, например, выражение длины одного объекта через длину другого объекта, измерение длины посредством сопоставления со шкалой. Искомая величина может быть получена непосредственно или путем измерения какой-либо другой переменной, функционально связанной с искомой величиной, как, например, измерение температуры может быть осуществлено путем измерения длины столбика ртути. Устройство или прибор могут быть использованы:

а) для непосредственной индикации;

б) для осуществления записи или формирования сигнала, записывающего переменную величину или управляющего ею;

в) в комбинации с другими устройствами или приборами для получения общего результата измерения двух или более однородных или различных переменных величин. В связи с этим термин «измерение» в этом разделе охватывает также операции, облегчающие получение цифрового выражения путем дополнительного преобразования искомой величины в числа. Таким образом, цифровое выражение может быть получено путем представления результатов измерения в виде последовательности цифр или считывания, например со шкалы; индикация результата измерения может быть достигнута также и без использования цифр, например, с помощью учета заметных изменений в каком-либо объекте (например, в веществе, световом пучке и т. д.), связанном с измеряемой величиной (например, учет положения указывающего элемента без какой-либо шкалы, учет напряжения, генерируемого определенным образом). Часто прибегают к относительному способу измерений, т. е. к оценке совпадения или отклонения измеряемой величины (цифровое

значение которой может быть известно или неизвестно). В простейшей форме измерение может быть как простой индикацией наличия и отсутствия определенных условий или качества, например движения (в любом или в определенном направлении), так и индикацией факта превышения измеряемой величины заданного уровня.

(3) Следует обратить внимание на Примечания, следующие за заголовком класса B81 и подкласса B81B, которые относятся к «микроструктурным устройствам» и «микроструктурным системам», и на Примечания, следующие за заголовком подкласса B82B, которые относятся к «наноструктурам».

(4) Необходимо обратить внимание на примечание к разделу G, особенно на определение термина «переменная».

(5) Во многих измерительных устройствах первую измеряемую переменную преобразуют во вторую или последующие переменные. Вторая или последующие переменные могут быть:

(а) состоянием, имеющим отношение к первой переменной и получаемым в элементе, или

(б) перемещением элемента.

Может быть необходимо и дальнейшее преобразование.

При классифицировании такого устройства

(i) классифицируют стадию преобразования или каждую стадию преобразования, которая представляет интерес, либо,

(ii) если интерес заключается только в системе в целом, первую переменную классифицируют в соответствующем подклассе.

Это особенно важно, когда имеют место два или более преобразования, например, когда первую переменную, например давление, преобразуют во вторую переменную, например, оптическое свойство чувствительного элемента, и эту вторую переменную выражают с помощью третьей переменной, например, электрического эффекта. В таком случае следует обратить внимание на подклассы для преобразования первой переменной, для восприятия состояния, вызванного этой переменной, подкласс G01D для выражения измерения и наконец подкласс для всей системы, если такое имеется.

(6) Измерение изменений какой-либо величины следует относить к тем же подклассам, к которым отнесено измерение данной физической величины, например удлинений, следует классифицировать в подклассе G01B.

G02 Оптика (изготовление оптических элементов или приборов B24B, B29D 11/00, C03 или другие соответствующие подклассы или классы; материалы как таковые см. соответствующие подклассы, например C03B, C03C).

Примечание:
В данном классе термин «оптический» употребляется применительно не только к видимому свету, но также к ультрафиолетовому и инфракрасному излучению.

G03 Фотография; кинематография; аналогичное оборудование, использующее волны иные, чем оптические; электрография; голография (воспроизведение изображений или образов путем развертки и преобразования в электрический сигнал H04N)

Примечание:
В данном классе применяемым терминам придаются следующие значения:
— «запись» — фотография или какое-либо скрытое или визуализированное, длительно сохраняющееся на носителе изображение, полученное способами фотографии, электрофотографии или любыми другими способами, в которых используется распределение на носителе электрических зарядов, намагниченных точек или участков, например рисунок, образованный электрическими зарядами, записанный на носителе;
— «оптический» употребляется не только применительно к видимому свету, но также к ультрафиолетовому и инфракрасному излучениям .

G04 Часы и прочие измерители времени

G05 Управление; регулирование (специально предназначенные для определенной области применения см. соответствующие такой области участки МКИ, например A62C 37/00, B03B 13/00, B23Q)

Примечание: (1) К данному классу отнесены способы, устройства и системы общего назначения для регулирования и управления.
(2) В данном классе применяемым терминам придаются следующие значения:
— «управление» — воздействие каким-либо образом на переменную величину, например, изменение ее знака (направления) или значения (в том числе изменение ее от нулевого значения), поддержание ее постоянной, ограничение области ее изменения;
— «регулирование» — автоматическое поддержание определенного значения переменной величины или поддержание этой величины в определенном диапазоне ее значений; определенное значение или области изменения переменной величины могут быть фиксированными, изменяемыми вручную, изменяемыми во времени по заданной программе или в соответствии с изменением другой переменной величины; регулирование является одной из форм управления;
— «автоматическое регулирование» часто используется в качестве синонима термина «регулирование».
(3) Необходимо обратить внимание на примечание к разделу G, особенно на определение термина «переменная».

G06 Вычисление; счет (счетные устройства для подсчета очков при играх A63B 71/06, A63D 15/20, A63F 1/18; комбинации счетных устройств с пишущими приспособлениями B43K 29/08)

Примечание : (1) К данному классу отнесены:

— моделирующие устройства, предназначенные для математической обработки существующих или ожидаемых условий или состояний в рабочих устройствах и системах;

— моделирующие устройства в сочетании с вычислительными средствами, демонстрирующие работу машин или систем, если для них не предусмотрены специальные рубрики в других классах;

— обработка или генерация графических данных.

(2) К данному классу не относятся:

— регулирование и управление с помощью моделирующих устройств, которые в основном отнесены к G05, однако они могут быть отнесены к подклассам для регулируемых устройств данного класса;

— измерение или анализ отдельных переменных величин для получения входного сигнала для моделирующего устройства, которое отнесено к G01;

— моделирующие устройства для учебных или тренировочных целей, если они вызывают в обучающемся ощущения, идентичные действительным ощущениям, возникающим в ответ на его действия; эти устройства отнесены к G09;

— элементы моделирующих устройств, подобные рабочим устройствам или машинам, отнесенным к соответствующим подклассам (но не к G09).

(3) В данном подклассе применяемым терминам придаются следующие значения:

— «данные» — синонимичен термину «информация», поэтому термин «информация» не используется в G06C и G06F;

— «вычисление» — операции над цифрами или цифровыми данными;

— примечание переводчика: в этом пункте оригинала рассматриваются оттенки значений терминов на английском и французском языках, которые переводятся на русский язык как «вычисление»;

— «моделирующее устройство» — устройство, которое может использовать те же масштабы времени, что и рабочие устройства, или работает с увеличенным или уменьшенным масштабом времени. Модели рабочих устройств для увеличения или уменьшения масштабов времени не рассматриваются как моделирующие устройства;

— «носитель информации» — тело, например, цилиндр, диск, перфокарта, лента или проволока, способное длительное время

удерживать информацию, которая может быть считана с помощью чувствительного элемента, перемещаемого относительно записываемой информации.

(4) Необходимо обратить внимание на примечание (в частности, на определение термина «переменная») к разделу G.

G07 Контрольные устройства

G08 Сигнализация (рекламные и демонстрационные устройства как таковые G09F; передача изображений H04N)

G09 Средства обучения; тайнопись; дисплеи; рекламное и выставочное дело; печати и опечатывание

G10 Музыкальные инструменты; акустика

Примечание: (1) К этому классу отнесены все устройства для получения звука.

(2) Под термином «музыкальный инструмент», применяемым в этом классе, следует понимать любые устройства, издающие звуковые сигналы.

(3) Для удобства лиц, пользующихся МПК, в данном классе приводится его содержание без деления на подклассы. Делается это для того, чтобы показать деление по подклассам трех основных видов инструментов:

— духовых инструментов,
— струнных инструментов,
— шумовых инструментов,

которые охватывают большинство музыкальных инструментов.

(4) Имеется ряд инструментов, не относящихся к указанным в п.(3). Они относятся к группам G10D 17/00 или G10K 7/00, G10K 9/00 или G10K 15/04.

G11 Накопление информации

G12 Конструктивные элементы приборов

ЯДЕРНАЯ ФИЗИКА И ТЕХНИКА И ПРИМЫКАЮЩИЕ К НИМ ОТРАСЛИ НАУКИ

G21 Ядерная физика, ядерная техника

G11 — Накопление информации

G11B Накопление информации, основанное на относительном перемещении носителя записи и преобразователя (запись измеряемых величин способами, не требующими воспроизведения через преобразователь, G01D; светочувствительные материалы или процессы для фотографических целей G03C; электрография, электрофотография, магнитография G03G; записывающая или воспроизводящая аппаратура с использованием механически маркированной ленты, например, перфорированной бумажной ленты, или с использованием отдельных

записей, например, карточек с перфорированной или магнитной маркировкой G06K; перенос данных с носителя записи одного типа на другой G06K 1/18; печатание информации с носителя записи G06K 3/00; устройства для получения постоянного визуального представления выходных данных G06K 15/00; устройства или схемы для управления индикаторными устройствами с использованием статических средств представления меняющейся информации G09G; конструктивные элементы устройств, использующих метод сканирующего зонда, вообще G12B 21/00; кодирование, декодирование или преобразование кода вообще H03M; схемы связи выхода воспроизводящего устройства с радиоприемником H04B 1/20; запись телевизионных сигналов, H04N 5/76, H04N 9/79; громкоговорители, микрофоны, адаптеры или подобные акустические электромеханические преобразователи или схемы для них H04R)

Примечание: (1) К данному подклассу отнесены:

— запись и воспроизведение информации при относительном движении носителя информации и преобразователя, при этом глубина модуляции записи соответствует изменениям записывающего сигнала;

— конструктивные элементы для записи и воспроизведения, например головки;

— носители информации, используемые данной аппаратурой;

— устройства для согласования и соединения этой аппаратуры с другими аппаратами и устройствами.

(2) В данном подклассе применяемым терминам придаются следующие значения:

— «носитель информации» — тело в форме цилиндра, диска, карты, ленты или проволоки, способное долговременно хранить информацию, которая может быть считана с помощью чувствительного элемента, перемещаемого относительно носителя записи;

— «головка» — любое устройство для преобразования синусоидальных или несинусоидальных электрических колебаний в изменения характеристик физических свойств поверхности носителя информации и наоборот;

— «близкое взаимодействие» — взаимодействие в ближней зоне (на очень коротком расстоянии) с использованием техники сканирующего зонда, например, квазиконтакт или бесконечно малый контакт между головкой и носителем записи.

(3) Следует обратить внимание на Примечания, следующие за заголовками класса B81 и подкласса B81B, которые относятся к «микроструктурным устройствам» и «микроструктурным системам».

G11C Запоминающие устройства статического типа (накопление информации, основанное на относительном перемещении носителя записи и

преобразователя G11B; полупроводниковые приборы для запоминающих устройств H01L, например H01L 27/108-H01L 27/115; импульсная техника вообще H03K, например, электронные переключатели H03K 17/00)

Примечание: (1) К данному подклассу отнесены устройства или приспособления для хранения цифровой или аналоговой информации:

(i) без относительного движения устройства для хранения информации и преобразователя;

(ii) с устройством выбора для записывания или выдачи информации из запоминающего устройства.

(2) Элементы, не предназначенные для запоминания и не имеющие средств для хранения информации, следует классифицировать в соответствующих подклассах, например классов H01, H03K.

(3) В данном подклассе запоминающий элемент имеет устройства для записывания или выборки каждой отдельной информации.

G11B — Накопление информации, основанное на относительном перемещении носителя записи и преобразователя (запись измеряемых величин способами, не требующими воспроизведения через преобразователь, G01D; светочувствительные материалы или процессы для фотографических целей G03C; электрография, электрофотография, магнитография G03G; записывающая или воспроизводящая аппаратура с использованием механически маркированной ленты, например, перфорированной бумажной ленты, или с использованием отдельных записей, например, карточек с перфорированной или магнитной маркировкой G06K; перенос данных с носителя записи одного типа на другой G06K 1/18; печатание информации с носителя записи G06K 3/00; устройства для получения постоянного визуального представления выходных данных G06K 15/00; устройства или схемы для управления индикаторными устройствами с использованием статических средств представления меняющейся информации G09G; конструктивные элементы устройств, использующих метод сканирующего зонда, вообще G12B 21/00; кодирование, декодирование или преобразование кода вообще H03M; схемы связи выхода воспроизводящего устройства с радиоприемником H04B 1/20; запись телевизионных сигналов, H04N 5/76, H04N 9/79; громкоговорители, микрофоны, адаптеры или подобные акустические электромеханические преобразователи или схемы для них H04R)

Примечание: (1) К данному подклассу отнесены:

— запись и воспроизведение информации при относительном движении носителя информации и преобразователя, при этом глубина модуляции записи соответствует изменениям записывающего сигнала;

— аппаратура и конструктивные элементы для записи и воспроизведения, например головки;

— носители информации, используемые данной аппаратурой;

— устройства для согласования и соединения этой аппаратуры с другими аппаратами и устройствами.

(2) В данном подклассе применяемым терминам придаются следующие значения:

— «носитель информации» — тело в форме цилиндра, диска, карты, ленты или проволоки, способное долговременно хранить информацию, которая может быть считана с помощью чувствительного элемента, перемещаемого относительно носителя записи;

— «головка» — любое устройство для преобразования синусоидальных или несинусоидальных электрических колебаний в изменения характеристик физических свойств поверхности носителя информации и наоборот;

— «близкое взаимодействие» — взаимодействие в ближней зоне (на очень коротком расстоянии) с использованием техники сканирующего зонда, например, квазиконтакт или бесконечно малый контакт между головкой и носителем записи.

(3) Следует обратить внимание на Примечания, следующие за заголовками класса B81 и подкласса B81B, которые относятся к «микроструктурным устройствам» и «микроструктурным системам».

Содержание:

ЗАПИСЬ ИНФОРМАЦИИ И СРЕДСТВА ВОСПРОИЗВЕДЕНИЯ ИНФОРМАЦИИ ОДНОГО И ТОГО ЖЕ ТИПА	
Механическая	G11B 3/00
Магнитная	G11B 5/00
Оптическая	G11B 7/00
Иная, чем выше перечисленные	G11B 9/00
ЗАПИСЬ ИНФОРМАЦИИ ОДНОГО ТИПА, А СРЕДСТВА ВОСПРОИЗВЕДЕНИЯ ДРУГОГО ТИПА	G11B 11/00
ОДНОВРЕМЕННАЯ ИЛИ ВЫБОРОЧНАЯ ЗАПИСЬ РАЗЛИЧНЫХ ТИПОВ; СРЕДСТВА ДЛЯ ОДНОВРЕМЕННОГО ИЛИ ВЫБОРОЧНОГО ВОСПРОИЗЕДЕНИЯ	G11B 13/00
ОБРАБОТКА СИГНАЛА, НЕ ЗАВИСЯЩАЯ ОТ СПОСОБА ЗАПИСИ ИЛИ ВОСПРОИЗВЕДЕНИЯ	G11B 20/00
УСТРОЙСТВА, ОТЛИЧАЮЩИЕСЯ ФОРМОЙ НОСИТЕЛЯ ЗАПИСИ	G11B 25/00
КОНСТРУКТИВНЫЕ ЭЛЕМЕНТЫ; ОБЩИЕ ХАРАКТЕРИСТИКИ	
Пусковые, останавливающие и приводные устройства 15/00	G11B 19/00
Направляющие устройства	G11B 17/00

ГОЛОВКИ; НОСИТЕЛИ ЗАПИСИ	G11B 21/00,G11B 23/00
КОМБИНИРОВАННЫЕ УСТРОЙСТВА	G11B 31/00
МОНТАЖ, ИНДЕКСИРОВАНИЕ, СИНХРОНИЗАЦИЯ И КОНТРОЛЬ ЗАПИСИ	G11B 27/00
ИЗГОТОВЛЕНИЕ НОСИТЕЛЕЙ ЗАПИСИ	G11B 3/70,G11B /84, G11B 7/26
ПРОЧИЕ КОНСТРУКТИВНЫЕ ЭЛЕМЕНТЫ, ДЕТАЛИ ИЛИ ВСПОМОГАТЕЛЬНЫЕ ПРИНАДЛЕЖНОСТИ	G11B 33/00

G11B 3/00 Запись информации путем механического нарезания, деформации или прессования, например, канавок (бороздок) или углублений; воспроизведение путем механического считывания; носители информации для этого (G11B 11/00 имеет преимущество; запись путем нарезания или деформации с помощью лазерного луча G11B 7/00; с помощью электронного луча G11B 9/10)

G11B 5/00 Запись путем намагничивания или размагничивания носителя информации; воспроизведение с использованием магнитных средств; носители информации для этого (G11B 11/00 имеет преимущество)

Примечание:
Рубрики G11B 5/02-G11B 5/86 имеют преимущество перед рубриками G11B 5/004-G11B 5/016.

G11B 7/00 Запись или воспроизведение с помощью оптических средств, например, запись с использованием теплового луча оптического излучения, воспроизведение с использованием оптического луча при более низкой мощности; носители записи для этого (G11B 11/00, G11B 13/00 имеют преимущество)

G11B 9/00 Запись или воспроизведение с помощью способов или средств, не отнесенных ни к одной из групп в интервале G11B 3/00-G11B 7/00; носители записи для этих целей (G11B 11/00 имеет преимущество; конструктивные элементы устройства сканирующего зонда вообще G12B 21/00)

Примечание:
В данной группе рубрика G11B 9/12 имеет преимущество перед рубриками G11B 9/02-G11B 9/10.

G11B 11/00 Запись или воспроизведение с одного и того же носителя записи в случае, когда для этих двух операций используются способы или средства, охватываемые различными главными группами в интервале G11B 3/00-G11B 7/00 или различными подгруппами группы G11B 9/00; носители записи для этих целей

Примечание:
В данной группе рубрика G11B 11/24 имеет преимущество перед рубриками G11B 11/03-G11B 11/16.

G11B 13/00 Одновременная или выборочная запись способами или средствами, отнесенными к различным рубрикам; носители записи для этого; одновременное или выборочное воспроизведение

Примечание:
(1) Данная группа охватывает устройства по крайней мере с двумя видами записи информации, выполненной двумя различными способами или средствами, или использующие два разных физических свойства в одном и том же или разных местах на одном и том же носителе записи, причем записи создаются или воспроизводятся одновременно или выборочно.
(2) Если такие комбинации средств используются для изменения только одного основного свойства, классифицирование производится только в одной из соответствующих основных групп G11B 3/00, G11B 5/00, G11B 7/00, G11B 9/00 или G11B 11/00.

G11B 15/00 Привод, пуск или остановка носителей записи, выполненных в форме нитей или ленты; привод как носителей записи, так и головок; направляющие устройства для таких носителей записи или контейнеры для них; управление такими устройствами; управление режимами работы (приводные или направляющие головки G11B 3/00-G11B 7/00, G11B 21/00)

G11B 17/00 Направляющие средства для носителей записи, отличающихся по форме от нити или ленты, или не зависящие от опор для носителей (направляющие средства для карт или листов G06K 13/00)

G11B 19/00 Привод, пуск, остановка носителей записи, отличающихся по форме от нити или ленты, или опор этих носителей; управление ими; управление режимами работы (направляющие устройства для таких носителей записи G11B 17/00)

G11B 20/00 Обработка сигнала, не зависящая от способа записи или воспроизведения; схемы для них

G11B 21/00 Расположение головок, не зависящее от способа записи или воспроизведения

G11B 23/00 Носители записи, не зависящие от способа записи или воспроизведения; вспомогательные принадлежности, например, контейнеры, специально приспособленные для совместной работы с записывающей или воспроизводящей аппаратурой

Примечание:
В группе G11B 23/00 записывающие или воспроизводящие устройства не включают носителей записи [5].

G11B 25/00 Аппаратура, отличающаяся формой используемого носителя записи, но не зависящая от способа записи или воспроизведения (отдельные узлы аппаратуры G11B 3/00-G11B 23/00, G11B 33/00)

G11B 27/00 Монтаж; индексация; адресация; хронирование или синхронизация; контроль; измерение движения ленты

G11B 31/00 Устройства для совместной работы записывающей или воспроизводящей аппаратуры с аппаратурой, имеющей родственное функциональное назначение (с кинопроекционными камерами G03B 31/00; оборудование подстанции для записи телефонных разговоров или сообщений при отсутствии абонентов H04M 1/65)

G11B 33/00 Конструктивные элементы, детали или вспомогательные принадлежности, не предусмотренные в предшествующих группах (тара, упаковочные материалы или упаковки, специально приспособленные для носителей записи B65D 85/00)

G11B 7/00 — Запись или воспроизведение с помощью оптических средств, например, запись с использованием теплового луча оптического излучения, воспроизведение с использованием оптического луча при более низкой мощности; носители записи для этого (G11B 11/00, G11B 13/00 имеют преимущество)

7/002 . системы записи, воспроизведения или стирания, характеризуемые формой носителя информации

7/0025 . . с цилиндрами или носителями записи в форме, подобной цилиндру, например, в форме усеченного конуса

7/003 . . с рулонами, например, лентами, магнитными лентами на катушке или пленками неопределенной длины

7/0033 . . с перфокартами

7/0037 . . с дисками

7/004 . способы записи, воспроизведения или стирания; используемые схемы считывания, записи или стирания

7/0045 . . запись (G11B 7/006, G11B 7/0065 имеют преимущество)

7/005 . . воспроизведение (G11B 7/0065 имеет преимущество)

7/0055 . . стирание (G11B 7/006, G11B 7/0065 имеет преимущество)

7/006 . . перезапись (G11B 7/0065 имеет преимущество)

7/0065 . . запись, воспроизведение или стирание с использованием изображений (образцов) оптической интерференции, например голограмм

7/007 . расположение информации на носителе записи, например, форма дорожек

7/013 . . для дискретной информации, т. е. когда каждая единица информации хранится в отдельном положении

7/08 . расположение или установка головок и(или) источников света относительно носителей информации

7/085 . . с обеспечением перемещения светового луча в его рабочее положение или для выхода из него (модулирование с помощью информационных сигналов G11B 7/12; управление положением или направлением световых лучей, т. е. отклонение G02F 1/29)

7/09 . . с обеспечением перемещения светового луча или фокальной плоскости с целью поддержания положения светового луча относительно носителя записи во время работы, например, для компенсации неоднородности поверхности носителя или для отслеживания дорожки

7/095 . . . специально приспособленные для дисков, например, для компенсации эксцентриситета или биений

7/10 . . разъемные крепления головок без повторной регулировки

7/12 . головки

7/125 . . источники оптического луча для этих целей (светоизлучающие диоды H01L 33/00, полупроводниковые лазеры H01S 5/00); модуляторы, т. е. средства для управления размерами или интенсивностью оптического пятна или оптического следа (электро-, магнито- или акустооптические модуляторы G02F 1/00; оптические диафрагмы G03B 9/02)

7/13 . . оптические детекторы для этих целей (оптические детекторы вообще G01J; демодуляция света, перенос модуляции модулированного света, изменение частоты света G02F 2/00)

7/135 . . средства для направления луча от источника к носителю записи или от носителя к детектору

7/14 . . приспособленные для записи и воспроизведения на нескольких дорожках одновременно (G11B 7/20 имеет преимущество)

7/16 . с использованием фильтров, например светофильтров

7/18 . . с использованием оптических щелей

7/20 . . с устройствами двойной записи, т. е. головки, в которых информация записывается в двух различных формах одновременно на той же самой или связанных дорожках, например, запись мгновенных и средних величин (запись звука на кинопленку G03C)

7/22 . . способы и устройства для изготовления головок, например, для сборки

7/24 . носители записи, отличающиеся материалом или структурой, или формой (отличающиеся расположением информации на носителе G11B 7/007; светочувствительные материалы как таковые G03C)

7/26 . . способы и устройства для изготовления носителей информации (способы, предусматривающие применение отдельных технологических операций, см. в соответствующих классах, например B29, G03)

7/28 . перезапись, т. е. перенос информации с одного оптического носителя на другой или на несколько подобных или другого типа носителей с использованием оптических воспринимающих средств

7/30 . перезаписываемые носители информации (G11B 7/24 имеет преимущество)

Фрагмент классификации наук ГРНТИ

20 ИНФОРМАТИКА
УДК 002
ВАК 05.25.00; 05.13.17

Примечание. Информационная деятельность в отдельных областях науки, техники и отраслях экономики отражается в соответствующих разделах с окончанием кода XX.01.29

20.01 Общие вопросы информатики
УДК 002
ВАК 05.25.00; 05.13.17
20.01.01 Руководящие материалы
УДК 002(094)
ВАК 05.25.00; 05.13.17
20.01.04 Информатизация общества. Информационная политика
УДК 002; 002:338.2
ВАК 05.25.00; 05.13.17
См. также 12.41 Организация науки. Политика в области науки
Отс. от 26.11 Глобальные проблемы
20.01.07 Теория и методология информатики
УДК 002.001; 002:001.8
ВАК 05.13.17
20.01.09 История информатики и информационной деятельности. Персоналия
УДК 002(091); 002(092)
ВАК 05.25.00; 05.13.17; 07.00.10
20.01.13 Научные и технические общества, конгрессы, конференции, симпозиумы, семинары, выставки
УДК 002:061.2/.4
ВАК 05.25.00; 05.13.17; 05.26.05
20.01.17 Международное сотрудничество, деятельность международных организаций по информатике
УДК 002+02].009(100); 002.61
ВАК 05.25.00; 05.13.17
20.01.33 Терминология информатики. Справочная литература. Учебная литература
УДК 002:001.4; 002(03); 002(075)
ВАК 05.25.00; 05.13.17; +13.00.02

20.01.37 Стандартизация в научно-информационной деятельности
УДК 002+02]:006
ВАК 05.25.05, 08.00.05
20.01.45 Преподавание информатики
УДК 002:372.8
ВАК 05.25.00; 05.13.17; 13.00.02
20.01.79 Кадры
УДК 002.6.08
ВАК 05.25.00; 05.13.17; 13.00.02
20.01.80 Правовые вопросы
УДК 002:34
ВАК 05.25.00; 05.13.17; 12.00.03
20.15 Организация информационной деятельности
УДК 002.6; 021
ВАК 05.25.00, 08.00.05
(Развитие рубрики введено с 1995 г.)
Библиотечное дело
см. 13.31 Библиотечное дело. Библиотековедение
Архивное дело
см. 13.71 Архивное дело. Архивоведение
См. также 12.41 Организация науки. Политика в области науки
20.15.05 Информационные службы, сети, системы в целом
УДК 002.6
20.15.06 Наднациональные и международные органы информации
УДК 002.61
20.15.07 Национальные органы информации
УДК 002.63
20.15.09 Отраслевые и ведомственные органы информации
УДК 002.63
20.15.11 Региональные, локальные органы информации
УДК 002.63
20.15.13 Информационные службы на предприятиях и в учреждениях
УДК 002.66
20.15.31 Научные и технические библиотеки и библиотечные сети
УДК 026
См. также 13.31 Библиотечное дело. Библиотековедение
20.15.71 Архивы, службы перевода и др. информационные органы
УДК 930.25; 651.927.07
Издательские организации
см. 19.51.61 Издательское дело
Распространение книжной продукции
см. 19.51.65 Книжная торговля. Пропаганда и распространение печати
См. также 13.71 Архивное дело. Архивоведение

20.17 Документальные источники информации
ВАК +05.25.02; +05.25.03; +05.25.04
20.17.01 Общие вопросы
УДК 002.2
20.17.15 Виды источников информации
УДК 002.2
20.17.17 Комплектование, учет и хранение фондов источников информации
УДК 002.52/.59
20.19 Аналитико-синтетическая переработка документальных источников информации
УДК 002.53/.55
ВАК 05.25.05
Вопросы автоматического перевода текста
см. 16.31.21 Автоматическая обработка текста. Автоматический перевод. Автоматическое распознавание речи
См. также 13.41 Библиография. Библиографоведение
20.19.01 Общие вопросы
УДК 002.53/.55
ВАК 05.25.05
20.19.15 Библиографическое описание источников информации
УДК 025.32
ВАК 05.25.05, 05.25.03
20.19.17 Предметизация и индексирование
УДК 025.4.025
ВАК 05.25.05, 05.25.03
20.19.19 Аннотирование и реферирование
УДК 002.53/.55:001.814
ВАК 05.25.05
20.19.21 Составление обзоров
УДК 002.53/.55:001.891.32
ВАК 05.25.05
20.19.23 Перевод научных текстов
УДК 651.926
ВАК 05.25.05, 10.02.19
20.19.27 Автоматизация знаковой обработки текста
УДК 002.53.(084); 002.53.(086); 004.91
ВАК 05.25.05, +05.11.18
20.19.29 Обработка изобразительных и аудиовизуальных документов
(Введено с 1998 г.)
20.23 Информационный поиск
УДК 025.4.03
ВАК 05.25.05
20.23.01 Общие вопросы
УДК 025.4.03

20.23.15 Информационно-поисковые языки
УДК 025.4
См. также 16.21.47 Лексикология. Терминоведение
16.31.31 Информационные и формализованные языки
20.23.17 Информационно-поисковые массивы. Базы данных. Манипулирование данными и файлами
УДК 002.53; 002.53:004.65; 002.53:004.62/.63
(Содержание уточнено с 1998 г.)
20.23.19 Процессы информационного поиска
УДК 025.4.03
20.23.21 Информационно-поисковые системы. Банки данных
УДК 025.4.03; 002.53:004.65
20.23.25 Информационные системы с базами знаний
УДК 002.53:004.89
См. также 28.23.35 Экспертные системы
20.51 Информационное обслуживание
УДК 002.55; 024
ВАК 05.25.05
См. также 12.41.55 Информационное обеспечение научной деятельности
20.51.01 Общие вопросы
УДК 002.55; 024
20.51.15 Потребители информации
УДК 002-052
20.51.17 Информационные потребности и запросы
УДК 002.009.7:330.163; 025.4.03
20.51.19 Виды информационного обслуживания
УДК 002.55
20.51.21 Научно-техническая пропаганда
УДК 001.92
20.51.23 Эффективность информационного обслуживания
УДК 002.55.003.13
20.53 Технические средства обеспечения информационных процессов
УДК 002+02].002.5; 002:004.8; 002:004
ВАК 05.25.05; 05.13.14
См. также 13.20.31 Техническое оснащение библиотек
20.53.01 Общие вопросы
УДК 002+02].002.5; 002:004
20.53.15 Средства ввода информации
УДК 004.35
20.53.17 Средства хранения информации
УДК 004.33.07/.08; 004.33; 004.08
20.53.19 Средства обработки и поиска информации
УДК 002.5:004

20.53.21 Средства выдачи информации
УДК 004.35; 004.08
20.53.23 Средства передачи информации
УДК 004.71; 621.39.002.5
20.53.25 Средства копирования информационных материалов
УДК 002.513.3:681.621.12
20.53.27 Средства тиражного размножения информационных материалов
УДК 681.6
20.53.29 Средства микрофильмирования информационных материалов
УДК 778.14
20.53.31 Средства оргтехники в научно-информационной деятельности
УДК 002:651
20.53.33 Здания информационных центров и их оборудование
УДК 002.6.006.002.5; 022

Фрагмент тезауруса INIS

АБСОРБЦИОННАЯ СПЕКТРОСКОПИЯ
UF атомная абсорбционная спектроскопия
UF колориметрия
SF спектрохимия
BT1 спектроскопия
RT инфракрасные спектры
RT лазерная спектроскопия
RT методы двойного резонанса
RT поглощение
RT спектры крайнего ультрафиолетового излучения
RT спектры поглощения
RT структурный химический анализ
RT ультрафиолетовые спектры
RT фотоакустические спектрометры

ПОГЛОЩЕНИЕ
UF торможение (в веществе)
BT1 сорбция
NT1 k-поглощение
NT1 всасывание в кишечнике
NT1 поглощение корнями
NT1 поглощение полярной шапкой
NT1 поглощение через кожу
NT1 поглощение энергии
NT1 резонансное поглощение
NT1 самопоглощение
RT абсорбционная спектроскопия
RT абсорбционный цикл охлаждения
RT гетерогенные эффекты
RT замедление
RT излучения
RT пробег
RT пропускание
RT самоэкранирование
RT слой половинного поглощения
RT спектры поглощения
RT тормозная способность

RT точечные ядра
RT экранирование

ТЕМПЕРАТУРА АБСОЛЮТНОГО НУЛЯ
UF температура абсолютного нуля
RT диапазон температуры
RT криогенная техника

ЯДЕРНОЕ ОРУЖИЕ
UF атомное оружие
UF атомные бомбы
UF термоядерное оружие
UF ядерное нападение
SF проект тамблер
RT гражданская оборона
RT договор паротонга
RT договор тлателолко
RT испытательный полигон шт. Невада
RT локальные выпадения
RT манхэттенский проект
RT Нагасаки
RT национальная оборона
RT политика нераспространения ядерного оружия
RT проект кастл
RT проект пламббоб
RT проект редвинг
RT проект типот
RT противоракетная оборона
RT радиоактивные выпадения
RT снаряды
RT убежища
RT Хиросима
RT ядерная зима
RT ядерное разоружение
RT ядерные взрывы

Описание языка запросов ИАС xIRBIS

Предложение запроса — это структурная единица *Запроса*. В нотации Бэкуса-Наура *Предложение запроса* имеет следующий синтаксис:

```
<Предложение запроса> ::= <Условие поиска> |
<Предложение запроса><Логическая операция><Предложение
запроса>|
(<Предложение запроса>)
<Логическая операция> ::= И | AND | ИЛИ | OR | , | НЕ | NOT | ^
```

Предложение запроса в общем случае состоит из произвольного числа *Условий поиска*, связанных логическими операциями И (AND, «пробел»), ИЛИ (OR, «,») и НЕ (NOT, «^»). Внутри предложения допускается использование скобок, задающих приоритеты выполнения условий поиска.

Условие поиска устанавливает критерии соответствия поисковых дескрипторов запроса некоторой области поиска, представляющей собой совокупность структурных единиц документа — полей:

```
<Условие поиска> ::=
<Область поиска><Оператор критерия><Выражение условия> |
<Результат поиска>
```

Область поиска внутри документа задается именем отдельного поля или логическим выражением, объединяющим имена нескольких полей.

Выражение условия — набор терминов (поисковых дескрипторов) или идентификаторов результатов поиска, объединенных с помощью булевых или контекстных операторов в логическое выражение.

Оператор критерия задает условие включения или сравнения дескрипторов запроса и терминов, содержащихся в указанных полях документов.

В простейшем случае условие поиска состоит из имени поля, оператора вхождения и одного дескриптора, например:

KW: РОССИЯ

Область поиска задается именами структурных единиц документа — полей:

```
<Область поиска> ::= <Имя поля>|
(<Область поиска> <Логическая операция> <Область поиска>)
```

Из нотации видно, что допускается использование логических операций[1] при формировании области поиска. Например:

(AB OR TI): (РОССИЯ NOT *СССР)*

означает, что в результат поиска включаются все документы, в которых хотя бы в одном из заданных полей (или в обоих) встречается дескриптор РОССИЯ, но не встречается дескриптор СССР.

Если в условии поиска область поиска явно не задана, то поиск проводится в области, заданной «по умолчанию». Область поиска «по умолчанию» задается обычно либо средствами описания документа (схемой), либо параметрами интерфейсных форм построения запроса.

Оператор критерия. Для связи области поиска с выражением условия используются оператор вхождения или один из операторов сравнения.

Оператор вхождения («:») позволяет найти документы, которые в указанной области содержат термины в сочетании, указанном выражением условия;

Операторы сравнения (**= | EQ | <> | NE | > | GT | >= | GE | < | LT | <= | LE**) позволяют найти документы, значения указанного поля которых соотносятся со значением, указанным в правой части выражения условия согласно оператору. Например, условие **DT < 2000** позволит отобрать документы, дата публикации которых (поле DT) имеет значение меньше «2000». Существенно знать тип поля, так как, например, цифровые поля могут сравниваться как по правилам числовых, так и символьных данных.

Выражение условия. Синтаксис выражения условия в языке запросов следующий:

```
<Выражение условия> ::= <Дескриптор> |
<Выражение условия> <Операция> <Выражение условия> |
      (<Выражение условия> <Операция> <Выражение условия>)
<Операция> ::= <Логическая операция> | <Контекстная операция>
<Контекстная операция> ::=
CTX|CTX[N]|+|NEAR|NEAR[N]|SENT|CON[N]
```

При использовании в запросе нескольких дескрипторов они должны быть связаны контекстными или логическими операторами, а для указания приоритета выполнения операций — помещены в круглые скобки.

Приведем описание основных логических операций, примеры их использования и графическую интерпретацию (результат операции — затемненная область):

• Логическая операция **OR (ИЛИ)**

Например:

KW:('ЯДЕРНЫЙ РЕАКТОР' OR *'ЯДЕРНЫЙ ЦИКЛ')*

означает, что в результаты поиска включаются все документы, в которых в

[1] Не все ИПЯ допускают использование логических операций для указания области поиска

поле *KW* встречаются термины (словосочетания) «ЯДЕРНЫЙ РЕАКТОР» или «ЯДЕРНЫЙ ЦИКЛ», или оба вместе

• Логическая операция **AND (И)**

Например:

KW:('ЯДЕРНЫЙ РЕАКТОР' AND *'ЯДЕРНЫЙ ЦИКЛ')*

означает, что в результаты поиска включаются только те документы, в которых в поле **KW** встречаются оба термина «ЯДЕРНЫЙ РЕАКТОР» и «ЯДЕРНЫЙ ЦИКЛ»

• Логическая операция **NOT (НЕ)**

Например:

KW:(ЯДЕРНЫЙ РЕАКТОР' NOT *'ЯДЕРНЫЙ ЦИКЛ')*

означает, что в результаты поиска включаются документы, в которых в поле **KW** встречается термин «ЯДЕРНЫЙ РЕАКТОР» и не встречается «ЯДЕРНЫЙ ЦИКЛ»

Контекстные операторы. К этой группе операторов относятся:

- оператор расстояния (NEAR[N]);
- оператор расстояния со строгим следованием (CTX[N]);
- оператор предложения (SENT);
- оператор пересечения полей (CON[N]).

Параметр *N* в операторах NEAR и CTX может принимать значения от 0 до 255 (по умолчанию *N* равно 0). Отсутствие параметра означает следование терминов в поле непосредственно друг за другом (идентично значению 0).

Оператор NEAR позволяет найти документы, в заданной области поиска которых в одном предложении присутствуют поисковые дескрипторы на расстоянии *N* слов друг от друга (в произвольном порядке). Выражение условия имеет вид:

```
<дескриптор1> NEAR[N] <дескриптор2>
```

Оператор CTX позволяет найти документы, в заданной области поиска которых в одном предложении присутствуют поисковые дескрипторы, расположенные в указанном порядке на расстоянии не более *N* слов друг от друга. Выражение условия имеет вид:

```
<дескриптор1> CTX[N] <дескриптор2>
```

Оператор SENT позволяет найти документы, в заданной области поиска которых поисковые дескрипторы находятся в одном предложении. Выражение условия имеет вид:

```
<дескриптор1> SENT <дескриптор2>
```

При задании поисковых дескрипторов допускается использование операторов (символов) маскирования, средств нормализации и ссылок на ранее полученные результаты поиска.

Маскирование реализует функции двух видов:

- маскирование (или замена) произвольного числа рядом стоящих символов дескриптора (оператор маскирования — символы «*» или «$»);
- маскирование одного (непустого) символа дескриптора (оператор маскирования — символ «%»).

Символы маскирования могут использоваться вместо любого символа дескриптора, и их количество внутри дескриптора не ограничено. Параметризированные операторы маскирования ***(N)**, означают, что в дескрипторе на месте символа маскирования может стоять произвольная последовательность длиной не более *N* символов (где N от 0 до 255[1]).

Использование ранее полученных результатов поиска. В качестве операнда условия поиска в предложении запроса может использоваться ранее полученный *результат поиска*:

```
<Результат поиска> ::= # <Идентификатор результата поиска>
```

Для включения в предложение поискового запроса результатов ранее проведенного поиска используются ссылки на номер предложения в текущем запросе.

Например, запрос может иметь вид:

```
#2 AND ((KW OR AB) : Россия)
```

где #2 — ссылка на результат второго предложения запроса.

Символ «#» является индикатором ссылки. За ним указывается номер одного из предыдущих предложений текущего запроса или имя сохраненного запроса, результат поиска по последнему предложению которого используется для уточнения в этом предложении.

[1] Но, естественно, не более чем длина термина.

Оглавление

Голицына Ольга Леонидовна
Максимов Николай Вениаминович
Попов Игорь Иванович

Информационные системы и технологии

Учебное издание

Издание не подлежит маркировке
в соответствии с п. 1 ч. 1 ст. 11 ФЗ № 436-ФЗ

Выпускающий редактор *Г.Г. Семенова*
Корректор *Н.Б. Вторушина*
Компьютерная верстка *И.В. Кондратьевой*
Оформление серии *Л. Зарецкой*

Подписано в печать 20.01.2014. Формат 60×90/16.
Гарнитура «Таймс». Усл. печ. л. 25,0. Уч.-изд. л. 25,8.
Печать офсетная. Бумага офсетная. Тираж 500 экз.
Заказ №601

Издательство «ФОРУМ»
101990, Москва — Центр, Колпачный пер., д. 9а
Тел./факс: (495) 625-32-07, 625-52-43
E-mail: forum-knigi@mail.ru

Отдел продаж издательства «ФОРУМ»:

101990, Москва — Центр, Колпачный пер., д. 9а
Тел./факс: (495) 625-52-43
E-mail: forum-ir@mail.ru
www.forum-books.ru

Книги издательства «ФОРУМ»
вы также можете приобрести:

Отдел продаж «ИНФРА-М»
127282, Москва, ул. Полярная, д. 31в
Тел.: (495) 380-05-40 (доб. 252)
Факс: (495) 363-92-12

Отдел «Книга-почтой»
E-mail: podpiska@infra-m.ru;
books@infra-m.ru

Отпечатано с готовых файлов заказчика
в ОАО «Первая Образцовая типография»,
филиал «УЛЬЯНОВСКИЙ ДОМ ПЕЧАТИ»
432980, г. Ульяновск, ул. Гончарова, 14