

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

Д.Н. Борисов

КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ
СИСТЕМЫ

Учебно-методическое пособие для вузов

Издательско-полиграфический центр
Воронежского государственного университета
2007

Утверждено научно-методическим советом факультета компьютерных наук 21 марта 2007 г., протокол № 3

Рецензент И.В. Илларионов

Учебно-методическое пособие подготовлено на кафедре информационных систем факультета компьютерных наук Воронежского государственного университета.

Учебное пособие предназначено для использования на лабораторных занятиях и для самостоятельной работы студентов 5-го курса вечернего отделения факультета компьютерных наук по курсу «Корпоративные информационные системы».

Для специальности 230201 (071900) – Информационные системы и технологии

ВВЕДЕНИЕ

На пороге третьего тысячелетия человечество столкнулось с высоким темпом развития информационных технологий. Процесс информатизации затрагивает в настоящее время практически все сферы человеческой деятельности – управление, науку, производство, образование и т. д. Создаваемые информационные системы (ИС) призваны, прежде всего, повысить эффективность работы человека, помочь ему в быстром нахождении эффективных решений в различных отраслях деятельности.

Средства управления информацией и достижения отлаженной организации работы уже давно являются необходимым условием успешного функционирования корпоративных систем. Однако разработка и внедрение современных корпоративных ИС связана, как правило, с серьезным риском. Этот риск связан с тем, что наиболее непредсказуемым фактором создания решений в области информационных технологий является человеческий фактор. Поэтому в современной информационной индустрии очень важны индивидуальный подход, а также использование новых технологий и методологий, позволяющих повысить эффективность использования разрабатываемых ИС.

1. Корпоративные информационные системы

1.1. Основные задачи корпоративной ИС

Корпоративную информационную систему (КИС) можно определить как совокупность информационных ресурсов, процессов и технологий, собирающих, преобразующих и распространяющих корпоративную информацию. **Обобщенной целью** корпоративной ИС является накопление, хранение и преобразование информации для использования ее в процессе принятия управленческих решений. Современные КИС обладают сложной, зачастую гетерогенной структурой и предназначены для решения большого количества разнотипных задач автоматизации, возникающих на предприятии.

Структура и функциональность корпоративных ИС, как правило, определяется характером организации источников и потребителей информации ИС, среди которых могут быть как пользователи, так и другие ИС.

Основными задачами корпоративной ИС являются поддержка динамической информационной модели предметной области и обеспечение решения на основе информационной модели задач управленческого, исследовательского, конструкторского и иного характера. Для успешного решения этих задач корпоративная ИС должна обладать следующими функциями:

1. Функции сбора и регистрации информационных ресурсов. Сбор информации о предметной области необходим для поддержания

информационной модели в адекватном состоянии. Возможны следующие варианты реализации данных функций:

а) путем измерений или наблюдений фактов в реальном мире и ввода данных в систему вручную с помощью клавиатуры и/или каких-либо манипуляторов. Например, ввод данных после социологического опроса населения;

б) полуавтоматически, путем ввода данных в компьютер с некоторых носителей. Например, сканирование паспортных данных, их распознавание и ввод в базу данных;

в) автоматически, с помощью различного рода датчиков или механизмов обмена данными с другими ИС. Например, определение температуры, влажности, содержание углекислого газа в воздухе, выявление IP адресов с которых производились атаки сетевого периметра и др.

При реализации функций сбора и регистрации информационных ресурсов в системе, как правило, необходимо решить следующие задачи:

а) задача фильтрации данных. Как правило, информация поступает в ИС из разных источников данных и поэтому зачастую избыточна. В результате фильтрации производится отбор нужных данных из множества поступающих в систему;

б) задача верификации данных. Верификация данных обеспечивает достоверность и логическую целостность данных. Проверка достоверности данных осуществляется на основе экспертного анализа вводимой информации на предмет адекватности. Проверка логической целостности данных осуществляется на стадии предварительной обработки информации с помощью средств логического контроля (т. е. в контексте используются логические связи поступающих данных);

в) задача сжатия данных. Сжатие данных осуществляется с целью минимизации ресурсов памяти, необходимых для их хранения, а также для снижения затрат на передачу данных по коммуникационным каналам;

г) задача преобразования данных из одного формата в другой. Эта задача возникает при необходимости передачи данных из одной информационной системы в другую, либо при передаче данных между различными частями системы.

2. Функции хранения информационных ресурсов. Данные функции связаны, прежде всего, с необходимостью управления ресурсами хранимых данных и ресурсами памяти. Требования к этим функциям различаются в разных классах информационных систем. Наиболее часто организация хранения данных осуществляется на основе файловой системы, при этом логической единицей информации является файл.

В качестве альтернативы файловым системам можно использовать механизмы управления данными, основанные на технологиях баз данных.

3. Функции актуализации информационных ресурсов. Актуализация

информационных ресурсов заключается в приведении их в соответствие текущему состоянию предметной области системы. Это осуществляется путем добавления новой или удаления устаревшей информации, а также изменения связей между информационными единицами.

Процесс актуализации информационных ресурсов в корпоративных ИС обладает дискретным характером, поэтому адекватность состояния модели предметной области и ее состояния в реальности обеспечивается с временным лагом, величина которого зависит от установленного регламента актуализации информационных ресурсов.

4. Функции обработки информационных ресурсов. Одним из важнейших качеств корпоративных ИС является возможность производства новых данных и знаний на основе уже существующих. Для анализа существующих данных и выявления закономерностей в современных корпоративных ИС используется инструментарий различного рода. Наиболее часто в процессе обработки данных используются методы статистического анализа, однако в последнее время все более широкое распространение получают приложения искусственного интеллекта – нейронные сети, нечеткая логика и т. д.

5. Функции предоставления информационных ресурсов пользователям. Целью создания корпоративной ИС является, прежде всего, удовлетворение информационных потребностей пользователей, поэтому функции обеспечения интерфейса системы с пользователем являются одной из важнейших составляющих ИС. Предоставление информационных ресурсов пользователю может осуществляться с использованием двух технологий:

а) pull-технологий. В этом случае инициатором процесса предоставления информационных ресурсов является пользователь. Для поддержки этого процесса в системе предусматриваются пользовательские интерфейсы – средства взаимодействия пользователей с ИС. Существуют следующие виды интерфейсов, обеспечивающих взаимодействие конечных пользователей с ИС:

- интерфейсы командной строки;
- графические интерфейсы.

В качестве примера использования данной технологии рассмотрим предоставление провайдером информационных (мультимедийных) услуг. Пользователь получает какую-либо информацию, предварительно послав запрос по линии через местного провайдера интернет или провайдера мультимедийных услуг. Затребованные данные пересылаются по линии связи. Типичные случаи применения такой технологии – просмотр Интернет сайтов, передача программного обеспечения, on-line газет и т. д.

Если в качестве пользователя корпоративной ИС выступает не человек, а компонент прикладного программного обеспечения, доступ к системе осуществляется с помощью интерфейсов прикладного

программирования – Application Programming Interface;

б) push-технологии. В этом случае инициатором предоставления информационных ресурсов является сама информационная система согласно определенному регламенту. В соответствии с регламентом для какого-либо рода информационных ресурсов определяется круг зарегистрированных пользователей, после чего распространение информации осуществляется путем рассылки сообщений всем подписчикам.

Push-технологии рассмотрим на том же примере предоставления провайдером информационных услуг. Провайдер передает информацию всем подписчикам или определенной части подписчиков. Передача данных группируется провайдерами услуг в соответствии с потребностями абонентов, и обратный канал не требуется. Типичные случаи применения: передача прогноза погоды, финансовых и спортивных новостей, газет и бюллетеней, а также передача данных, аудио, видео, игр и т. д. Передача данных в некоторых случаях возможно совместить с теле- или радиосигналами.

1.2. Классификация информационных систем

В современном мире существует достаточно большое количество разновидностей информационных систем. Классификация информационных систем обычно осуществляется на основе каких-либо выделенных признаков. Например, с точки зрения управленческого уровня, на котором осуществляется использование ИС, принято делить корпоративные ИС на следующие виды (рис. 1.1):

1. ИС для обеспечения текущих бизнес-операций.
2. ИС для поддержки процесса принятия решений.
3. ИС для обеспечения стратегических преимуществ.



Рис. 1.1. Классификация корпоративных ИС с точки зрения управленческого уровня

1. **ИС для обеспечения текущих операций** предназначены для решения задач оперативного учета и контроля бизнес-процессов предприятия и используются на уровне работников предприятия и их непосредственных начальников. Существуют следующие разновидности таких ИС:

1.1. Системы обработки операций, выполняющие поддержку финансовых операций, осуществляющие модернизацию баз данных, обработку данных, формирование отчетов и прочих документов.

1.2. Системы управления процессами, управляющие физическими процессами производства продукции.

1.3. Системы автоматизации офиса, автоматизирующие коммуникации, делопроизводство и производительность офиса, охватывающие обработку текстов, электронную почту, организацию телеконференций, обработку и хранение документов.

2. **Системы поддержки процесса принятия решений** также имеют три разновидности:

2.1. Системы предоставления информации, предоставляющие менеджерам predetermined и регламентированные сообщения и отчеты о текущих бизнес-операциях.

2.2. Системы поддержки принятия решений, имеющие в своем составе набор диалоговых и специальных средств проектирования альтернативных решений для использования в непрограммируемых ситуациях.

2.3. ИС руководителей, обслуживающие менеджеров высшего уровня с целью мгновенного формирования критической информации в свободном формате. Информировать о состоянии предприятия по ключевым факторам.

3. **ИС для обеспечения стратегических преимуществ** дают мгновенный доступ к информации о важнейших факторах, влияющих на достижение фирмой своих задач. ИС этого типа должны производить такие информационные продукты и услуги, которые содействуют привлечению клиентов и поставщиков, а также увеличению объема продаж за счет ресурсов конкурентов.

Следует отметить, что несмотря на выделение некоторого количества видов ИС, реальные системы достаточно часто воплощают в себе несколько концептуальных типов ИС. Современные концепции ИС обусловлены пониманием функций, которые выполняют ИС в современных организациях, анализом роли ИС в обеспечении массовых текущих операций, а также влиянием ИС на достижение стратегических преимуществ в конкурентной борьбе.

Помимо классификации с точки зрения управленческого уровня, можно *классифицировать* корпоративные ИС на основе следующих характеристик:

1) предметная область системы;

- 2) вид поддерживаемых информационных ресурсов;
- 3) функции обработки информационных ресурсов;
- 4) степень детализации предметной области;
- 5) среда хранения информационных ресурсов;
- 6) объем информационных ресурсов;
- 7) степень динамичности информационных ресурсов;
- 8) состав лингвистических ресурсов системы;
- 9) архитектура системы;
- 10) регламент обслуживания пользователей системы;
- 11) расписание функционирования системы;
- 12) способы и характер доступа к системе;
- 13) поддерживаемые стандарты информационных технологий;
- 14) реализуемые интерфейсы;
- 15) программно-аппаратная платформа;
- 16) коммуникационное оборудование;
- 17) состав программного обеспечения;
- 18) состав системного персонала;
- 19) методология и инструментальные средства разработки систем.

1.3. Технологии корпоративных ИС

Современный уровень развития аппаратных и программных средств с некоторых пор сделал возможным повсеместное ведение баз данных оперативной информации на разных уровнях управления. В процессе своей деятельности промышленные предприятия, корпорации, ведомственные структуры, органы государственной власти и управления накопили большие объемы данных. Они хранят в себе большие потенциальные возможности по извлечению полезной аналитической информации, на основе которой можно выявлять скрытые тенденции, строить стратегию развития, находить новые решения.

В последние годы в мире оформился ряд новых концепций хранения и анализа корпоративных данных:

- 1) хранилища данных (Data Warehouse);
- 2) оперативная аналитическая обработка (On-Line Analytical Processing, OLAP);
- 3) интеллектуальный анализ данных – ИАД (Data Mining).

Технологии OLAP тесно связаны с технологиями построения хранилища данных (Data Warehouse) и методами интеллектуальной обработки – Data Mining. Поэтому наилучшим вариантом является комплексный подход к их внедрению.

1.3.1. OLTP-системы (On-Line Transaction Processing)

Информационные системы класса OLTP (On-Line Transaction Processing) или OLTP-системы предназначены, прежде всего, для

обслуживания повседневной деятельности предприятия. Основная функция подобных систем заключается в выполнении большого количества коротких транзакций. *Транзакцией* называют неделимую с точки зрения воздействия на базу данных последовательность операций манипулирования данными.

Сами транзакции являются достаточно простыми, но проблемы состоят в том, что таких транзакций очень много, выполняются они одновременно и при возникновении ошибок транзакция должна откатиться и вернуть систему в состояние, в котором та была до начала транзакции. Практически все запросы к базе данных в OLTP-приложениях состоят из команд вставки, обновления и удаления. Типичными примерами OLTP-приложений являются системы складского учета, заказов билетов, операционные банковские системы и др. Запросы на выборку в OLTP-системах, в основном, предназначены для предоставления пользователям выборки данных из различного рода справочников. Поскольку большая часть запросов известна заранее ещё на этапе проектирования системы, то критическим для OLTP-приложений является скорость и надежность выполнения коротких операций обновления данных.

Таким образом, OLTP-системы имеют следующие особенности:

- рассчитаны на быстрое обслуживание относительно простых запросов большого числа пользователей;
- работают с данными, которые требуют защиты от несанкционированного доступа, нарушений целостности, аппаратных и программных сбоев.

Для обеспечения целостности данных и изолированности пользователей транзакции в OLTP-системах должны обладать четырьмя основными свойствами:

1. Атомарность. Транзакция должна выполняться как единая операция доступа к базе данных (БД) и может быть выполнена полностью либо не выполнена совсем.

2. Согласованность. Свойство согласованности гарантирует взаимную целостность данных, т. е. выполнение ограничений целостности БД после окончания обработки транзакции.

3. Изолированность. Это свойство означает, что транзакции должны выполняться независимо друг от друга, и доступ к данным, изменяемым с помощью одной транзакции, для других транзакций должен быть запрещен, пока изменения не будут завершены.

4. Долговечность. Свойство долговечности означает, что если транзакция выполнена успешно, то произведенные ею изменения в данных не должны быть потеряны ни при каких обстоятельствах.

Стратегия разработки OLTP-систем

Длительное время в качестве стратегии разработки OLTP-систем использовались следующие принципы:

– построение отдельных автоматизированных рабочих мест (АРМ), предназначенных для обработки групп функционально связанных документов, и тиражирование готовых АРМ на места;

– построение полнофункциональных параметризуемых систем с тиражированием и настройкой по местам. Однако получаемые таким способом системы имели невысокие адаптационные возможности, предъявляли высокие требования к эксплуатационному персоналу и требовали больших накладных расходов на сопровождение.

Относительно недавно начала применяться новая, третья стратегия разработки информационных систем класса OLTP. Ее суть состоит в том, что тиражируются не готовые системы, а некоторые заготовки и технологический инструмент, позволяющие непосредственно на месте быстро построить/достроить систему с необходимой функциональностью и далее с помощью этого же инструмента ее модифицировать в соответствии с динамикой предметной области.

1.3.2. Хранилища данных (Data Warehouse)

Хранилище данных (ХД) – предметно-ориентированный, интегрированный, неизменяемый, поддерживающий хронологию набор данных, организованный для целей поддержки управления.

По аналогии с реальными хранилищами, в хранилищах данных имеются большие области для сбора/хранения/перемещения существующих данных. Понятие «хранение данных» возникло, в середине 1980-х гг., и, по сути, предназначалось для описания архитектурной модели потока данных от операционной системы к средствам поддержки принятия решений. Без такой архитектурной модели передаваемая управляющая информация обычно содержит большое количество избыточных данных.

В больших корпорациях множественные проекты принятия решений обычно осуществляются независимо и при этом используется один и тот же набор данных. Таким образом, происходит накопление дублированных данных, что в конечном итоге приводит к снижению эффективности поддержки принятия решений.

1.3.2.1. Очистка данных в хранилище данных

Для повышения эффективности поддержки принятия решений и уменьшения дублированности данных применяют очистку данных (data cleaning, data cleansing или scrubbing). В ХД очистку данных также применяют для выявления и удаления ошибок, несоответствий в данных с целью улучшения их качества.

Хранилища данных требуют и одновременно обеспечивают всестороннюю поддержку очистки данных. Они загружают и постоянно обновляют огромные объемы данных из различных источников, поэтому

вероятность попадания в них «грязных данных» весьма высока. Более того, хранилища данных используются в процессе принятия решений, следовательно, чтобы некорректные данные не привели к некорректным выводам, необходимо проводить корректировки таких данных. Например, дублирующаяся или утраченная информация может стать причиной некорректной или неадекватной статистики («мусор на входе – мусор на выходе»). Ввиду большого спектра возможных несоответствий в данных и большого объема данных их очистка считается одной из самых крупных проблем в технологии хранилищ данных.

Приложения очистки данных обычно выполняют одну или несколько следующих функций:

1. **Парсинг.** Имя и адрес клиента часто хранится в текстовых полях свободного формата. Текст свободного формата иногда труден для разбиения на самостоятельные подстроки, соответствующие типу поля, к которому они относятся (индекс, область, город, улица и другие данные подобного характера). Программное обеспечение, осуществляющее парсинг, распознает такие подстроки и назначает им соответствующие поля. Например, приложение Firstlogic I.D. Sentic содержит функции парсинга, включающие в себя возможность идентификации компонент имени, должности, компании и адреса даже в случае непоследовательно введенных данных.

2. **Стандартизация.** Данные имен и адресов могут вводиться в различных форматах, многие из которых вполне грамматически корректны. Например, «Улица», «ул.» и «ул" обозначают одно и то же очевидное понятие в составе адреса. Программы стандартизации трансформируют такие поля в согласованный набор обозначений, подходящих для Почтовой службы.

3. **Проверка допустимости.** К этой функции относятся средства распознавания допустимости вводимых данных. Например, приложение компании Firstlogic, объединенное с программой проверки допустимости и файлами почтовых адресов (postal address files, PAFs) позволяет проверить допустимость международных адресных данных.

4. **Улучшение.** В данной функции используются ряд дополнительных данных, фактов или записей, изначально не содержащиеся в исходных данных. Так, например, программное обеспечение Firstlogic содержит возможность присвоения клиентам пола на основании анализа его имени и других данных. Кроме того, многие производители программ используют географическую информацию, обозначающую гео-код: долготу и широту указанной местности. Наиболее же ценным дополнением данных о клиентах являются данные третьих фирм, содержащие демографическую, психографическую и другие виды информации. Например, программа Trillium Software специализируется на внедрении психографической и демографической информации в профайл клиента.

5. **Согласование и консолидация.** Как только имя и адрес очищены, для устранения дублированных данных о клиентах в рамках каждого списка и соединения данных из различных источников применяется функция согласования. Большинство таких средств содержат алгоритмы расстановки приоритетов между полями (в процессе согласования) и контроля очередности сравнения полей.

1.3.2.2. Структура хранилища данных

В состав хранилища данных, как правило, входит:

- виртуальное хранилище данных;
- витрины данных;
- глобальное хранилище данных;
- многоуровневая архитектура хранилища данных.

Виртуальное хранилище данных. В его основе лежит репозиторий метаданных, который описывается источниками информации (БД транзакционных систем, внешние файлы и др.), SQL-запросами для их считывания и процедурами обработки и предоставления информации. Непосредственный доступ к последним обеспечивает программное обеспечение промежуточного слоя. В этом случае избыточность данных нулевая. Конечные пользователи фактически работают с транзакционными системами напрямую со всеми вытекающими отсюда плюсами (доступ к не агрегированным данным в реальном времени) и минусами (интенсивный сетевой трафик, снижение производительности OLTP-систем и реальная угроза их работоспособности вследствие неудачных действий пользователей-аналитиков).

Витрина данных. Витрина данных (Data Mart) – это облегченный вариант хранилища данных, содержащий только тематически объединенные данные. Целевая база данных максимально приближена к конечному пользователю и может содержать тематически ориентированные агрегатные данные. Витрина данных существенно меньше по объему, чем хранилище данных, поэтому его реализации не требуется мощная вычислительная техника.

Глобальное хранилище данных. В последнее время 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. Тогда естественной становится следующая трехуровневая архитектура системы.

На **первом уровне** реализуется корпоративное хранилище данных на основе одной из развитых современных реляционных СУБД. Это хранилище состоит, в основном, из детализированных данных. Реляционные СУБД обеспечивают эффективное хранение и управление данными очень большого объема, но не слишком хорошо соответствуют

потребностям OLAP-систем, в частности, в связи с требованием многомерного представления данных.

На **втором уровне** поддерживаются витрины данных на основе многомерной системы управления базами данных (примером такой системы является Oracle Express Server). Такие СУБД почти идеально подходят для целей разработки OLAP-систем, но пока не позволяют хранить сверхбольшие объемы данных (предельный размер многомерной базы данных составляет 10–40 Гбайт). В данном случае это и не требуется, поскольку речь идет о витринах данных. Необходимо заметить, что витрина данных не обязательно должна быть полностью сформирована. Она может содержать ссылки на хранилище данных и добирать оттуда информацию по мере поступления запросов. Конечно, это несколько увеличивает время отклика, но зато снимает проблему ограниченного объема многомерной базы данных.

На **третьем уровне** находятся клиентские рабочие места конечных пользователей, на которых устанавливаются средства оперативного анализа данных.

На рис. 1.2 изображены потоки данных в информационной системе (ИС) для большой организации, имеющей самостоятельные, часто удаленные подразделения.

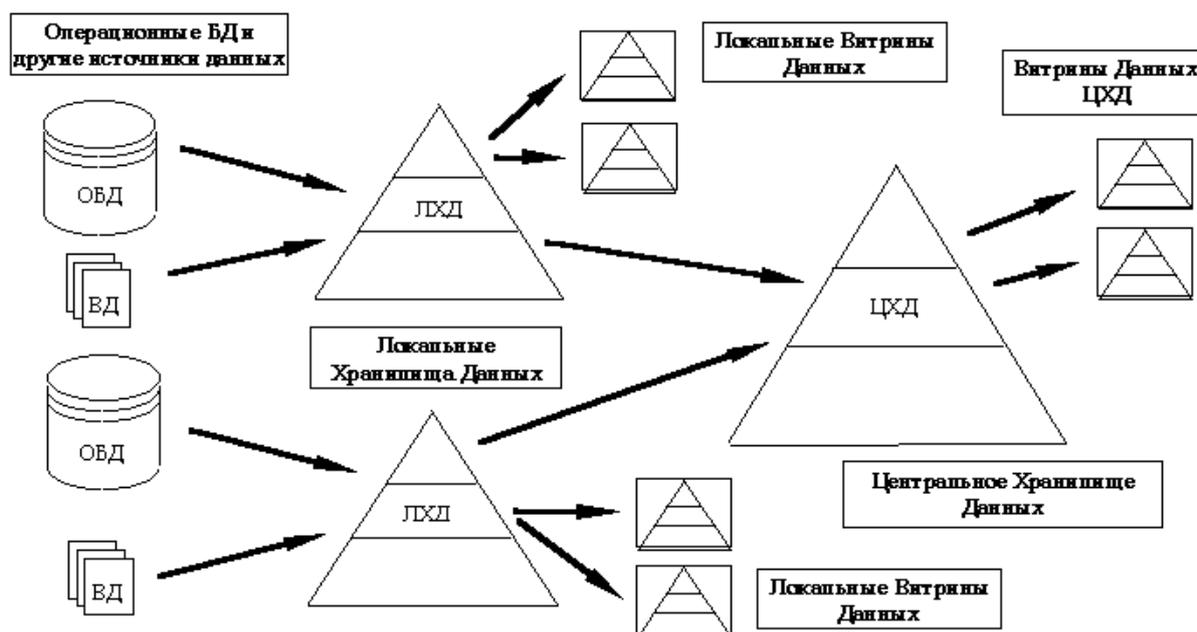


Рис. 1.2. Потоки данных в ИС

В такой ИС данные, попадающие в хранилище данных, не используются напрямую системами представления и анализа. Эти системы получают данные из специально сформированных витрин данных. Введение витрин данных позволяет получить несколько важных преимуществ:

- конечный пользователь работает только с теми данными, которые

ему нужны;

- повышается безопасность доступа к данным;
- структура данных отражает требования конечного пользователя;
- упрощается проектирование данных;
- снижается нагрузка на основное хранилище данных.

Для больших организаций технология хранилища данных реализуется в иерархической схеме. Для хранилища данных верхнего уровня хранилище данных уровнем ниже являются таким же источником данных, как и операционные базы данных (ОБД). ОБД является основным, но не единственным источником информации. Это связано с тем, что часть информации (иногда даже существенная) хранятся в форматах, отличных от принятых в БД. Среди таких форматов самым распространенным является текстовый файл, а средством доступа – файловая операционная система. Эти источники данных называются внешними данными.

Хранилища данных обладают рядом свойств:

1. *Предметная ориентация*

В отличие от традиционной схемы реализации информационной системы, где источником данных для средств анализа являются ОБД, в которых данные ориентированы на обработку и функциональность систем сбора информации, данные в ХД ориентированы на решение задач анализа и представления данных. Предметная ориентация является фундаментальным отличием ОБД от ХД. Именно это свойство позволяет конечному пользователю работать с данными, охватывающими деятельность организации в целом. Разные приложения ОБД могут описывать одну и ту же предметную область с разных точек зрения и решение, принятое на основе данных, отражающих только одну сторону вопроса, могут быть неэффективными, а порой и просто неверными.

Следует отметить, что предметная ориентация позволяет также существенно ускорить доступ к данным за счет предварительной переструктуризации данных в момент загрузки.

Предметная ориентация позволяет также хранить в ХД только те данные, которые необходимы для средств анализа, что существенно сокращает затраты на носители информации и повышает безопасность доступа к данным.

Поскольку в технологии ХД объекты данных выходят на первый план, то особые требования предъявляются к структурам БД, используемым для создания информационных хранилищ. Принципиально отличаются и структуры баз данных для OLTP-систем и систем ХД. Во втором случае в них помещается только та информация, которая может быть полезной для работы систем поддержки принятия решений (СППР).

2. *Интегрированность данных*

Данные в информационное хранилище поступают из различных источников, где они могут иметь разные имена, атрибуты, единицы

измерения и способы кодировки. После загрузки в ХД данные очищаются от индивидуальных признаков. С этого момента они представляются пользователю в виде единого информационного пространства.

В качестве примера можно привести обработку данных о поле человека. Если в четырех разных приложениях пол клиента кодировался различными способами (женский, мужской, жен., муж., ж, м, Ж, М), то в информационном хранилище будет использована единая для всех данных схема кодировки (например, жен., муж.).

3. *Инвариантность во времени*

В OLTP-системах истинность данных гарантирована только в момент чтения, поскольку уже в следующее мгновение они могут измениться в результате очередной транзакции. Важным отличием ХД от OLTP-систем является сохранение истинности данных в любой момент процесса чтения.

В OLTP-системах информация часто модифицируется как результат выполнения каких-либо транзакций. Временная инвариантность данных в ХД достигается за счет введения полей с атрибутом «время» (день, неделя, месяц) в ключи таблиц. В результате записи в таблицах ХД никогда не изменяются, представляя собой «моментальные снимки» данных, сделанные в определенные отрезки времени. Каждый элемент в своем ключе явно или косвенно хранит временной параметр, например, день, месяц или год.

4. *Неразрушаемость – стабильность информации*

В OLTP-системах записи могут регулярно добавляться, удаляться и редактироваться. В системах ХД, как следует из требования временной инвариантности, однажды загруженные данные теоретически никогда не меняются. По отношению к ним возможны только две операции: начальная загрузка и чтение (доступ).

5. *Интеграция*

Различные ОБД разрабатываются различными коллективами разработчиков, зачастую в разное время и различными средствами разработки. Это приводит к тому, что объекты, отражающие одну сущность, имеют различные наименования и единицы измерения. Обязательная интеграция данных в ХД позволяет решить эту проблему.

6. *Минимизация избыточности информации*

В ХД информация загружается из ОБД или OLTP-систем, при этом избыточность оказывается минимальной (около 1%), что объясняется следующими причинами:

– при загрузке информации из OLTP-систем в ХД данные фильтруются. Многие из них вообще не попадают в ХД, поскольку лишены смысла с точки зрения использования в системах поддержки принятия решений;

– в ХД хранится некая итоговая информация, которая в базах данных OLTP-систем вообще отсутствует;

– во время загрузки в ХД записи сортируются, очищаются от ненужной информации и приводят к единому формату. После такой обработки это уже совсем другие данные.

1.3.2.3. Классификация данных в хранилище данных

Все данные в хранилище данных делятся на три основных категории:

- метаданные;
- детальные (текущие) данные;
- агрегированные данные.

В отличие от ОБД ХД имеет хорошо развитую структуру метаданных (или данных о данных). Метаданные играют роль справочника, содержащего сведения об источниках первичных данных, алгоритмах обработки, которым исходные данные были подвергнуты и т. д. Наличие метаданных позволяет осуществлять быструю и удобную навигацию по различным уровням данных, а также сильно упрощает реализацию и использование средств анализа.

Детальные (текущие) данные в ХД являются денормализованными, по сравнению с нормализованными данными в большинстве ОБД. Это свойство ХД позволяет существенно повысить скорость доступа к необходимым данным, хотя и требует большей емкости носителей информации.

Наличие хорошо развитой иерархии агрегированных данных по уровням агрегации является отличительной чертой хранилища данных. Проведенные исследования показали, что большинство конечных пользователей не работают с детальными данными, а обращаются в основном к агрегированным показателям. Структура ХД отражает эту ситуацию и позволяет конечному пользователю быстро и удобно получать интересующую его агрегированную информацию с последующей навигацией по всем уровням агрегирования.

В процессе эксплуатации необходимость в ряде детальных данных может сильно упасть, что является причиной подразделения детальных данных на текущие и старые (рис. 1.3). В то время как текущие данные регулярно используются и поэтому хранятся на накопителях с быстрым доступом (в основном на жестких дисках), старые детальные данные могут храниться на более емких накопителях с более медленным доступом (например, на оптических дисках или магнитных лентах).

При создании хранилища данных крайне нежелательно использование оптимизированных структур (многомерных, звездообразных и др.), ввиду свойственной им негибкости. Использование реляционной, нормализованной модели в качестве основы хранилища данных максимально облегчит дальнейшее развитие такого хранилища. Если при этом, запросы пользователя поступают только в витрины данных, то от хранилища данных потребуется вместо необходимости поддерживать

специализированные запросы, лишь периодически создавать выборки для витрин данных.

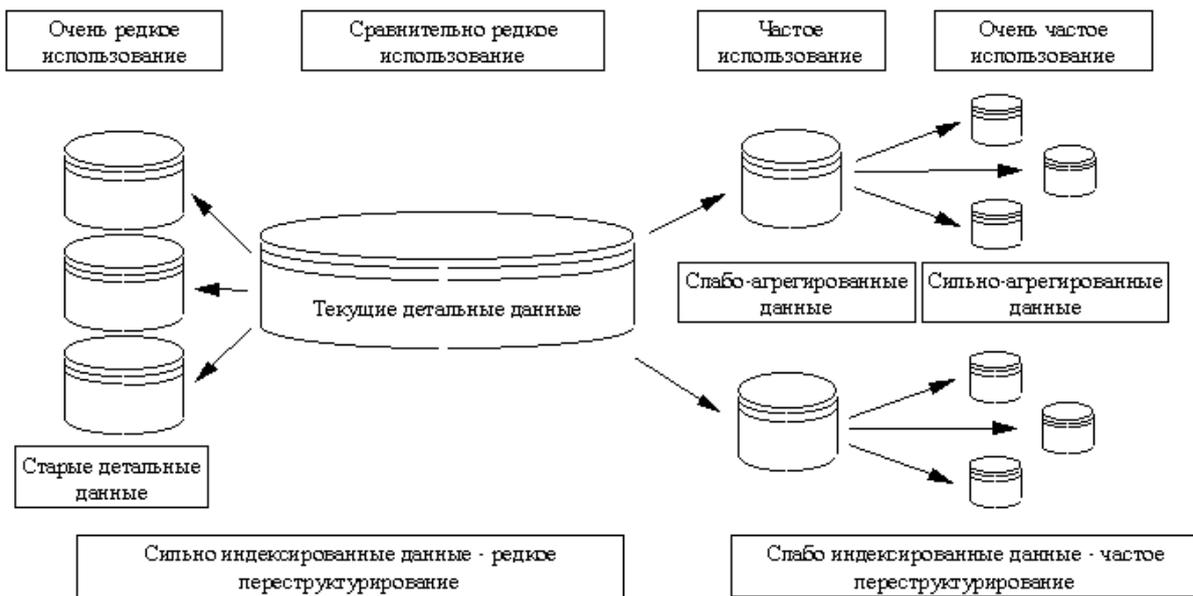


Рис. 1.3. Использование данных

После того, как данные попали в хранилище, они могут быть распространены по любому количеству витрин данных для обеспечения к ним доступа пользователей. Эти витрины данных могут иметь произвольную форму (от базы данных клиент-сервер до настольных баз данных, OLAP кубов и даже электронных таблиц). Выбор средств создания пользовательских запросов может быть довольно широк, и может соответствовать предпочтениям и опыту пользователя.

1.3.2.4. Моделирование хранилищ данных

Традиционные подходы моделирования хранилищ данных основываются, как правило, на использовании временных отметок создания записей и их модификации. На данный момент известны три основных способа моделирования времени в хранилищах данных:

1. Модель снимков данных

Снимок данных – это представление данных в определенный момент времени. Данная модель характерна для оперативных систем (OLTP). Обновления данных носят деструктивный характер, т. е. предыдущие значения атрибутов замещаются новыми значениями (рис. 1.4).

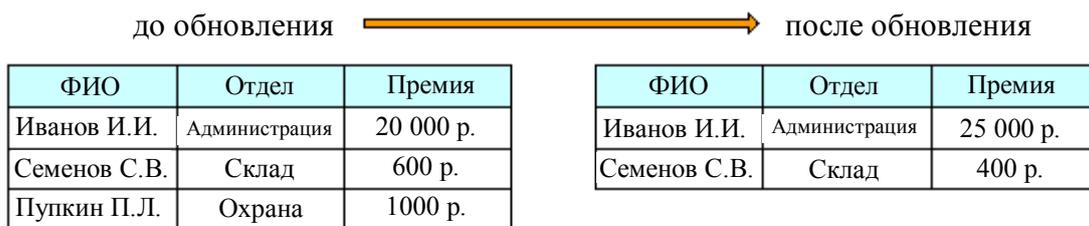


Рис. 1.4. Модель снимков данных

Модель имеет достаточно ограниченный круг применения в хранилищах данных, поскольку не обеспечивает хранения истории изменений.

2. Событийная модель

Событийная модель (рис. 1.5) используется для моделирования событий (данных), возникающих в определенные моменты времени. Данная модель подходит для моделирования транзакций, таких как: продажи, финансовые транзакции, складские операции и т. д.

Счет	Клиент	Сумма	Дата
2000	Школа-Инфо	55 000 р.	22.03.2006
2001	Beeline	4 500 р.	22.03.2006
2002	ТехноСтиль	12 000 р.	23.03.2006

Рис. 1.5. Событийная модель

3. Статусная модель

Статусная модель используется для моделирования состояния объектов во времени. Она подходит для представления данных, имеющих нетранзакционный характер.

Существует три способа моделирования изменяющихся во времени статусов:

а) непрерывная модель – для хранения промежутков времени используется одно поле даты. Дата начала следующего периода совпадает с датой окончания предыдущего;

б) начало и конец – для хранения промежутков времени используется два поля – дата начала и дата окончания периода действия статуса;

в) начало и длительность – для хранения промежутков времени используется одно поле даты (дата начала) и поле длительности периода. Больше распространение при создании статусных моделей получил способ «начало и конец» (рис. 1.6).

ФИО	Отдел	Зарплата	Дата начала	Дата окончания
Иванов И.И.	Администрация	30 000 р.	01.01.2006	
Семенов С.В.	Склад	4 500 р.	05.07.2006	05.12.2006
Пупкин П.Л.	Охрана	7 000 р.	01.09.2006	01.01.2006

Рис. 1.6. Статусная модель

Статусная и событийная модели являются взаимно дополняющими. Путем преобразований из одной можно получить другую. Например, зная остаток на счете на определенный момент и историю транзакций в событийной модели, можно восстановить все статусы счета (остатки на счете) в периоды между транзакциями. И наоборот, имея статусную модель остатков на счете, можно вычислить события (т. е. транзакции), которые происходили со счетом в начале (конце) каждого периода.

1.3.2.5. Моделирование витрин данных

Приемы моделирования витрин данных отличаются от приемов моделирования хранилищ данных в силу различных требований к структурам данных. Если основной задачей хранилища данных является хранение консолидированной исторической информации, то витрина данных строится с учетом требований по доступу к данным и представления информации.

Как правило, для моделирования витрин данных используются типы модели под названием: схема «звезда» и схема «снежинка».

Схема «звезда»

Схема «звезда» – популярный тип модели данных для витрин данных. Данная модель характеризуется наличием таблицы фактов, окруженной связанными с ней таблицами размерностей. Запросы к такой структуре включают простые объединения таблицы фактов с каждой из таблиц размерностей (рис. 1.7). Данная модель характеризуется высокой производительностью запросов.



Рис. 1.7. Схема «звезда»

Схема «звезда» имеет следующие компоненты:

1. **Размерность**

В технологии многомерного моделирования размерность – это аспект, в разрезе которого можно получать, фильтровать, группировать и отображать информацию о фактах.

Типичные размерности, встречающиеся практически в любой модели:

- покупатель,
- продукт,
- время (период),
- география,
- сотрудник.

Размерность, как правило, имеет многоуровневую иерархическую структуру. Например, размерность время может иметь следующую структуру: год, квартал, месяц, день и т. д.

2. Факты

Факты – это величины, обычно числовые, хранящиеся в таблице фактов и являющиеся предметом анализа. Примеры фактов: объем операций, количество проданных единиц товара и т. д.

Факты бывают трех типов:

а) аддитивные факты. Аддитивность определяет возможность суммирования факта вдоль определенной размерности. Аддитивные факты можно суммировать и группировать вдоль всех размерностей на любых уровнях иерархии;

б) полуаддитивные факты. Полуаддитивный факт – это факт, который можно суммировать вдоль определённых размерностей, и нельзя – вдоль других. Примером может служить остаток на счете (или остаток товара на складе). Данную величину нельзя суммировать вдоль размерности время. Однако сумма остатков по счетам является часто встречаемой величиной для анализа;

в) неаддитивные факты. Неаддитивные факты вообще нельзя суммировать. Пример неаддитивного факта – отношение (например, выраженное в процентах).

3. Таблицы покрытия

Таблицы покрытия используются с целью моделирования сочетания размерностей, для которых отсутствуют факты. Например, нужно найти количество категорий продуктов, которые сегодня ни разу не продавались. Таблица фактов продаж не может ответить на данный вопрос, поскольку она регистрирует только факты продаж. Для того чтобы модель позволяла отвечать на подобные вопросы, необходима дополнительная таблица фактов (которая, по сути дела, не содержит фактов) и которая и называется таблицей покрытия.

Схема «снежинка»

Данная схема (рис. 1.8) используется для нормализации схемы «звезда».



Рис. 1.8. Схема «снежинка»

Такая схема сокращает избыточность в таблицах размерностей. Основным достоинством схемы «снежинка» является возможность создания таблицы фактов с разным уровнем детализации. Например, фактические данные на уровне дня, а плановые – на уровне месяца. Вследствие чего возрастает скорость выполнения запросов в структуре размерностей. Например, запросы вида «выбрать все строки из таблицы размерности на определенном уровне».

1.3.3. Способы аналитической обработки данных

Для того чтобы существующие хранилища данных способствовали принятию управленческих решений, информация должна быть представлена аналитику в нужной форме, т. е. он должен иметь развитые инструменты доступа к данным хранилища и их обработки.

Очень часто информационно-аналитические системы, создаваемые в расчете на непосредственное использование лицами, принимающими решения, оказываются чрезвычайно просты в применении, но жестко ограничены в функциональности. Такие статические системы называются Информационными системами руководителя (ИСР), или Executive Information Systems (EIS). Они содержат в себе множества запросов и, будучи достаточными для повседневного обзора, неспособны ответить на все вопросы которые могут возникнуть при принятии решений. Результатом работы такой системы, как правило, являются многостраничные отчеты, после тщательного изучения, которых у аналитика появляется новая серия вопросов. Однако каждый новый запрос, непредусмотренный при проектировании такой системы, должен быть сначала формально описан, закодирован программистом и только затем выполнен. Время ожидания в таком случае может составлять часы и дни, что не всегда приемлемо.

1.3.3.1. Системы поддержки принятия решений

Системы поддержки принятия решений (СППР) – это класс ИС, предназначенных для решения задач поддержки всех стадий принятия решений в слабо структурируемых и неструктурируемых предметных областях непосредственными пользователями в процессе аналитического моделирования. Специфика СППР заключается, прежде всего, в предоставлении пользователю возможности получения нерегламентированных отчетов, различных методов анализа данных, что позволяет эффективнее решать слабо структурируемые и неструктурируемые задачи, вырабатывать специфические, нетиповые решения. Для решения подробного рода задач пользователю, как правило, требуется дополнительная уникальная и, зачастую, разовая информация из корпоративного информационного хранилища. Поэтому, в отличие от традиционных отчетных ИС, осуществляющих предоставление

пользователю регламентной информации, СППР, как правило, обладают мощными механизмами интерактивного поиска, обобщения и анализа информации на основе нерегламентированных запросов.

1.3.3.1.1. Алгоритмы СППР

Полная структура информационно-аналитической системы, построенной на основе хранилища данных, показана на рис. 1.9.

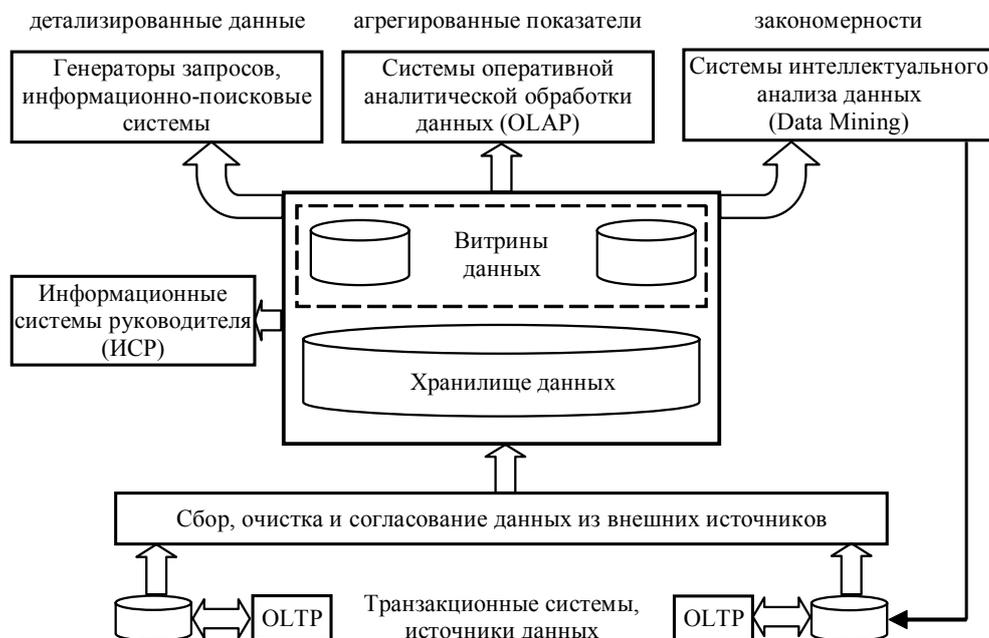


Рис. 1.9. Структура корпоративной информационно-аналитической системы

Важнейшей целью СППР является, в первую очередь, обеспечение пользователя технологией формирования информации. Так как процесс формирования нетиповых решений является трудно формализуемым, то вместо модулей, отражающих регламентную информацию, в составе СППР используется набор средств, каждое из которых может быть использовано пользователем для получения и анализа информации. СППР ориентированы не на процесс, а на набор возможностей, предоставляемых пользователю.

В статических СППР конечному пользователю предоставляется не поддержка однозначно описанного процесса обработки данных, а набор возможностей, не зависящих от процесса. Поэтому использование статических СППР невозможно без глубоких знаний пользователем предметной области.

Динамические СППР, напротив, ориентированы на обработку нерегламентированных запросов аналитиков к данным. Работа аналитиков с этими системами заключается в интерактивной последовательности формирования запросов и изучения их результатов.

Поддержка принятия управленческих решений на основе

накопленных данных может выполняться с помощью трех алгоритмов (рис. 1.9):

1. Генерация запросов, с использованием детализированных данных. Используется в большинстве систем, нацеленных на поиск информации. Как правило, реляционные СУБД отлично справляются с возникающими здесь задачами. Общеизвестным 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. Оперативная аналитическая обработка данных, с использованием агрегированных показателей. Позволяют реализовать комплексный взгляд на собранную в хранилище данных информацию, обобщение и агрегацию, гиперкубическое представление и многомерный анализ данных. Здесь могут использоваться либо специальные многомерные СУБД, либо реляционные технологии. 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. Интеллектуальный анализ данных, с использованием закономерностей. Главными задачами интеллектуального анализа данных являются поиск функциональных и логических закономерностей в накопленной информации, построение моделей и правил, объясняющие найденные аномалии и/или прогнозирующие развитие некоторых процессов.

1.3.3.1.2. Варианты архитектуры СППР

На сегодняшний день известно несколько способов построения СППР. Среди них можно выделить четыре наиболее популярных типа архитектур систем поддержки принятия решений:

- I – функциональная СППР;
- II – независимые витрины данных;
- III – двухуровневое хранилище данных;
- IV – трехуровневое хранилище данных.

I. Функциональная СППР

Функциональная СППР (рис. 1.10) является наиболее простой с архитектурной точки зрения. Каждый пользователь (аналитик) использует данные из любого доступного источника данных. Такие системы часто встречаются на практике, в особенности, в организациях с недостаточно развитой информационной инфраструктурой.

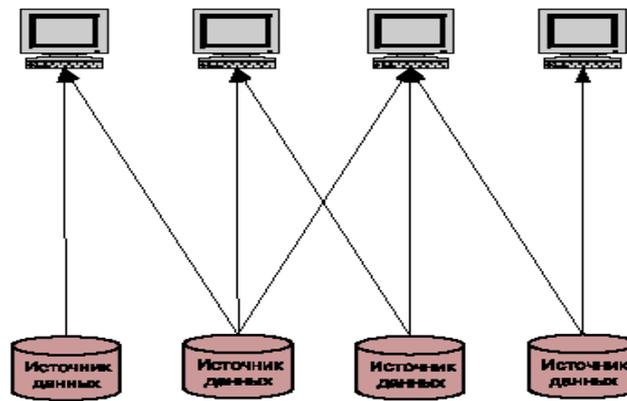


Рис. 1.10. Функциональная СДПР

К преимуществам функциональной СДПР можно отнести следующее:

- быстрое внедрение за счет отсутствия этапа перегрузки данных в специализированную систему;

- минимальные затраты за счет использования одной платформы.

Недостатками функциональной СДПР являются:

- оперативные системы характеризуются очень низким качеством данных с точки зрения их роли в поддержке принятия стратегических решений. В силу отсутствия этапа очистки данных данные функциональной СДПР, как правило, обладают невысоким качеством;

- большая нагрузка на оперативную систему. Сложные запросы могут привести к остановке работы оперативной системы.

II. СДПР с использованием независимых витрин данных

В СДПР с использованием независимых витрин данных (рис. 1.11) в зависимости от предметной области анализа формируются витрины данных, куда поступает информация из внешних источников данных. Пользователи (аналитики) для решения конкретных вопросов работают только с необходимыми витринами данных.

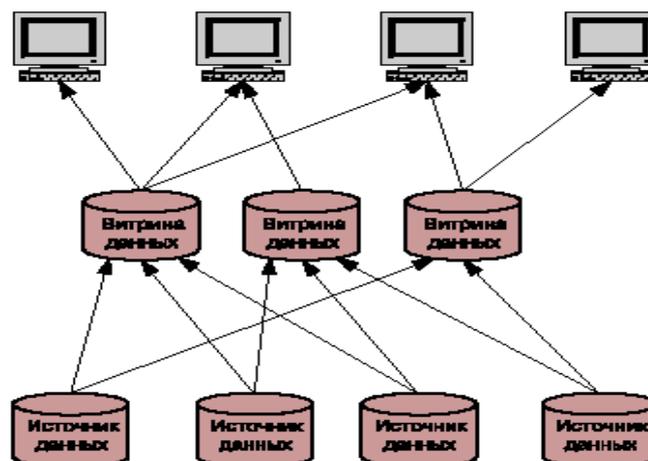


Рис. 1.11. СДПР с независимыми витринами данных

Как правило, отдельная витрина данных формируется для отдельного подразделения. Такие СДПР обычно встречаются в крупных организациях

с большим количеством независимых подразделений, зачастую имеющих свои собственные отделы информационных технологий.

Преимущества СППР с независимыми витринами данных:

- высокая скорость внедрения киосков (витрин) данных;
- проектирование витрины для конкретного ряда вопросов;
- повышение производительности СППР в связи с оптимизацией данных для определенных групп пользователей.

Недостатками СППР с независимыми витринами данных являются:

- многократное хранение данные в различных витринах данных. Это приводит к дублированию данных и, как следствие, к увеличению расходов на хранение и потенциальным проблемам, связанным с необходимостью поддержания непротиворечивости данных;
- потенциально очень сложный процесс наполнения витрин данных при большом количестве источников данных;
- данные не консолидируются на уровне предприятия, что способствует отсутствию единой картины бизнеса.

III. СППР на основе двухуровневого хранилища данных

В СППР на основе двухуровневого ХД (рис. 1.12) от всех доступных внешних источников информация поступает в ХД, где она проходит обработку, необходимую для эффективного функционирования ХД (парсинг, нормализация и т. д.). Для поддержки архитектуры двухуровневого ХД необходимо согласование всех определений и процессов преобразования данных в рамках всей организации.

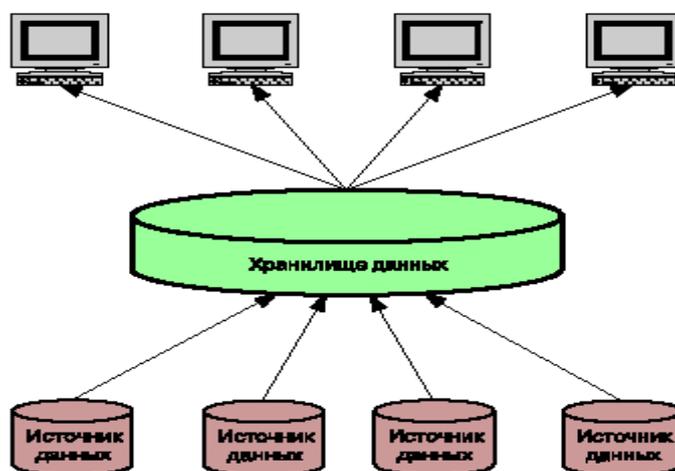


Рис. 1.12. Двухуровневое ХД

Преимуществами СППР с двухуровневым ХД являются:

- хранение данных в единственном экземпляре;
- минимальные затраты на хранение данных;
- отсутствие проблем, связанных с синхронизацией нескольких копий данных;
- консолидация данных на уровне предприятия, что позволяет иметь единую картину бизнеса.

К недостаткам можно отнести:

- данные не структурируются для поддержки потребностей отдельных пользователей или групп пользователей;
- возможны проблемы с производительностью системы;
- возможны трудности с разграничением прав пользователей на доступ к данным.

IV. СППР на основе трехуровневого хранилища данных

Трехуровневое ХД (рис. 1.13) представляет собой единый централизованный источник информации. Витрины данных представляют подмножества данных из ХД, организованные для решения задач отдельных подразделений компании. Конечные пользователи (аналитики) имеют возможность доступа к детальным данным хранилища, в случае если данных в витрине недостаточно, а также для получения более полной картины состояния бизнеса.

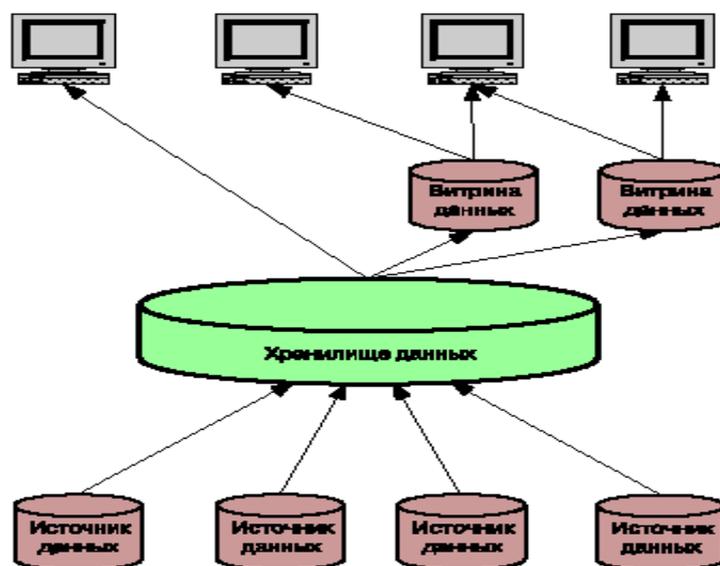


Рис. 1.13. Трехуровневое хранилище данных

Преимуществами СППР с трехуровневым ХД являются:

- упрощение процедуры создания и наполнения витрин данных. Это связано с тем, что наполнение происходит из единого стандартизованного надежного источника очищенных нормализованных данных;
- синхронизация и совместимость витрины данных с корпоративным представлением. Существует возможность сравнительно легкого расширения хранилища и добавления новых витрин данных;
- гарантированная производительность.

Недостатками таких СППР являются:

- существование избыточности данных, приводящих к росту требований на хранение данных;
- требование согласованности принятых архитектурных решений с различными корпоративными требованиями (например, скорость внедрения иногда конкурирует с архитектурными решениями).

1.3.3.2. Оперативная аналитическая обработка (On-Line Analytical Processing, OLAP)

Технология комплексного многомерного анализа данных получила название OLAP (On-Line Analytical Processing). OLAP – это ключевой компонент организации хранилищ данных. Концепция OLAP была описана в 1993 г. Эдгаром Коддом и имеет следующие требования к приложениям для многомерного анализа:

- многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (ключевое требование OLAP);
- предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;
- возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;
- многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;
- возможность обращаться к любой нужной информации независимо от ее объема и места хранения.

OLAP-система состоит из множества компонент. На самом высоком уровне представления система включает в себя источник данных, многомерную базу данных (МБД), предоставляющая возможность реализации механизма составления отчетов на основе технологии OLAP, OLAP-сервер и клиента. Система построена по принципу клиент-сервер и обеспечивает удаленный и многопользовательский доступ к серверу МБД (рис. 1.14).

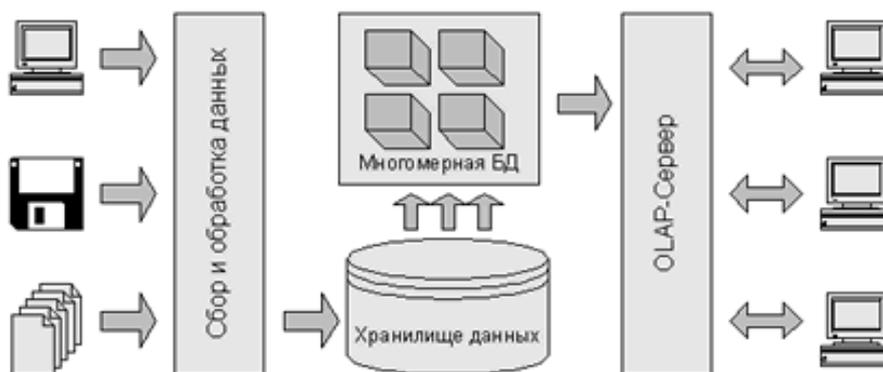


Рис. 1.14. Структурная схема системы аналитической обработки

Рассмотрим составные части OLAP-системы.

Источники. Источником в OLAP-системах является сервер, поставляющий данные для анализа. В зависимости от области использования OLAP-продукта источником может служить хранилище данных, наследуемая база данных, содержащая общие данные, набор

таблиц, объединяющих финансовые данные или любая комбинация перечисленного.

Хранилище данных. Исходные данные собираются и помещаются в хранилище, спроектированное в соответствии с принципами построения хранилищ данных. ХД представляет из себя реляционную базу данных (РБД). Основная таблица ХД (таблица фактов) содержит числовые значения показателей, по которым собирается статистическая информация.

Многомерная база данных. Хранилище данных служит поставщиком информации для многомерной базы данных, которая является набором объектов. Основными классами этих объектов являются измерения и показатели. К измерениям относятся множества значений (параметров), по которым происходит индексация данных, например, время, регионы, тип учреждения и пр. Каждое измерение заполняется значениями из соответствующих таблиц измерений хранилища данных. Совокупность измерений определяет пространство исследуемого процесса. Под показателями понимаются многомерные кубы данных (гиперкубы). В гиперкубе содержатся сами данные, а также агрегатные суммы по измерениям, входящим в состав показателя. Показатели составляют основное содержание МБД и заполняются в соответствии с таблицей фактов. Вдоль каждой оси гиперкуба данные могут быть организованы в виде иерархии, представляющей различные уровни их детализации. Это позволяет создавать иерархические измерения, по которым при последующем анализе данных будут осуществляться агрегирование или детализация представления данных. Типичным примером иерархического измерения служит список территориальных объектов сгруппированных по районам, областям, округам.

Сервер. Прикладной частью OLAP-системы является OLAP-сервер. Эта составляющая выполняет всю работу (в зависимости от модели системы), и хранит в себе всю информацию, к которой обеспечивается активный доступ. Архитектурой сервера управляют различные концепции. В частности, основной функциональной характеристикой OLAP-продуктов является использование МБД либо РБД для хранения данных.

Клиентское приложение. Данные, структурированные соответствующим образом и хранимые в МБД доступны для анализа с помощью клиентского приложения. Пользователь получает возможность удаленного доступа к данным, формулирования сложных запросов, генерации отчетов, получения произвольных подмножеств данных. Получение отчета сводится к выбору конкретных значений измерений и построению сечения гиперкуба. Сечение определяется выбранными значениями измерений. Данные по остальным измерениям суммируются.

1.3.3.2.1. Многомерная модель данных в OLAP

Основными понятиями многомерной модели данных являются:

- Гиперкуб данных (Data Hypercube)
- Измерение (Dimension)
- Метки (Members)
- Ячейка (Cell)
- Мера (Measure)

Гиперкуб данных содержит одно или более измерений и представляет собой упорядоченный набор ячеек (рис. 1.15). Каждая ячейка определяется одним и только одним набором значений измерений – меток. Ячейка может содержать данные – меру или быть пустой.

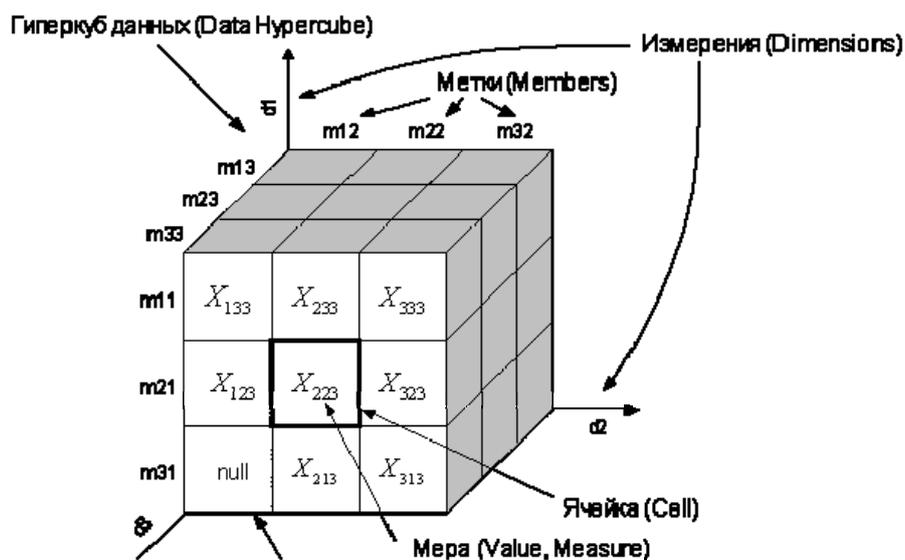


Рис. 1.15. Гиперкуб данных

Измерением называется множество меток, образующих одну из граней гиперкуба. Примером временного измерения является список дней, месяцев, кварталов. Примером географического измерения может быть перечень территориальных объектов: населенных пунктов, районов, регионов, стран и т. д.

Для получения доступа к данным пользователю необходимо указать одну или несколько ячеек путем выбора значений измерений, которым соответствуют необходимые ячейки. Процесс выбора значений измерений называется фиксацией меток, а множества выбранных значений измерений – множеством фиксированных меток.

Пусть $D = \{d_1, d_2, \dots, d_n\}$ – множество измерений гиперкуба, $M_{d_i} = \{m_1, m_2, \dots, m_k\}$, $(i = 1, \dots, n)$ – множество меток измерения d_i , $M = M_{d_1} \cup M_{d_2} \cup \dots \cup M_{d_n}$ – множество меток гиперкуба, $D' \subseteq D$ – множество фиксированных измерений, $M' \subseteq M$ – множество фиксированных меток.

Гиперкуб данных является множеством ячеек $H(D, M)$, соответствующий множествам D, M . Подмножество гиперкуба данных, соответствующий множествам фиксированных значений D', M'

обозначим как $\mathbf{H}'(D', M')$.

Каждой ячейке гиперкуба данных $h \in \mathbf{H}$ соответствует единственно возможный набор меток измерений $M_h \subset M$. Ячейка может быть пустой (не содержать данных) или содержать значение показателя – меру.

Множество мер гиперкуба $\mathbf{H}(D, M)$ обозначим как $V(\mathbf{H})$. Рассмотрим операции манипулирования данными в гиперкубе данных.

Операции манипулирования данными

Операция «Среза». Подмножество гиперкуба $\mathbf{H}'(D', M')$, получившееся в результате фиксации меток одного или более измерений, называется срезом (Slice). Операция построения среза проводится с целью получения требуемого подмножества ячеек $\mathbf{H}' \subset \mathbf{H}$ и отсека «ненужных» значений путем последовательной фиксации меток. Срез, как правило, представляет собой двумерный массив (таблицу).

Метка $m_{ji} \in M$ задает гиперплоскость сечения гиперкуба данных, соответствующую измерению $d_i \in D$. Множество фиксированных меток $M' \subseteq M$, таким образом, задает множество гиперплоскостей сечений гиперкуба данных, соответствующее множеству фиксированных измерений $D' \subseteq D$. Пересечение этих гиперплоскостей определяет множество ячеек (срез) гиперкуба данных $\mathbf{H}'(D', M')$, интересующих пользователя. Суть процесса выборки данных из гиперкуба данных, таким образом, состоит в построении среза гиперкуба данных $\mathbf{H}'(D', M')$ путем задания множеств D' и M' .

Операция «Вращения». Изменение порядка представления (визуализации) измерений называется вращением (Rotate). Вращение обеспечивает возможность визуализации данных в форме, наиболее комфортной для их восприятия.

В терминах рассматриваемой модели данных вращение означает смену последовательности фиксации меток при построении среза. Результатом вращения для двумерного среза (таблицы) будет замена столбцов на строки, а строк на столбцы.

Операции «свертки и детализации» осуществляются благодаря наличию иерархической структуры измерений. Значения измерений (метки) могут объединяться в иерархии, состоящие из одного или нескольких уровней (levels). Например, метки измерения «Время» естественно объединяются в иерархию с уровнями: год, квартал, месяц, день. Операции свертки и детализации принципиально не отличаются от операции построения среза гиперкуба данных, однако их выделяют для описания работы с агрегированными данными.

Наличие иерархической структуры измерений позволяет осуществлять агрегацию данных.

Агрегация данных. Агрегация данных – получение значений,

соответствующих меткам некоторого уровня l , $l \geq 1$ иерархического измерения D на основе значений уровня $l-1$.

Рассмотрим иерархическое измерение D с L уровнями (рис. 1.16). Первичные данные (факты) соответствуют нижнему уровню иерархии ($l=0$).

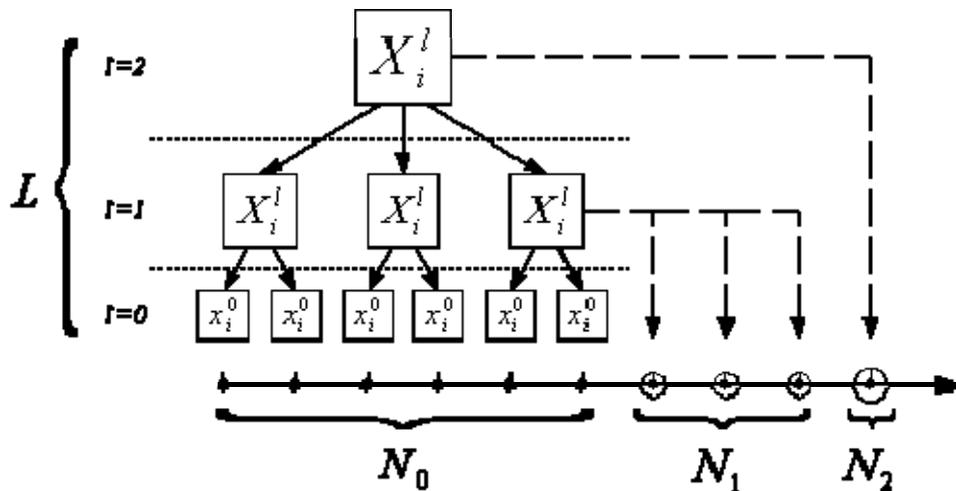


Рис. 1.16. Агрегация гиперкуба данных. Одномерное представление

Вычисление агрегатов производится в соответствии с применяемым методом агрегирования. Например, в случае суммирования значение агрегата X_j^1 на уровне иерархии $l=1$ может быть вычислено по формуле:

$X_j^1 = \sum_{i=1}^{M_j} x_i^0$, где M_j – количество фактов, соответствующих меткам, являющимся дочерними по отношению к метке j .

Обобщая, получим формулы вычисления агрегатов по методу суммирования на остальных уровнях иерархии: $X_j^l = \sum_{i=1}^{M_j} x_i^{l-1}$, $l=1, \dots, L$; $j=1, \dots, N_l$.

Ось измерения D , первоначально содержащая метки, соответствующие нижнему уровню иерархии ($l=0$), может быть дополнена метками, соответствующими уровням иерархии, начиная с $l=1$. Таким образом, различие между метками, соответствующими первичным данным и меткам, соответствующим агрегатам является условным.

Операция свертки данных в этом случае представляет собой построение среза гиперкуба данных, соответствующего смене метки уровня агрегации l_1 , где $0 \leq l_1 \leq L$ на уровень l_2 , где $l_1 \leq l_2 \leq L$. Операция детализации соответствует смене метки уровня l_1 ($1 \leq l_1 \leq L$) на уровень l_2 ($0 \leq l_2 \leq l_1$). Количество агрегатов для одного измерения определяется выражением: $N_A = \sum_{i=1}^L N_i$.

Рассмотрим случай двух измерений (рис. 1.17). Количество агрегатов,

хранимых в гиперкубе данных наряду с первичными данными, зависит от количества меток, соответствующих уровням иерархии измерений гиперкуба, начиная с $l=1$, и может существенно превышать количество первичных данных.

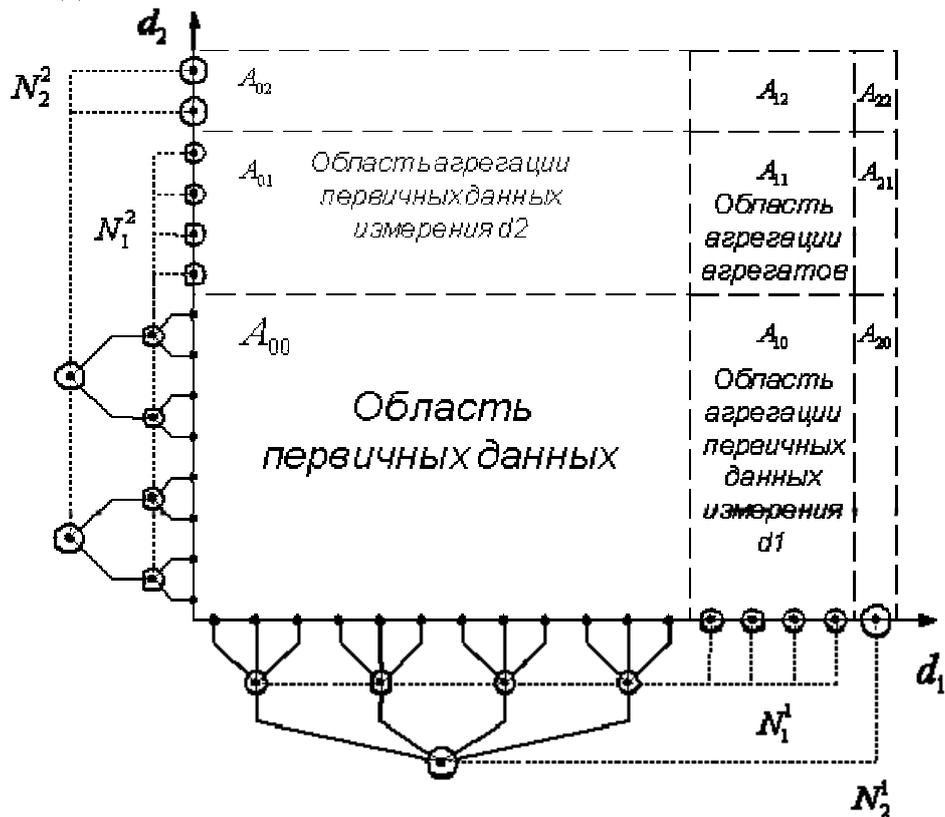


Рис. 1.17. Агрегация гиперкуба данных. Двумерное представление

В случае двух измерений число агрегатов будет составлять сумма значений областей: A_{01} , A_{02} , A_{10} , A_{11} , A_{12} , A_{20} , A_{21} , A_{22} . С другой стороны, количество агрегатов может быть вычислено как разность количества всех значений гиперкуба и количества значений, соответствующих области первичных данных A_{00} . Количество значений последней есть произведение $N_0^1 \times N_0^2$.

Таким образом, количество агрегатов N_A гиперкуба данных в двумерном случае составляет:

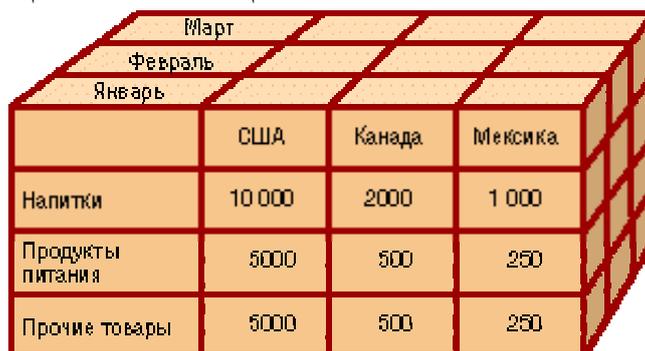
$$N_A = (N_0^1 + N_1^1 + \dots + N_{L_1}^1) \times (N_0^2 + N_1^2 + \dots + N_{L_2}^2) - N_0^1 \times N_0^2 = \sum_{i=0}^{L_1} N_i^1 \times \sum_{i=0}^{L_2} N_i^2 - N_0^1 \times N_0^2$$

Обобщая на случай произвольного числа измерений D , получим:

$$N_A = \prod_{j=1}^D \sum_{i=0}^{L_j} N_i^j - \prod_{j=1}^D N_0^j, \text{ где } N_i^j \text{ — количество меток } i\text{-го уровня иерархии измерения } j, j = 1, \dots, D, \text{ а } L_j \text{ — количество уровней иерархии измерения } j.$$

Рассмотрим многомерную модель данных на конкретном примере: трехмерных куб, определяющий продажи товаров в разных странах. В качестве мер в трехмерном кубе, изображенном на рис. 1.18, использованы суммы продаж, а в качестве измерений — время, товар и магазины.

Измерения представлены на определенных уровнях группировки: товары группируются по категориям, магазины – по странам, а данные о времени совершения операций – по месяцам.



	США	Канада	Мексика
Напитки	10 000	2000	1 000
Продукты питания	5000	500	250
Прочие товары	5000	500	250

Рис. 1.18. Трехмерный куб, определяющий продажи товаров в разных странах

«Разрезание» куба

Для визуализации данных, хранящихся в кубе, применяются, как правило, привычные двумерные, т. е. табличные, представления, имеющие сложные иерархические заголовки строк и столбцов.

Двумерное представление куба можно получить, «разрезав» его поперек одной или нескольких осей (измерений). Для этого фиксируются значения всех измерений, кроме двух и в результате получается обычная двумерная таблица. В горизонтальной оси таблицы (заголовки столбцов) представлено одно измерение, в вертикальной (заголовки строк) – другое, а в ячейках таблицы – значения мер. При этом набор мер фактически рассматривается как одно из измерений – необходимо либо выбирать для показа одну меру (и тогда можем разместить в заголовках строк и столбцов два измерения), либо показываем несколько мер (и тогда одну из осей таблицы займут названия мер, а другую – значения единственного «неразрезанного» измерения).

На рис. 1.19 изображен двумерный срез куба для одной меры – количество продаж и двух «неразрезанных» измерений – страна и месяц.

	США	Канада	Мексика
Январь	20 000	4000	2000
Февраль	30 000	6000	3000
Март	50 000	10 000	5000

Рис. 1.19. Двумерный срез куба для одной меры

Двумерное представление куба возможно и тогда, когда «неразрезанными» остаются и более двух измерений. При этом на осях среза (строках и столбцах) будут размещены два или более измерений «разрезаемого» куба (рис. 1.20).

	Январь			Февраль		
	США	Канада	Мексика	США	Канада	Мексика
Unit Sales	500	100	50	500	100	50
Store Sales	7500	1500	750	7500	1500	750
Store Cost	2500	500	250	2500	500	250

Рис. 1.20. Двумерный срез куба с несколькими измерениями на одной оси

В данном случае метки могут быть использованы неоднозначно. Они могут быть использованы как для «разрезания» куба, так и для ограничения (фильтрации) выбираемых данных – когда в измерении, остающемся «неразрезанным», нас интересуют не все значения, а их подмножество, например, три города из нескольких десятков лет. Значения меток отображаются в двумерном представлении куба как заголовки строк и столбцов.

Метки могут объединяться в иерархии, состоящие из одного или нескольких уровней (levels). Например, метки измерения «магазин» могут быть объединены в иерархию с уровнями:

- мир
- страна
- регион
- город
- магазин

В соответствии с уровнями иерархии вычисляются агрегатные значения. Например, объем продаж для России имеет уровень «страна», для Воронежской области – уровень «регион» и т. д. В одном измерении можно реализовать более одной иерархии, например, для времени: (Год, Квартал, Месяц, День) или (Год, Неделя, День).

1.3.3.2.2. OLAP на клиенте и на сервере

Многомерный анализ данных может быть проведен с помощью различных средств, которые условно можно разделить на клиентские и серверные OLAP-средства.

Клиентские OLAP-средства (например, Pivot Tables в Excel 2000 фирмы Microsoft или ProClarity фирмы Knosys) представляют собой приложения, осуществляющие вычисление агрегатных данных и их отображение. При этом сами агрегатные данные содержатся в кэше внутри адресного пространства такого OLAP-средства.

Если исходные данные содержатся в настольной СУБД, вычисление агрегатных данных производится самим OLAP-средством. Если же источник исходных данных – серверная СУБД, многие из клиентских OLAP-средств посылают на сервер SQL-запросы и в результате получают агрегатные данные, вычисленные на сервере.

Как правило, OLAP-функциональность реализована в средствах статистической обработки данных и в некоторых электронных таблицах.

Многие средства разработки содержат библиотеки классов или компонентов, позволяющие создавать приложения, реализующие простейшую OLAP-функциональность (такие, например, как компоненты Decision Cube в Borland Delphi и Borland C++ Builder). Помимо этого многие компании предлагают элементы управления ActiveX и другие библиотеки, реализующие подобную функциональность.

Клиентские OLAP-средства применяются, как правило, при малом числе измерений (обычно не более шести) и небольшом разнообразии значений этих параметров – поскольку полученные агрегатные данные должны уместиться в адресном пространстве подобного средства, а их количество растет экспоненциально при увеличении числа измерений.

Многие клиентские OLAP-средства позволяют сохранить содержимое кэша с агрегатными данными в виде файла, для того чтобы не производить их повторное вычисление. Однако нередко такая возможность используется для отчуждения агрегатных данных с целью передачи их другим организациям или для публикации.

Идея сохранения кэша с агрегатными данными в файле получила свое дальнейшее развитие в серверных OLAP-средствах (например, Oracle Express Server или Microsoft OLAP Services), в которых сохранение и изменение агрегатных данных, а также поддержка содержащего их хранилища осуществляются отдельным приложением или процессом, называемым OLAP-сервером. Клиентские приложения могут запрашивать подобное многомерное хранилище и в ответ получать те или иные данные. Некоторые клиентские приложения могут также создавать такие хранилища или обновлять их в соответствии с изменившимися исходными данными.

Преимущества применения серверных OLAP-средств по сравнению с клиентскими OLAP-средствами сходны с преимуществами применения серверных СУБД по сравнению с настольными: в случае применения серверных средств вычисление и хранение агрегатных данных происходят на сервере, а клиентское приложение получает лишь результаты запросов к ним, что позволяет в общем случае снизить сетевой трафик, время выполнения запросов и требования к ресурсам, потребляемым клиентским приложением.

1.3.3.2.3. Технические аспекты многомерного хранения данных

Многомерность в OLAP-приложениях может быть разделена на три уровня:

1. Многомерное представление данных – средства конечного пользователя, обеспечивающие многомерную визуализацию и манипулирование данными; слой многомерного представления

абстрагирован от физической структуры данных и воспринимает данные как многомерные.

2. Многомерная обработка – средство (язык) формулирования многомерных запросов (традиционный реляционный язык SQL здесь оказывается непригодным) и процессор, умеющий обработать и выполнить такой запрос.

3. Многомерное хранение – средства физической организации данных, обеспечивающие эффективное выполнение многомерных запросов.

Первые два уровня в обязательном порядке присутствуют во всех OLAP-средствах. Третий уровень, хотя и является широко распространенным, не обязателен, так как данные для многомерного представления могут извлекаться и из обычных реляционных структур. Процессор многомерных запросов, в этом случае, транслирует многомерные запросы в SQL-запросы, которые выполняются реляционной СУБД.

В любом хранилище данных – и в обычном, и в многомерном – наряду с детальными данными, извлекаемыми из оперативных систем, хранятся и агрегированные показатели (суммарные показатели), такие, как суммы объемов продаж по месяцам, по категориям товаров и т. д. Агрегаты хранятся в явном виде с единственной целью – ускорить выполнение запросов. Ведь, с одной стороны, в хранилище накапливается, как правило, очень большой объем данных, а с другой – аналитиков в большинстве случаев интересуют не детальные, а обобщенные показатели. И если каждый раз для вычисления суммы продаж за год пришлось бы суммировать миллионы индивидуальных продаж, скорость, скорее всего, была бы неприемлемой. Поэтому при загрузке данных в многомерную БД вычисляются и сохраняются все суммарные показатели или их часть.

Тем не менее, использование агрегированных данных чревато недостатками. Основными недостатками являются увеличение объема хранимой информации (при добавлении новых измерений объем данных, составляющих куб, растет экспоненциально) и времени на их загрузку. Причем объем информации может увеличиваться в десятки и даже в сотни раз. Например, в одном из опубликованных стандартных тестов полный подсчет агрегатов для 10 Мб исходных данных потребовал 2,4 Гб, т. е. данные выросли в 240 раз!

Степень увеличения объема данных при вычислении агрегатов зависит от количества измерений куба и структуры этих измерений, т. е. соотношения количества «родителей» и «потомков» на разных уровнях измерения. Для решения проблемы хранения агрегатов применяются сложные схемы, позволяющие при вычислении далеко не всех возможных агрегатов достигать значительного повышения производительности выполнения запросов.

Как исходные, так и агрегатные данные могут храниться либо в

реляционных, либо в многомерных структурах. В связи с этим в настоящее время применяются три способа хранения многомерных данных:

MOLAP (Multidimensional OLAP) – исходные и агрегатные данные хранятся в многомерной базе данных. Хранение данных в многомерных структурах позволяет манипулировать данными как многомерным массивом, благодаря чему скорость вычисления агрегатных значений одинакова для любого из измерений. Однако в этом случае многомерная база данных оказывается избыточной, так как многомерные данные полностью содержат исходные реляционные данные.

Эти системы обеспечивают полный цикл OLAP-обработки. Они либо включают в себя, помимо серверного компонента, собственный интегрированный клиентский интерфейс, либо используют для связи с пользователем внешние программы работы с электронными таблицами.

ROLAP (Relational OLAP) – исходные данные остаются в той же реляционной базе данных, где они изначально и находились. Агрегатные же данные помещают в специально созданные для их хранения служебные таблицы в той же базе данных.

HOLAP (Hybrid OLAP) – исходные данные остаются в той же реляционной базе данных, где они изначально находились, а агрегатные данные хранятся в многомерной базе данных.

Некоторые OLAP-средства поддерживают хранение данных только в реляционных структурах, некоторые – только в многомерных. Однако большинство современных серверных OLAP-средств поддерживают все три способа хранения данных. Выбор способа хранения зависит от объема и структуры исходных данных, требований к скорости выполнения запросов и частоты обновления OLAP-кубов.

1.3.3.3. Интеллектуальный анализ данных (Data Mining)

Термин Data Mining обозначает процесс поиска корреляций, тенденций и взаимосвязей посредством различных математических и статистических алгоритмов: кластеризации, регрессионного и корреляционного анализа и т. д. для систем поддержки принятия решений. При этом накопленные сведения автоматически обобщаются до информации, которая может быть охарактеризована как знания.

В основу современной технологии Data Mining положена концепция шаблонов, отражающих закономерности, свойственные подвыборкам данных и составляющие так называемые скрытые знания (рис. 1.21).

Поиск шаблонов производится методами, не использующими никаких априорных предположений об этих подвыборках. Важной особенностью Data Mining является нестандартность и неочевидность разыскиваемых шаблонов. Иными словами, средства Data Mining отличаются от инструментов статистической обработки данных и средств OLAP тем, что вместо проверки заранее предполагаемых пользователями взаимосвязей

между данными, они на основании имеющихся данных способны самостоятельно находить такие взаимосвязи, а также строить гипотезы об их характере.



Рис. 1.21. Уровни знаний, извлекаемых из данных

В общем случае процесс интеллектуального анализа данных (Data Mining) состоит из трёх стадий (рис. 1.22):

- выявление закономерностей (свободный поиск);
- использование выявленных закономерностей для предсказания неизвестных значений (прогностическое моделирование);
- анализ исключений, предназначенный для выявления и толкования аномалий в найденных закономерностях.

Иногда в явном виде выделяют промежуточную стадию проверки достоверности найденных закономерностей между их нахождением и использованием (стадия валидации).

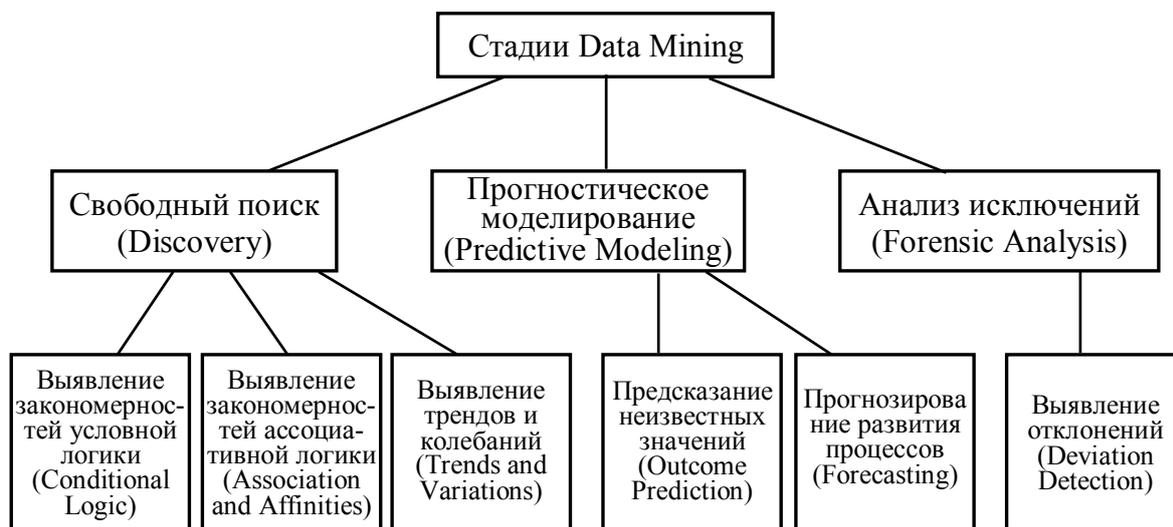


Рис. 1.22. Стадии процесса интеллектуального анализа данных

Выделяют пять стандартных типов закономерностей, выявляемых методами Data Mining:

Ассоциация позволяет выделить устойчивые группы объектов, между

которыми существуют неявно заданные связи. Частота появления отдельного предмета или группы предметов, выраженная в процентах, называется распространенностью. Низкий уровень распространенности (менее одной тысячной процента) говорит о том, что такая ассоциация не существенна. Ассоциации записываются в виде правил: $A \Rightarrow B$, где A – посылка, B – следствие. Для определения важности каждого полученного ассоциативного правила необходимо вычислить величину, которую называют доверительность A к B (или взаимосвязь A и B). Доверительность $\delta(A/B)$ показывает, как часто при появлении A появляется B , и рассчитывается как $\delta(A/B) = \varepsilon(A \cap B) / \varepsilon(A)$, где $\varepsilon(A \cap B)$ – распространенность совместного появления A и B ; $\varepsilon(A)$ – распространенность A . Например, если $\delta(A/B) = 20\%$, то это значит, что при покупке товара A в каждом пятом случае приобретается и товар B . Необходимо отметить, что если $\varepsilon(A) \neq \varepsilon(B)$, то $\delta(A/B) \neq \delta(B/A)$. В самом деле, покупка компьютера влечет за собой покупку дисков, но покупка дисков не ведет к покупке компьютера. Важной характеристикой ассоциации является мощность, которая рассчитывается по формуле $M(A/B) = \delta(A/B) / \varepsilon(B)$. Чем больше мощность, тем сильнее влияние, которое наличие A оказывает на появление B .

Типичным примером применения ассоциации является анализ структуры покупок. Например, при проведении исследования в супермаркете можно установить, что 65 % купивших картофельные чипсы берут также и «кока-колу», а при наличии скидки за такой комплект «колу» приобретают в 85 % случаев. Подобные результаты представляют ценность при формировании маркетинговых стратегий.

Последовательность – это метод выявления ассоциаций во времени. В данном случае определяются правила, которые описывают последовательное появление определенных групп событий. Такие правила необходимы для построения сценариев. Кроме того, их можно использовать, например, для формирования типичного набора предшествующих продаж, которые могут повлечь за собой последующие продажи конкретного товара.

Классификация – инструмент обобщения. Она позволяет перейти от рассмотрения единичных объектов к обобщенным понятиям, которые характеризуют некоторые совокупности объектов и являются достаточными для распознавания объектов, принадлежащих этим совокупностям (классам). Суть процесса формирования понятий заключается в нахождении закономерностей, свойственных классам. Для описания объектов используются множества различных признаков (атрибутов). Проблема формирования понятий по признаковым описаниям была сформулирована М.М. Бонгартом. Ее решение базируется на применении двух основных процедур: обучения и проверки. В процедурах

обучения строится классифицирующее правило на основе обработки обучающего множества объектов. Процедура проверки (экзамена) состоит в использовании полученного классифицирующего правила для распознавания объектов из новой (экзаменационной) выборки. Если результаты проверки признаны удовлетворительными, то процесс обучения заканчивается, в противном случае классифицирующее правило уточняется в процессе повторного обучения.

Кластеризация – это распределение информации (записей) из БД по группам (кластерам) или сегментам с одновременным определением этих групп. В отличие от классификации здесь для проведения анализа не требуется предварительного задания классов.

Прогнозирование временных рядов является инструментом для определения тенденций изменения атрибутов рассматриваемых объектов с течением времени. Анализ поведения временных рядов позволяет прогнозировать значения исследуемых характеристик.

Для решения таких задач используются различные методы и алгоритмы Data Mining. Ввиду того, что Data Mining развивалась и развивается на стыке таких дисциплин, как статистика, теория информации, машинное обучение, теория баз данных, вполне закономерно, что большинство алгоритмов и методов Data Mining были разработаны на основе различных методов из этих дисциплин.

Из многообразия существующих методов исследования данных можно выделить следующие:

- *регрессионный, дисперсионный и корреляционный анализ* (реализован в большинстве современных статистических пакетов, в частности, в продуктах компаний SAS Institute, StatSoft и др.);

- *методы анализа* в конкретной предметной области, базирующиеся на эмпирических моделях (часто применяются, например, в недорогих средствах финансового анализа);

- *нейросетевые алгоритмы* – метод имитации процессов и явлений, позволяющий воспроизводить сложные зависимости. Метод основан на использовании упрощенной модели биологического мозга и заключается в том, что исходные параметры рассматриваются как сигналы, преобразующиеся в соответствии с имеющимися связями между «нейронами», а в качестве ответа, являющегося результатом анализа, рассматривается отклик всей сети на исходные данные. Связи в этом случае создаются с помощью так называемого обучения сети посредством выборки большого объема, содержащей как исходные данные, так и правильные ответы. Нейронные сети широко применяются для решения задач классификации;

- *нечеткая логика* применяется для обработки данных с размытыми значениями истинности, которые могут быть представлены разнообразными лингвистическими переменными. Нечеткое

представление знаний широко применяется для решения задач классификации и прогнозирования, например, в системе XpertRule Miner (Attar Software Ltd., Великобритания), а также в AIS, NeuFuz и др;

- *индуктивные выводы* позволяют получить обобщения фактов, хранящихся в БД. В процессе индуктивного обучения может участвовать специалист, поставляющий гипотезы. Такой способ называют обучением с учителем. Поиск правил обобщения может осуществляться без учителя путем автоматической генерации гипотез. В современных программных средствах, как правило, сочетаются оба способа, а для проверки гипотез используются статистические методы. Примером системы с применением индуктивных выводов является XpertRule Miner, разработанная фирмой Attar Software Ltd. (Великобритания);

- рассуждения на основе *аналогичных случаев* (метод «ближайшего соседа») (Case-based reasoning – CBR) основаны на поиске в БД ситуаций, описания которых сходны по ряду признаков с заданной ситуацией. Принцип аналогии позволяет предполагать, что результаты похожих ситуаций также будут близки между собой. Недостаток этого подхода заключается в том, что здесь не создается каких-либо моделей или правил, обобщающих предыдущий опыт. Кроме того, надежность выводимых результатов зависит от полноты описания ситуаций, как и в процессах индуктивного вывода. Примерами систем, использующих CBR, являются: KATE Tools (Acknosoft, Франция), Pattern Recognition Workbench (Unica, США);

- *деревья решений* – метод структурирования задачи в виде древовидного графа, вершины которого соответствуют продукционным правилам, позволяющим классифицировать данные или осуществлять анализ последствий решений. Этот метод дает наглядное представление о системе классифицирующих правил, если их не очень много. Простые задачи решаются с помощью этого метода гораздо быстрее, чем с использованием нейронных сетей. Для сложных проблем и для некоторых типов данных деревья решений могут оказаться неприемлемыми. Кроме того, для этого метода характерна проблема значимости. Одним из последствий иерархической кластеризации данных является отсутствие большого числа обучающих примеров для многих частных случаев, в связи с чем классификацию нельзя считать надежной. Методы деревьев решений реализованы во многих программных средствах, а именно: C5.0 (RuleQuest, Австралия), Clementine (Integral Solutions, Великобритания), SIPINA (University of Lyon, Франция), IDIS (Information Discovery, США);

- *эволюционное программирование* – поиск и генерация алгоритма, выражающего взаимосвязь данных, на основании изначально заданного алгоритма, модифицируемого в процессе поиска; иногда поиск взаимосвязей осуществляется среди каких-либо определенных видов функций (например, полиномов);

- *алгоритмы ограниченного перебора*, вычисляющие частоты комбинаций простых логических событий в подгруппах данных.

1.3.3.4. Интеграция OLAP и Data Mining

Оперативная аналитическая обработка (OLAP) и интеллектуальный анализ данных (Data Mining) – две составные части процесса поддержки принятия решений. Однако сегодня большинство систем OLAP заостряет внимание только на обеспечении доступа к многомерным данным, а большинство средств Data Mining, работающих в сфере закономерностей, имеют дело с одномерными перспективами данных. Для увеличения эффективности обработки данных для систем поддержки принятия решений эти два вида анализа должны быть объединены.

В настоящее время появляется составной термин «OLAP Data Mining» (многомерный интеллектуальный анализ) для обозначения такого объединения (рис. 1.23).

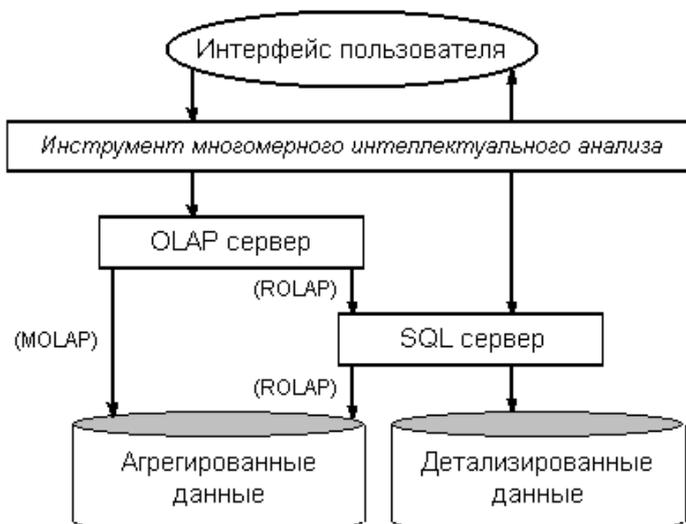


Рис. 1.23. Архитектура системы многомерного интеллектуального анализа данных

Существует три основных способа формирования «OLAP Data Mining»:

- «Cubing then mining». Возможность выполнения интеллектуального анализа должна обеспечиваться над любым результатом запроса к многомерному концептуальному представлению, то есть над любым фрагментом любой проекции гиперкуба показателей.
- «Mining then cubing». Подобно данным, извлечённым из хранилища, результаты интеллектуального анализа должны представляться в гиперкубической форме для последующего многомерного анализа.
- «Cubing while mining». Этот гибкий способ интеграции позволяет автоматически активизировать однотипные механизмы интеллектуальной обработки над результатом каждого шага многомерного анализа (перехода

между уровнями обобщения, извлечения нового фрагмента гиперкуба и т. д.).

Контрольные вопросы

1. Какие задачи решает корпоративные ИС?
2. Перечислите виды корпоративных ИС. Дайте краткую характеристику.
3. Что такое транзакция? Какие особенности OLTP-систем Вы знаете? Перечислите свойства OLTP-систем. Назовите принципы разработки OLTP-систем.
4. Дайте определение хранилищу данных. Для чего необходима очистка данных в хранилище данных? Какие функции выполняют приложения очистки?
5. Перечислите элементы хранилища данных. Дайте им характеристику.
6. Перечислите свойства хранилища данных. Охарактеризуйте каждое свойство.
7. Какие модели хранилища данных и витрин данных Вы знаете? Дайте им краткую характеристику.
8. Для чего необходимы системы поддержки принятия решений (СППР). Перечислите алгоритмы СППР.
9. Перечислите основные варианты архитектуры СППР. Назовите их достоинства и недостатки.
10. Что такое OLAP. Охарактеризуйте многомерную модель данных в OLAP.
11. Что такое Data Mining. Чем Data Mining отличается от OLAP. Назовите основные стадии Data Mining. Перечислите основные закономерности, выявляемые методами Data Mining.

2. Стандарты корпоративных ИС

В середине прошлого века быстрый рост производства с существующим способом его планирования привел к заметному снижению темпов производства. Так как объемы производства были огромны, то возникали различные проблемы, связанные с циклом, временем производства, отгрузкой готового товара, а также доставкой товара до конечного потребителя. Стало понятно, что необходимо поднять планирование производства на новый качественный уровень. Повсеместное внедрение электронно-вычислительной техники с начала 60-х гг. XX в. обусловило тенденцию развития планирования. Таким образом, стали формироваться концепции, которые впоследствии перешли в разряд стандартов не только для производства, но и всего бизнеса в целом. Так появились стандарты MRP, MRPII и ERP. Затем такие концепции были развиты и использованы в стратегическом планировании бизнеса и интеграции конечного потребителя в процесс производства. В результате появились стандарты MPC, SCRP.

2.1. MRP (Material Requirements Planning)

С целью предотвращения проблем с планированием производства, была разработана методология планирования потребности в материалах MRP (Material Requirements Planning). Реализация системы, работающей по этой методологии представляет собой компьютерную программу, позволяющую оптимально регулировать поставки комплектующих в производственный процесс, контролируя запасы на складе и саму технологию производства. Главной задачей MRP является обеспечение гарантии наличия необходимого количества требуемых материалов-комплектующих в любой момент времени в рамках срока планирования, наряду с возможным уменьшением постоянных запасов, а следовательно, разгрузкой склада.

Основными преимуществами использования подобной системы в производстве являются:

- гарантия наличия требуемых комплектующих и уменьшение временных задержек в их доставке, и, следовательно, увеличение выпуска готовых изделий без увеличения числа рабочих мест и нагрузок на производственное оборудование;

- уменьшение производственного брака в процессе сборки готовой продукции возникающего из-за использования неправильных комплектующих;

- упорядочивание производства, ввиду контроля статуса каждого материала, позволяющего однозначно отслеживать весь его конвейерный путь, начиная от создания заказа на данный материал, до его положения в уже собранном готовом изделии. Также благодаря этому достигается полная достоверность и эффективность производственного учета.

Основная цель MRP-системы формировать, контролировать и при необходимости изменять даты необходимого поступления заказов таким образом, чтобы все материалы, необходимые для производства поступали одновременно.

На практике MRP-система представляет собой компьютерную программу (рис. 2.1), которая состоит из трех элементов.

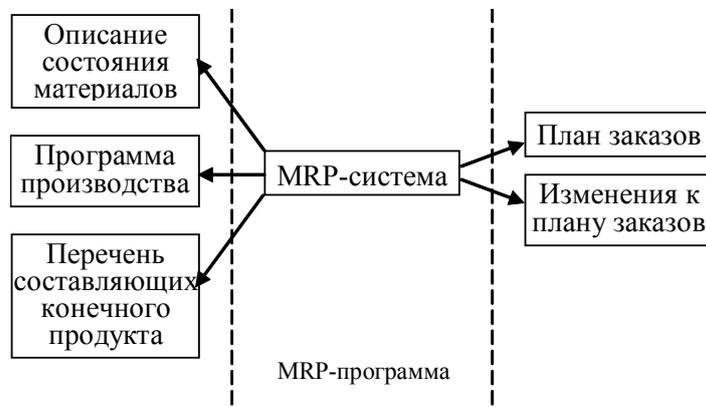


Рис. 2.1. Входные элементы и результаты работы MRP-программы

Описание состояния материалов – является основным входным элементом MRP-программы. В нем должна быть отражена максимально полная информация о всех материалах-компонентах, необходимых для производства конечного продукта: статус каждого материала, (находится на складе или в производстве), описания, его запасов, расположения, цены, возможных задержек поставок, реквизиты поставщиков. Информация по всем вышеперечисленным позициям должна быть заложена отдельно по каждому материалу, участвующему в производственном процессе.

Программа производства – представляет собой оптимизированный график распределения времени для производства необходимой партии готовой продукции за планируемый период или диапазон периодов. Параллельно проводится проверка наличия необходимой мощности производства на изготовление требуемой партии.

Перечень составляющих конечного продукта – это список и количество материала, требуемого для производства, а также описание структуры конечного продукта, который содержит в себе полную информацию по технологии его сборки.

2.2. MRP II (Manufactory Resource Planning)

С целью увеличить эффективность планирования, в конце 70-х гг. Оливер Уайт и Джордж Плосл предложили идею воспроизведения замкнутого цикла в MRP-системах. Идея заключалась в учете большего количества факторов при проведении планирования, путем введения дополнительных функций. К базовым функциям планирования производственных мощностей и потребностей в материалах было предложено добавить ряд дополнительных функций: контроль

соответствия количества произведенной продукции количеству использованных в процессе сборки комплектующих, составление регулярных отчетов о задержках заказов, об объемах и динамике продаж продукции, о поставщиках и т. д. Термин «замкнутый цикл» отражает основную особенность модифицированной системы, заключающуюся в анализе и учете отчетов производства. На основании результатов анализа, при необходимости, возможно изменение программы производства и плана заказов. Таким образом, введение дополнительных функций обеспечило системе гибкость планирования по отношению к внешним факторам, таким как уровень спроса, состояние дел у поставщиков и пр.

Дальнейшее усовершенствование привело к трансформации системы MRP с замкнутым циклом в расширенную модификацию, которую впоследствии назвали MRP II (Manufactory Resource Planning), ввиду идентичности аббревиатур. Эта система была создана для эффективного планирования всех ресурсов производственного предприятия, в том числе финансовых и кадровых. MRP II представляет собой интеграцию большого количества отдельных модулей, таких как планирование бизнес-процессов, планирование потребностей в материалах, планирование производственных мощностей, планирование финансов, управление инвестициями и т. д. Результаты работы каждого из модуля анализируются всей системой в целом, что собственно и обеспечивает ее гибкость по отношению к внешним факторам. Именно это свойство является краеугольным камнем современных систем планирования, поскольку большое количество производителей производят продукцию с заведомо коротким жизненным циклом, требующую регулярных доработок. В таком случае появляется необходимость в автоматизированной системе, которая позволяет оптимизировать объемы и характеристики выпускаемой продукции, анализируя текущий спрос и положение на рынке в целом.

В основу стандарта MRP II положена иерархия процессов планирования, причем планы нижних уровней зависят от планов уровней, находящихся выше, а результаты выполнения планов нижнего уровня оказывают влияние на планы верхнего уровня. В рамках MRP II определены следующие процессы:

1. Стратегическое планирование. Задачей стратегического планирования является составление плана развития предприятия на срок от одного до пяти лет в зависимости от макроэкономических показателей, тенденций развития экономики, изменения технологий и состояния рынка.

2. Бизнес-планирование. Заключается в создании бизнес-планов (обычно годовых), регламентирующих вопросы производства и сбыта продукции, развития основных средств, потребности в капиталовложениях и субсидиях.

3. Планирование объемов продаж и производства. Регламентирует процессы производства и сбыта продукции по номенклатурным группам.

Составляемый в результате этого процесса план является отправным документом при определении потребностей предприятия в материальных, трудовых и финансовых ресурсах.

4. Планирование ресурсов. В рамках данной задачи осуществляется долгосрочное планирование объема ключевых ресурсов предприятия.

5. Составление план-графика производства. Здесь осуществляется наложение плана производства продукции предприятия на шкалу времени, определение того, что, когда и в каких количествах будет произведено.

6. Планирование производственных мощностей. Одной из составляющих эффективного процесса производства является планирование производственных мощностей предприятия, определения необходимых затрат на достижение мощностей, необходимых для выполнения разработанного план-графика производства.

7. Планирование потребностей в материалах. На данном этапе осуществляется определение потребностей в материалах и комплектующих предприятия, из множества предприятий-партнеров определяются необходимые субподрядчики и поставщики, составляется необходимый график взаимодействия с ними.

8. Планирование потребностей в распределении. Здесь осуществляется координация спроса, предложения и ресурсов между подразделениями одной или нескольких компаний, участвующих в технологическом процессе.

2.3. ERP (Enterprise Requirements Planning)

В последние годы системы планирования класса MRPII в интеграции с модулем финансового планирования FRP (Finance Requirements Planning) получили название систем бизнес-планирования ERP (Enterprise Requirements Planning). Таким образом, ERP-системы обеспечили набор интегрированных модулей приложений, работающих с единой комплексной базой данных. При обработке транзакции в одном приложении, соответствующие данные и результаты автоматически обновлялись в оставшихся программах. Таким образом, удалось избежать операций экспорта и импорта, повысить целостность данных и интегрировать бизнес-процессы.

Самой видной фигурой на рынке ERP-систем является немецкая компания SAP AG с продуктом R/3. Она же считается и родоначальницей рынка ERP-систем в том объеме, в котором они существуют сегодня. К числу других зарубежных фирм, занимающихся разработкой ERP-систем можно отнести: PeopleSoft, Oracle, Baan и J.D. Edwards.

На российском рынке продаж ERP-систем основными являются: 1С Предприятие, Система «Галактика», КИС «Фрегат», КИС «Лагуна», КИС «Парус».



1С Предприятие Система "Галактика" КИС "Фрегат" КИС "Лагуна" КИС "Парус"

Эти системы ERP типа являются комплексными системами автоматизации управления предприятием. Каждая из них обладает большими функциональными возможностями и является основой единого информационного пространства предприятия. Основным отличием (за исключением специфических) является размер предприятий в которых они внедряются (рис. 2.2).

Предприятие				
Мелкое	Малое	Среднее	Крупное	Оч. крупное
	+	+	+	

Предприятие				
Мелкое	Малое	Среднее	Крупное	Оч. крупное
	+	+		

Предприятие				
Мелкое	Малое	Среднее	Крупное	Оч. крупное
			+	+

Предприятие				
Мелкое	Малое	Среднее	Крупное	Оч. крупное
	+	+	+	

Предприятие				
Мелкое	Малое	Среднее	Крупное	Оч. крупное
		+	+	+

Рис. 2.2. Российские ERP-системы

2.4. MPC (Management Planning and Control)

Хотя ERP-системы и созданы для повседневного управления бизнеса, однако они практически не пригодны для его стратегического планирования: расширения возможностей деловой активности, роста доли на рынке, увеличении продаж и эффективности бизнеса, а также повышении ценности бизнеса в целом. Комбинация этих процессов, объединенных в одну систему, стала называться планированием и управлением менеджмента (Management Planning and Control, MPC).

Решением является применение интегрированного пакета приложений, предназначенных для поддержки оптимальных методов стратегического планирования, бюджетирования, прогнозирования, финансовой консолидации, управленческой отчетности и анализа. Пакет MPC-приложений позволяет предприятию создавать планы и бюджеты, управлять их исполнением, оценивать фактическое исполнение по сравнению с запланированным, выделять исключения и моделировать решения на будущее.

В основе MPC-системы лежит централизованная реляционная база,

поддерживающая полный пакет приложений. Здесь хранятся не только данные и результаты бюджетирования, но также ключевые бизнес-факторы, полученные на основе стратегического планирования, а также реальных, прогнозируемых данных и результатов. И если в приложении используются отдельные файлы, хранящиеся на различных компьютерах, то данные, вводимые в МРС-базу, и результаты, рассчитываемые по этим данным, сразу же становятся доступными для дальнейшего анализа, поскольку уже не нужно перемещать эти данные и результаты из одного приложения в другое. Реляционность МРС-системы позволяет масштабировать пакет для поддержки больших объемов данных и более широкого круга пользователей, а также легко интегрировать его с существующими ERP-системами и другими операционными приложениями. Технически МРС-базу можно характеризовать как реляционную витрину данных. В отличие от реляционной базы, предназначенной для поддержки обработки транзакций, реляционная витрина разработана для поддержки принятия решений.

В целом, это означает, что лежащая в основе реляционная схема обеспечивает приложения пользователей многомерным представлением данных и результатов. Такое представление позволяет осуществлять планирование, бюджетирование, отслеживание и прогнозирование по разным измерениям. Например, пользователь может составлять бюджет продаж, расходов и доходов в определенной валюте, для некоторого периода времени по определенным продуктам, местам расположения и каналам распространения.

Существуют четкие критерии оценки МРС-продуктов от любых других финансовых аналитических приложений:

1. **Интегрированные процессы.** В системе поддерживается планирование, бюджетирование, консолидация, административная отчетность и анализ в одном замкнутом приложении. Пользователь имеет возможность переключаться между процессами не меняя рабочей среды и не перемещая данные, а руководство – быстро реагировать на все изменения, оценивая альтернативы, регулируя планы и информируя пользователей, без дополнительного изучения и поддержки множества технологий и приложений.

2. **Общие бизнес-правила.** Система имеет общий набор измерений, бизнес-элементов и правил. В результате при изменении структуры или элемента связанные с ними отчеты и анализы будут обновляться автоматически. По тем же причинам всегда будет обрабатываться только один набор данных, даже если потребуется несколько версий. Это означает, что любое число, стоящее в плане, бюджете или в наличности, хранится и обрабатывается только в одном месте.

3. **Встроенная финансовая логика.** Система автоматически генерирует финансовые отчеты и выполняет связанный с ними анализ, не

требуя вмешательства программиста. По мере того как пользователь манипулирует данными, система должна их корректно обрабатывать.

4. **Моделирование будущего.** Все бизнес-правила должны уметь обращаться к любому элементу в базе данных, а ПО – выполнять консолидацию по нескольку раз. Используя подобные правила, система сможет выполнить ряд задач (как, например, распределение и расчет долей участия дочерних компаний и пр.), которые иным способом решить не удастся.

5. **Временная логика.** В системе обрабатываются различные временные показатели, относящиеся к процессам, отчетности и другим функциям, таким как: периодичность учета, скользящие бюджеты (rolling budgets) и показатели за текущий год. Применение этой логики упрощает настройку систем и их перевод на непрерывное планирование (continuous planning).

6. **Стимуляция взаимодействия.** В любой момент времени без предварительного уведомления пользователь имеет online-доступ к данным, которые он без всяких ограничений сможет просматривать и анализировать по всем допустимым измерениям: проект, продукт, направление деятельности и т. д. Кроме того, необходимо располагать возможностью вращать и представлять измерения во вложенном виде, а также выполнять детализацию внутри модели.

7. **Управляемый анализ.** МРС-системы обращают внимание пользователя на исключения и предоставления ему специальных инструментов для детального анализа отклонений. Уведомления об исключениях приходят пользователю автоматически.

8. **Централизованная база данных.** МРС приложения работают на централизованной базе данных, не используя специальные файловые структуры, которые часто применяются в современных системах. Эта база основывается на широко распространенной реляционной технологии – IBM (Oracle) или Microsoft (OLAP), что обеспечивает масштабируемость и надежность.

2.5. CSRP (Customer Synchronized Resource Planning)

В условиях жесткой конкуренции современные ИС могут служить не только в качестве средств поддержки тех или иных аспектов деятельности предприятия, но также и для достижения различного рода конкурентных преимуществ. В условиях современного производства, когда качество продукции становится повсеместным и, таким образом, перестает быть определяющим фактором конкурентного превосходства, возникает вопрос о том, какие именно параметры продукта определяют его потребительскую ценность. От производителя в современном мире требуется создание товаров и услуг, соответствующих не абстрактным требованиям обобщенного рынка, а конкретным потребностям каждого уникального

покупателя. В связи с этим задачей каждого производителя становится привлечение и удержание каждого покупателя, при сохранении необходимого соотношения цена/эффективность и высокого качества товаров.

Для повышения эффективности работы предприятий производители должны постоянно разрабатывать новые технологии и бизнес-процессы, которые позволяли бы им удовлетворять индивидуальные покупательские нужды и ожидания, отвечать на эти нужды товарами и услугами, которые представляют уникальную ценность для каждого покупателя. В процессе достижения этой цели для предприятий необходимо совершить частичное изменение в стратегии и интегрировать покупателя в центр процесса планирования деятельности организации. Интеграция покупателя с ключевыми бизнес-процессами предприятия изменит его стратегию и потребует новой модели управления деятельностью – планирование ресурсов, синхронизированное с покупателем.

Модель планирования ресурсов предприятия на основе синхронизации с покупателем (CSRP, Customer Synchronized Resource Planning) охватывает практически весь жизненный цикл производимого изделия, включая предпродажный, производственный, логистический и послепродажный циклы. Сущность концепции CSRP состоит в том, что при планировании и управлении компанией можно и нужно учитывать не только основные производственные и материальные ресурсы предприятия, но и все те ресурсы, которые обычно рассматриваются как «вспомогательные» или «накладные». Это все ресурсы, потребляемые во время маркетинговой и «текущей» работы с клиентом, послепродажного обслуживания проданных товаров, перевалочных и обслуживающих операций, а также внутрицеховых ресурсов, т. е. всего «функционального» жизненного цикла товара. Это приобретает решающее значение для повышения конкурентоспособности предприятия в отраслях, где жизненный цикл товара невелик и требуется оперативно реагировать на изменение желаний потребителя.

CSRP – это первая бизнес-методология, которая включает деятельность, ориентированную на интересы покупателя, в ядро системы управления бизнесом. Впервые предложена методология ведения бизнеса, основанная на текущей информации о покупателе. CSRP перемещает фокус внимания с планирования производства к планированию заказов покупателей. Информация о клиентах и услуги внедряются в основу деятельности организации. Производственное планирование не просто расширяется, а замещается требованиями клиентов, поступающими из подразделений, ориентированных на работу с покупателями.

Таким образом, при использовании модели CSRP работа с клиентами становятся важным звеном деятельности предприятия, центром управления всей организацией. При этом подразделение технической

поддержки отвечает за доведение необходимой информации о покупателях к исполнительным центрам организации.

Для CSRP характерны следующие положения:

- средства поддержки пользователей сливаются с ключевыми приложениями планирования, производства и управления. Необходимая информация о покупателях и товарах заранее поставляется подразделениям, отвечающим за производство, продажи, исследования и развитие, а также другим подразделениям;

- технологии, основанные на Web, расширяют возможности поддержки покупателей, включая удалённый, круглосуточный сервис по принципу самообслуживания. Ключевые исполнительные системы автоматически обновляются, обеспечивая наибо́льший ответ на запросы покупателей;

- подразделения поддержки покупателей становятся центрами продаж и поддержки. Интеграция с продажами, обработкой заказов и управлением обеспечивает необходимую базу и инфраструктуру для распространения деятельности по поддержке покупателей на область продаж, обеспечивая канал для продвижения новых и сопутствующих продуктов и услуг.

Результатом успешного применения CSRP является повышение качества товаров, снижение времени поставки, повышение потребительской ценности продукции, и т. д., а в результате этого – снижение производственных издержек, но что более важно, способствует развитию инфраструктуры для создания индивидуализируемых, конфигурируемых решений, улучшая сервис и обратную связь с покупателями. В результате появляется не технологическая эффективность, которая обеспечивает лишь временное конкурентное преимущество, а способность создавать продукты, удовлетворяющие разнообразным потребностям покупателя, что приводит к получению устойчивого конкурентного преимущества.

2.6. Системы автоматизации документооборота

Системы документооборота берут свое начало с конца XVIII в. с появлением фотоаппарата, когда стало возможно создавать уменьшенные копии документов. Создание микрофиши дала реальный, качественный прорыв в стоимости хранения и копирования документов, но с точки зрения скорости поиска и коллективного использования документов все осталось на уровне бумажных документов.

Появление компьютерных систем с коллективным доступом позволило достигнуть прорыва и в этих областях. Произошла трансформация от микрографических документов к электронным образам бумажных документов, от микрографических систем к так называемым imaging-системам. Расширенная система поиска, возможность получать доступ к одному документу нескольким пользователям одновременно,

возможность удаленной и распределенной работы при сохранении незначительной стоимости хранения и копирования информации предопределило тенденцию замены микрографических документов на электронные.

В начале 90-х гг. прошлого века появился новый класс систем, названный системами управления документами, предназначенный для управления электронными документами, отличными от их образов. В основном это были файлы текстовых процессоров и запросы на такого рода решения. Кроме функций хранения и поиска, присущих imaging-системам, в системах управления документами появились функции контроля жизненного цикла документа.

2.6.1. Типы носителей информации в документообороте

Документ – это некая обособленная часть информации представленная на определенном носителе. В настоящее время можно выделить три главных типа носителей информации:

1. Бумажные носители. К этому типу относятся все документы, которые созданы на бумаге.

2. Микрофиши – практически то же самое, что и бумага с точки зрения поиска необходимого документа, но занимают меньше места (до 72 страниц на одном микрофисе). Микрофиши могут быть более тесно интегрированы с электронными средствами и доступны по компьютерной сети.

3. Электронные носители информации, которые также внутри себя могут быть подразделены на ряд подтипов (жесткие диски, оптические диски, магнитооптические диски и т. д.). Кроме того, для электронных документов важен формат, в котором они сохранены на электронном носителе информации.

Очевидно, что один и тот же документ может существовать в различных ипостасях, на различных носителях информации и переходить с одного носителя на другой. Применение того или иного носителя информации определяется следующими факторами:

1. Стоимость хранения информации прямо пропорциональна количеству документов и зависит от стоимости хранения одного документа. В настоящее время термин стоимость хранения информации все чаще применяется не для одной страницы документа (что используется для бумажных документов), а к одному мегабайту хранимой информации.

2. Стоимость (время) поиска необходимой информации – определяется системой индексации документов. В том случае если система индексации отсутствует, то скорость поиска также прямо пропорциональна количеству хранимых документов.

3. Стоимость коллективного использования информации определяет количество копий документа, необходимых для коллективного

использования при одновременном доступе.

4. Стоимость (время) передачи документа от одного рабочего места к другому.

2.6.2. Подходы к организации хранения электронных документов

На сегодняшний день применяется два подхода к организации хранения электронных документов. Первый состоит в хранении тела документов в файловой системе, второй – в хранении документов в реляционной или специализированной базе данных. Вторым подходом хотя и обладает большей степенью защиты собственно документов, но несет в себе следующие ключевые недостатки:

1. Трудности с поддержкой носителей информации, отличных от жестких дисков (только СУБД Informix поддерживает магнитооптические накопители) и практическая невозможность построения гетерогенных систем хранения.

2. При работе с приложениями, в которых создаются и изменяются электронные документы тела документов в любом случае проходят через файловую систему. В связи с тем, что приложение не умеет работать напрямую с базами данных это число операций записи и считывания с жесткого диска увеличивается. При больших размерах тел документов это серьезно влияет на скорость работы.

2.6.3. Требования, предъявляемые к системе документооборота

1. Масштабируемость – способность системы наращивать свою мощность. Выполнение такого требования может быть обеспечено с помощью поддержки промышленных серверов баз данных производства таких компаний, как Sybase, Oracle, Informix и др.

2. Распределенность – способность системы обеспечивать работу распределенных рабочих мест. Основные проблемы при работе с документами возникают в территориально-распределенных организациях, поэтому архитектура систем документооборота должна поддерживать взаимодействие распределенных рабочих мест. Причем распределенные рабочие места могут объединяться самыми разнообразными по скорости и качеству каналами связи. Такая архитектура системы должна поддерживать взаимодействие с удаленными пользователями.

3. Модульность – деление системы на соответствующие модули, отвечающие за работу отдельных частей системы. Тогда в случае необходимости, можно всегда увеличить спектр решаемых задач, добавив необходимые модули.

4. Интегрируемость – система документооборота может быть объединена с другими системами поддержки принятия решений.

2.6.4. Задачи, решаемые системами документооборота

Спектр задач системы автоматизации определяется стадией жизненного цикла документа, которую необходимо поддерживать. Жизненный цикл документа состоит из двух основных стадий:

1. Стадия разработки документа, которая состоит из разработки содержания документа, оформления документа и утверждения документа.

Если документ находится на стадии разработки, то он считается неопубликованным, и права на документ определяются правами доступа конкретного пользователя.

2. Стадия опубликованного документа, которая состоит из активного доступа, архивного документа, краткосрочного хранения, долгосрочного хранения и уничтожения документа.

Когда документ переходит на вторую стадию, он считается опубликованным, тогда права на документ остаются только одни – доступ на чтение. В качестве примера опубликованного документа можно привести шаблон стандартного бланка предприятия. Кроме права доступа на чтение могут существовать права на перевод опубликованного документа в стадию разработки.

В зависимости от конкретной стадии жизненного цикла документа, с которым имеет дело архивная система, архивы подразделяются на статические архивы и динамические архивы.

Статические архивы документов (либо просто архивы) – системы, которые имеют дело только с опубликованными документами.

Динамические архивы документов (либо системы управления документами) – системы, имеющие дело, как с опубликованными документами, так и с документами, которые находятся в разработке.

2.6.4.1. Задачи статических архивов

Архив предприятия – это комплекс программного и аппаратного обеспечения, предназначенный для решения следующих задач:

1. Организация хранения электронных документов.

Необходимо обеспечивать хранение произвольного количества электронных документов на разнообразных носителях. Носители электронных документов характеризуются двумя основными параметрами: стоимостью хранения одного мегабайта информации и скоростью доступа к информации. Причем эти два параметра обратно пропорциональны друг другу – в зависимости от решаемых задач, необходимо находить оптимальное соответствие и выбирать определенный носитель информации. Также на выбор носителя влияет срок хранения информации. Иногда для ряда задач необходимо создавать систему хранения, которая состоит из разнотипных носителей. Например, для оперативного доступа требуется применение высокоскоростных жестких дисков, а для архивного хранения достаточно роботизированных библиотек оптических дисков.

2. Организация учета бумажных и микрографических документов.

Архивная система кроме работы с электронными документами должна обеспечивать работу с бумажными и микрографическими документами. В этом случае система должна хранить только электронную карточку на документ данного типа и поддерживать контроль стандартных архивных операций: выдача документа, возврат документа и пр.

3. Организация поиска документов.

Наряду с организацией хранения документов необходимо их быстро и эффективно искать. Существует два подхода к поиску документов.

Первый подход состоит в том, что в процессе поиска ищется документ, который точно существует в системе, и задача состоит в том, чтобы процесс поиска свелся к нахождению требуемого документа или документов. Этот подход применяется в 90 % всех случаев поиска. *Второй подход* состоит в том, что ищутся все документы по интересующему вопросу. Для этого подхода присущи такие понятия, как полнота – соответствие между найденными документами по данному запросу и действительному списку документов и шум – соответствие документов, отвечающих запросу и документов, нерелевантных ему.

Существует два основных типа поиска. *Атрибутивный* – каждому документу присваивается набор определенных атрибутов, присваиваемых документу во время его размещения в архиве. В дальнейшем документ ищется на совпадение значений этих атрибутов полям запроса. К атрибутам документа можно отнести имя документа, время создания, автор, машинистка, имя подразделения, тип документа (факс, письмо, контракт, спецификация и т. д.). Совокупность атрибутов документа называется карточкой документа. Эти поля могут заполняться произвольно или из predetermined справочников.

Второй тип поиска – *полнотекстовый*. В этом случае автоматически обрабатывается все содержание документа, и затем по любому слову, входящему в данный документ, можно найти сам документ. Соответственно существует зависимость между типами поиска и подходами к поиску.

Для поиска известного документа, более применим атрибутивный поиск, для исследовательского поиска – полнотекстовый. Существует комбинация полнотекстового и атрибутивного поиска, когда атрибуты документа обрабатываются так же, как все содержание документа. Полнотекстовый поиск зависит от формата документа и языка, на котором он создан. Электронный документ любого формата необходимо предварительно преобразовывать в стандартный текст для обработки системой полнотекстового поиска. Зависимость от языка выражается в следующих факторах: поиск документа будет более полным, если в результате запроса будут найдены не только документы, которые точно соответствуют слову в запросе, но и те документы, в которых

присутствуют различные словоформы слова в запросе. Данная технология носит название нормализации. Причем эффективность метода зависит от применяемого алгоритма нормализации. Для русского языка наиболее эффективен словарный метод – слово нормализуется на основе словарей, в которых содержатся словоформы. Кроме словарного метода может применяться эвристический метод нормализации, когда слово может быть приведено к нормальному виду путем выполнения определенных правил, описывающих алгоритмику нормализации. Если для английского языка свод правил нормализации составляет 300 страниц машинописного текста, то для русского языка этот свод на несколько порядков больше. Аналогично нормализации было бы логично выполнять поиск не только по конкретному слову, но и по его синонимам.

4. Поддержка защиты документов от несанкционированного доступа и аудита работы.

Архивная система должна иметь защиту на уровне документа – каждый документ должен иметь ассоциированный с ним список пользователей, имеющих право совершать с ним определенные операции. Для статических архивов этот список операций может представлять следующий набор: просмотр и печать документа, право изменять карточку документа.

5. Поддержка просмотра документов без загрузки приложений, которые порождают документ.

Архивная система должна поддерживать специальные программы просмотра, позволяющие получить доступ к документам разнообразных форматов без загрузки ресурсоемких приложений.

6. Поддержка аннотирования документа.

Иногда для обеспечения коллективной работы с документом необходимо заносить комментарии на документ, не изменяя его основного содержания. Комментарии (знаки, текст, цветные пометки) хранятся в слоях, которые могут быть привязаны к имени автора, создавшего эти комментарии.

В качестве характерного примера реализации статического архива можно привести системы типа WaterMark, PaperWise или ImageWise.

2.6.4.2. Задачи динамических архивов

1. Поддержка коллективной работы с документом.

Такая коллективная работа выражается в поддержке целостности документов, для этого должны быть реализованы библиотечные функции выписки и возврата документов на/с редактирования, что предотвращает одновременное редактирование одного документа несколькими пользователями, устраняя тем самым возможные конфликты.

Для того чтобы позволить в рамках одного документа работать одновременно нескольким пользователям, вводится понятие версии и

подверсии документа – один документ может содержать несколько версий, а каждая версия – несколько подверсий. Кроме того, в рамках одного документа и одной версии (подверсии) документа может существовать несколько представлений документов (в разных форматах).

2. Поддержка составных документов.

Каждый документ может представлять собой совокупность других документов. Такой документ носит название составного или контейнера, а в делопроизводстве такой документ носит название «дело». По своим характеристикам он аналогичен простому документу. Документы могут быть объединены в составной документ с помощью нескольких типов связей. Эти связи определяют, какие версии объединяются в контейнер, например, последняя по дате, последняя отредактированная, старшая версия и т. д. Эти связи определяют, как будет осуществляться сборка документа в составной документ. Для составных документов должно существовать приложение, которое будет осуществлять окончательную сборку такого документа.

3. Поддержка распространения опубликованных документов.

После того как документ публикуется его иногда необходимо распространять. Это происходит двумя основными путями, либо через систему электронной почты, путем рассылки, либо через Internet, путем публикации на Web-сервере.

4. Поддержка расширенного спектра прав доступа к документу.

Расширение прав доступа к документу состоит из прав на редактирование документа, прав на публикацию, прав на снятие публикации, прав на создания новой версии документа.

Контрольные вопросы

1. Перечислите стандарты корпоративных ИС. В чем заключаются их особенности?

2. Назовите главную задачу стандарта MRP. В чем заключается преимущества использования стандарта MRP. Перечислите элементы приложения стандарта MRP.

3. Чем стандарт MRPII отличается от MRP. Перечислите основные процессы планирования стандарта MRPII.

4. В чем заключается особенность стандарта ERP. Назовите наиболее известные ERP-системы.

5. Для каких целей создан стандарт MPC. Назовите критерии оценки MPC-систем.

6. Перечислите особенности стандарта CSRP.

7. Какие типы носителей существует в системах автоматизации документооборота? Назовите причины использования разных носителей.

8. Назовите требования, предъявляемые к системе документооборота. Охарактеризуйте их.

9. Охарактеризуйте задачи, решаемые системами документооборота.

3. Принципы иерархического проектирования корпоративных сетей

3.1. Структура корпоративных информационных систем и их элементов

Структуру корпоративных ИС можно разделить на две относительно независимые составляющие.

Первая представляет собой инфраструктуру, используемую для манипулирования информацией, и является набором программных и аппаратных средств, организующих информационное взаимодействие внутри предприятия. Достаточно часто для обозначения этой составляющей корпоративной ИС используется термин корпоративная сеть (КС). Она представляет собой основу для интеграции функциональных подсистем и определяет свойства ИС, важные для ее успешной эксплуатации.

Вторая составляющая корпоративной ИС представляет собой набор взаимосвязанных функциональных подсистем, обеспечивающих решение задач организации и достижение ее целей. Эта составляющая строится целиком на базе первой составляющей и направлена на решение конкретных задач предприятия.

Несмотря на то, что эти составляющие функционируют и развиваются независимо друг от друга, построение эффективной корпоративной ИС невозможно без учета взаимного влияния этих двух составляющих. Обладая во многом противоречивыми свойствами (рис. 3.1), эти составляющие взаимодействуют между собой, образуя единую ИС, способную с необходимой степенью эффективности справляться с возложенными на нее задачами.



Рис. 3.1. Составляющие корпоративной ИС

Корпоративная сеть является структурой, выполняющей роль несущей конструкции для ИС современной организации. Ее построение обычно связано с большими капиталовложениями, исключающими возможность полной или частичной переделки созданного, поэтому корпоративная сеть создается, как правило, на годы вперед. Современные компании, специализирующиеся на построении корпоративных сетей, как правило, предлагают такие промышленные решения для построения

инфраструктуры предприятия, которые гарантированно обеспечат непрерывное развитие и совершенствование системно-технической базы ИС с перспективой на многие годы вперед.

Прикладная составляющая корпоративной ИС характеризуется значительно большей изменчивостью и гибкостью. Это связано с тем, что ее структура сильно зависит от организационно-управленческой структуры организации, ее функциональности, распределения функций, принятых в организации финансовых технологий и схем, существующей технологии документооборота и множества других факторов.

В теории и практике построения корпоративных ИС существует три подхода к формированию структур и механизмов взаимодействия рассмотренных выше двух компонентов ИС:

- Подход «сверху-вниз», когда архитектура ИС корпорации строится, исходя из требуемой прикладной функциональности, и состав корпоративной сети рассчитывается, исходя из нужд функциональности. Теоретически этот подход очень удобен, однако на практике функциональная составляющая ИС вынуждена достаточно часто меняться, отражая изменения в организационной структуре предприятия. Это приводит к неоправданным затратам на постоянную модификацию аппаратных и программных средств корпоративной сети.

- Подход «снизу-вверх», когда предприятие строит компьютерную сеть исходя из самых грубых оценок потребностей функциональной составляющей ИС. Данный подход чреват тем, что в случае неудачного выбора архитектуры компьютерной сети многие из ее возможностей будут использоваться не по назначению, либо не использоваться вообще.

- Комбинированный подход. Согласно этому подходу, компьютерная инфраструктура и системная функциональность строятся так, чтобы в максимальной степени обеспечить изменчивость на уровне прикладной функциональности. Параллельно проводится анализ и структуризация бизнес-процессов, сопровождающиеся внедрением соответствующих программных решений, приносящих в корпоративную ИС прикладную функциональность. Такой подход является наиболее оптимальным в условиях современного бизнеса.

При построении корпоративной сети, способной в достаточной мере поддержать функциональность прикладной части ИС, следует руководствоваться следующими основными принципами:

1. Корпоративная сеть должна рассматриваться как одна из стратегических систем жизнеобеспечения организации. Расчет эффективности внедрения технико-экономических решений, исходя из этого принципа, бессмыслен, так как подобное внедрение не имеет прямого денежно-экономического эффекта. Однако создание корпоративной сети необходимо для последующего внедрения

информационных технологий, приносящих положительный экономический эффект.

2. Основой корпоративной сети должна являться эффективная система централизованных коммуникаций. Для современных корпораций, имеющих распределенную управленческую структуру, ключевым фактором эффективного контроля, координации и стратегического управления является эффективная система централизованных коммуникаций.

3. Одним из основным принципов, положенных в основу создания корпоративных сетей, является максимальное использование типовых решений, стандартных унифицированных компонентов. Конкретизируя этот принцип применительно к прикладному программному обеспечению (ПО), можно выделить ряд универсальных сервисов, которые целесообразно сделать базовыми компонентами приложений. Такими сервисами являются сервис СУБД, файловый сервис, информационный сервис (Web-сервис), электронная почта, сетевая печать и другие. Основным средством для построения прикладных и системных сервисов является ПО промежуточного слоя. В трактовку ПО промежуточного слоя включено все, что находится между платформой (компьютер плюс операционная система) и приложениями. Примером ПО промежуточного слоя может служить СУБД.

3.2. Топология иерархических сетей

В соответствии с концепцией иерархических сетей процесс взаимодействия между компьютерами разбивается на несколько функциональных уровней, выполняющих определенный круг задач. Уровни иерархической модели должны как можно более точно соответствовать поставленным перед ними целям. В большинстве случаев иерархическая модель сети подразумевает определение трех уровней: ядро, уровень распределения и уровень доступа (рис. 3.2), выполняющих собственные функциональные задачи.

- *Ядро (core)* сети отвечает за высокоскоростную передачу сетевого трафика; первичное предназначение устройства, входящего в ядро сети, заключается в коммутации пакетов.

- На *уровне распределения (distribution layer)* происходит суммирование маршрутов и агрегация трафика.

- *Уровень доступа (access layer)* отвечает за формирование сетевого трафика, выполняет контроль точек входа в сеть и предоставляет другие службы пограничных устройств.

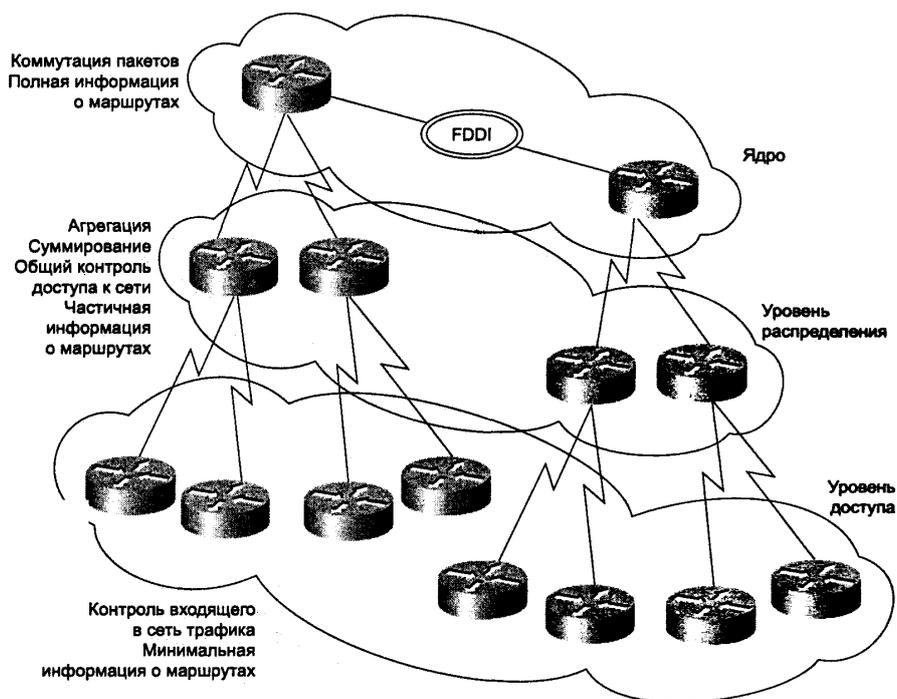


Рис. 3.2. Структура иерархической сети

3.2.1. Ядро сети

Целью внедрения ядра в иерархию сети является повышение скорости коммутации пакетов. Достигается увеличение скорости коммутации с помощью двух стратегий:

1. *Полная достижимость*. Осуществляется путем реализации запрета на использование стандартных маршрутов для достижения внутренних пунктов назначения и уменьшение частично оптимизированной маршрутизации.

2. *Запрет на реализацию сетевых правил*. Осуществляется путем реализации запрета контроля за доступом к сети, запрета на реализацию сетевых правил, а также уменьшение загрузки процессора и памяти.

В большинстве малых сетей используется *вырожденный (collapsed)* тип ядра. Вырожденное ядро состоит из одного маршрутизатора, выполняющего роль ядра сети и соединенного со всеми остальными маршрутизаторами уровня распределения.

Поскольку вырожденное ядро с одним маршрутизатором не соответствует потребностям крупной сети, в большинстве корпоративных сетей используется группа маршрутизаторов, соединенных между собой высокоскоростной локальной сетью (Local Area Network – LAN), или объединение высокоскоростных каналов глобальных сетей (Wide Area Network – WAN), которые формируют ядро сети. Использование базовой сети вместо единственного маршрутизатора в качестве ядра позволяет предусмотреть избыточность еще на этапе проектирования и сохраняет возможность масштабирования сети путем добавления в ядро новых маршрутизаторов и каналов передачи информации.

3.2.2. Уровень распределения

Целью внедрения уровня распределения в иерархию сети является локализация участка сети, на который влияет изменение топологии; управление размером таблицы маршрутизации, а также агрегация сетевого трафика.

Для достижения этих целей используются две основные стратегии, реализуемые на уровне распределения:

1. *Суммирование маршрутов.* Осуществляется путем реализации локализации участка сети, на который влияет изменение топологии, сокрытие детальной информации о маршрутах от устройств ядра сети и устройств уровня доступа;

2. *Минимизация числа каналов, соединяющих уровень распределения с ядром сети.* Осуществляется за счет уменьшения сложности принятия решения о коммутации пакета и создание предпосылок для проведения суммирования маршрутов и агрегации трафика.

Вторая стратегия реализуется за счет агрегации сетевого трафика. Агрегация достигается за счет объединения трафика, поступающего по большому числу низкоскоростных каналов передачи информации, которые связывают уровень распределения с устройствами уровня доступа, в несколько широкополосных каналов, которые, в свою очередь, связывают уровень распределения с ядром сети. Подобная стратегия порождает в сети эффективные точки суммирования и уменьшает количество маршрутов. Эта информация используется маршрутизаторами ядра при принятии решения о коммутации пакетов.

3.2.3. Уровень доступа

Целью внедрения уровня доступа в иерархию сети является формирование сетевого трафика, а также контроль доступа к сети. Устройства уровня доступа соединяют высокоскоростные каналы локальных сетей с каналами глобальной сети, несущими трафик на уровень распределения. Задача формирования сетевого трафика решается главным образом на этапе определения пропускной способности канала передачи информации. Она имеет отношение к расположению пары «сервер – служба». Контроль доступа к сети осуществляется на основе фильтрации пакетов. Трафик, не предназначенный для узла за пределами локальной сети, не должен передаваться устройством уровня доступа.

3.3. Адресация и суммирование

3.3.1. Суммирование в сетях IP

Стабильность сети в значительной степени зависит от числа маршрутизаторов, которые задействованы в процессе изменения маршрута данных (сходимости), вызванном какими-либо нарушениями в сети. Суммирование позволяет скрыть подробную информацию о топологии,

сужая тем самым участок сети, подверженный изменениям, и уменьшая число маршрутизаторов, участвующих в процессе сходимости.

В качестве примера рассмотрим сеть (рис. 3.3). Предположим, что канал, ведущий к сети 10.1.4.0/24 или 10.1.7.0/24, будет поврежден, в этом случае, маршрутизатору Н потребуется учесть данное изменение топологии и провести пересчет своей таблицы маршрутизации. В результате увеличивается время доставки данных. Если можно было бы «спрятать» информацию об изменении топологии от маршрутизатора Н так, чтобы ему не пришлось участвовать в процессе сходимости при повреждении каналов, ведущих к сетям 10.1.4.0/24, 10.1.5.0/24, 10.1.6.0/24 и 10.1.7.0/24, то сеть в значительной степени стала более стабильной.

IP-адрес состоит из четырех частей, каждая из которых представляет собой восемь двоичных цифр (*битов (bits)*) или один *октет (octet)*. Каждый октет, в свою очередь, представляет собой число в промежутке от 0 до 255. Таким образом, общее количество возможных IP-адресов равняется 2^{32} , или 4 294 967 296.

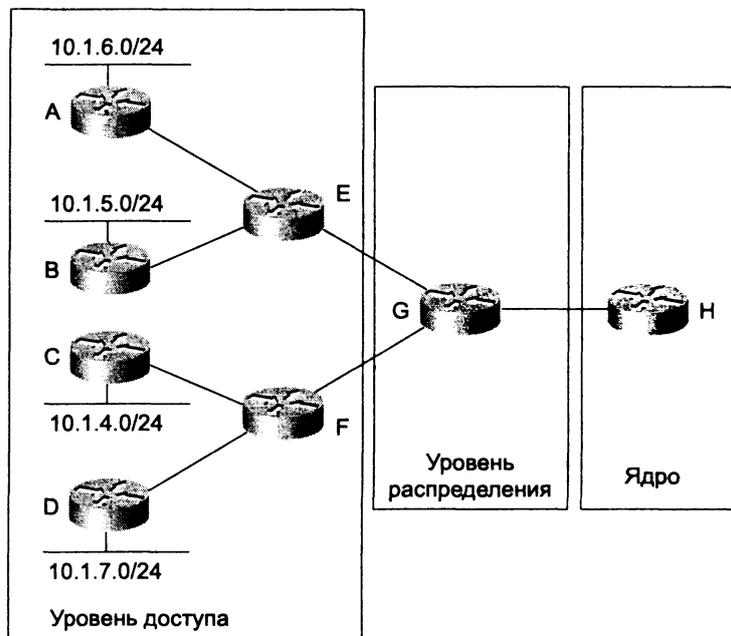


Рис. 3.3. Пример сети

Для поддержки иерархии IP-адрес делится на две части: адрес сети и адрес узла. Первый определяет сеть, к которой подключен данный узел, т. е. адрес сети обозначает канал передачи информации или физический сегмент. Адрес узла используется для однозначного представления каждого узла в сети. Разделение IP-адреса на адрес сети и адрес узла осуществляется с помощью маски (или маски подсети). Каждый бит IP-адреса, которому соответствует бит маски, установленный в 1, относится к адресу сети. Аналогично, каждый бит IP-адреса, которому соответствует бит маски, установленный в 0, относится к адресу узла.

На рис. 3.4 изображено двоичное представление IP-адреса 172.16.100.10.

уровня доступа должна быть суммирована маршрутизаторами уровня распределения, что позволит ограничить область распространения информации об изменении топологии сети одним лишь локальным участком уровня распределения. Суммирование, осуществляемое на уровне распределения по отношению к маршрутизаторам уровня доступа, позволяет существенно уменьшить объем информации, обрабатываемой данными маршрутизаторами.

Рассмотрим пример суммирования маршрутной информации (рис. 3.9). Маршрутизатор А, расположенный на уровне распределения сети, получает объявления маршрутов от следующих подсетей: 10.1.1.0/26, 10.1.1.64/26, 10.1.1.128/26 и 10.1.1.192/26.

В свою очередь, маршрутизатор А суммирует эти маршруты в одну точку назначения 10.1.1.0/24 и предоставляет информацию о едином маршруте устройствам ядра. Поскольку все четыре сети с более длинными префиксами – 10.1.1.0/26, 10.1.1.64/26, 10.1.1.128/26 и 10.1.1.192/26 – «спрятаны» от маршрутизаторов ядра, повреждение любой из этих сетей никоим образом не повлияет на ядро (другими словами, ни один из маршрутизаторов ядра не должен будет обновлять свою таблицу маршрутизации). Соккрытие подробной информации о топологии сети от устройств уровня ядра позволяет существенно сузить участок, на который влияет изменение топологии.

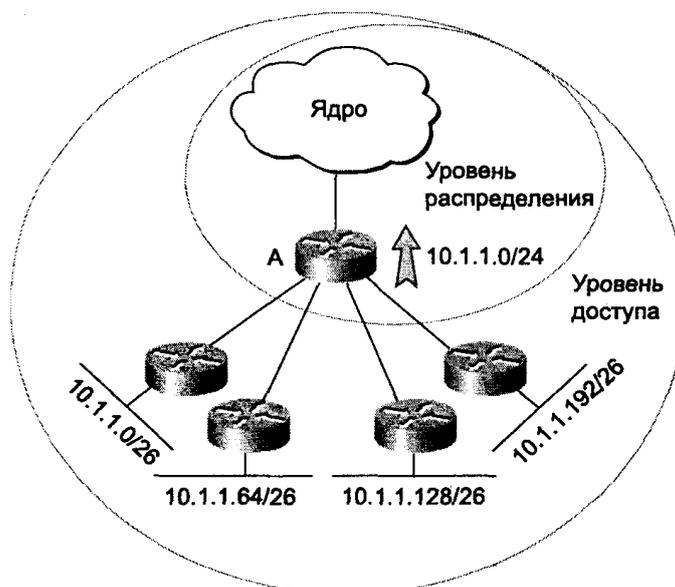


Рис. 3.9. Суммирование маршрутной информации, поступающей с уровня распределения в ядро сети

3.3.2. Стратегии адресации

Наиболее распространенными стратегиями адресации являются следующие:

1. «Первым пришел – первым обслужен». Адреса выделяются из списка зарезервированных адресов по мере их необходимости. Подобная

схема распределения сетевых адресов давно уже стала «многовековой» традицией в проектировании сетей. Фактически принцип выделения адресов «первым пришел – первым обслужен» до сих пор является наиболее распространенным. Недостаток этой стратегии начинает проявляться только при значительном увеличении масштабов сети. Понятно, что стратегия распределения адресов по принципу «первым пришел – первым обслужен» является далеко не лучшим способом выделения сетевых адресов из-за хаотичности адресации.

Для примера рассмотрим сеть, схематически изображенную на рис. 3.10. В данном случае администратор сети выделил блоки адресов по мере поступления заявок от подразделений организации.

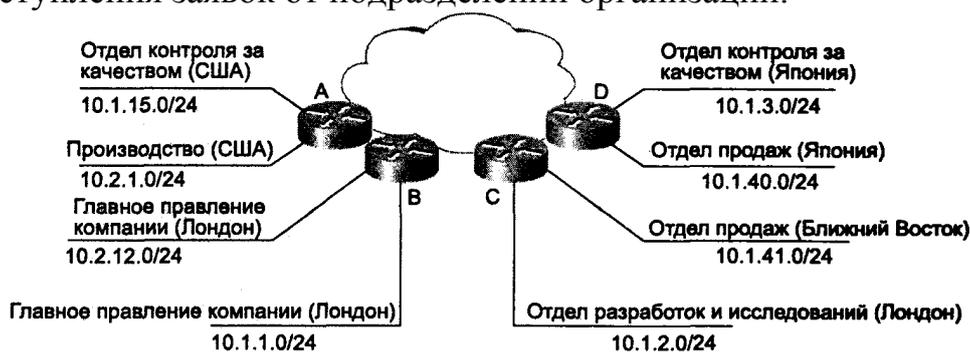


Рис. 3.10. Стратегия распределения адресов по принципу «первым пришел — первым обслужен»

- Маршрутизатор А соединен с двумя сетями: 10.1.15.0/24 и 10.2.1.0/24.
- Маршрутизатор В соединен с двумя сетями: 10.2.12.0/24 и 10.1.1.0/24.
- Маршрутизатор С соединен с двумя сетями: 10.1.2.0/24 и 10.1.41.0/24.
- Маршрутизатор D соединен с двумя сетями: 10.1.40.0/24 и 10.1.3.0/24.

Не существует ни одного относительно простого способа суммирования какой-либо из представленных пар сетей в одну сеть назначения, и чем больший участок сети мы рассмотрим, тем более проблематичным будет решение этой задачи. Если сеть с подобной схемой адресации разрастется до крупных размеров, то она определенно будет иметь проблемы, связанные со стабильностью функционирования.

2. Структурная схема адресации. Доступное адресное пространство между различными подразделениями организации распределяется так, чтобы каждое подразделение обладало собственным пространством выделяемых адресов (рис. 3.11).

- Главное правление компании: 10.1.0.0/16.
- Отдел исследований: 10.2.0.0/16.
- Отдел контроля за качеством: 10.3.0.0/16.
- Отдел продаж: 10.4.0.0/16.

- Производство: 10.5.0.0/16.

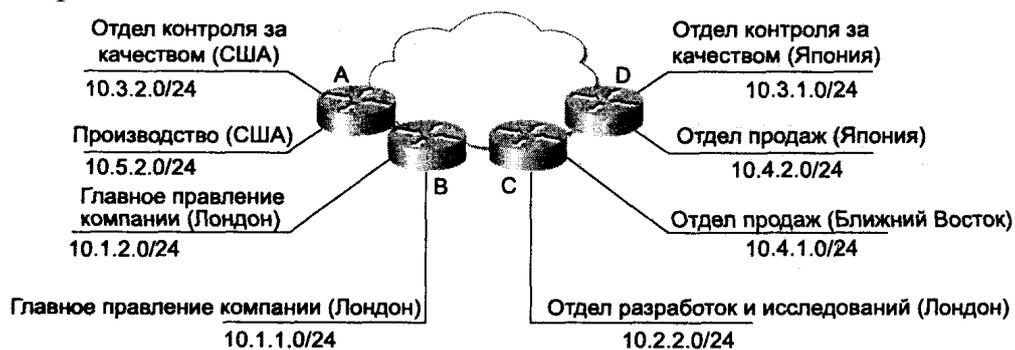


Рис. 3.11. Стратегия распределения адресов в соответствии со структурой организации

Подобная стратегия уже предоставляет некоторые шансы для проведения суммирования. Если адрес сети 10.1.3.0/24 еще не распределен, то существует возможность суммирования двух сетей главного правления компании. Однако по большому счету эта адресная схема приводит к такому же результату, что и схема распределения адресов по принципу «первым пришел – первым обслужен», т. е. проведение масштабирования сети в этом случае очень затруднено.

3. Географическая схема адресации. Доступное адресное пространство распределяется между различными региональными подразделениями организации так, чтобы каждый офис, в зависимости от географического расположения сегментов сети, обладал собственным пространством выделяемых адресов (рис. 3.12).

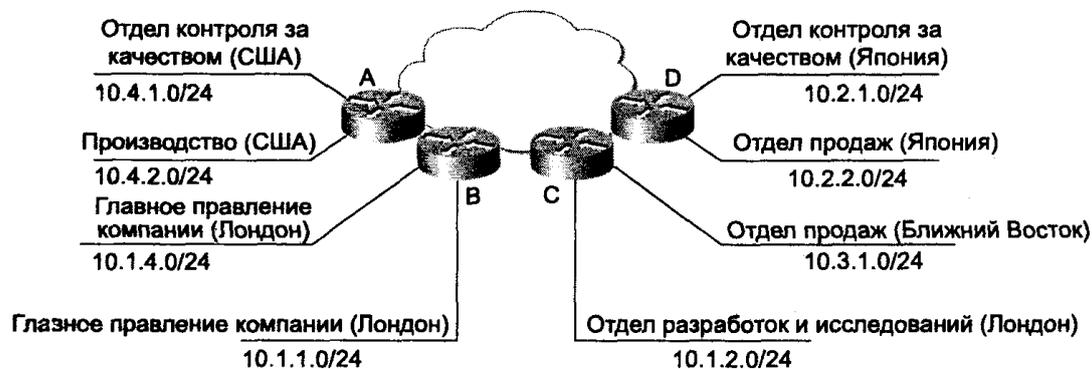


Рис. 3.12. Стратегия распределения адресов в соответствии с географическим расположением сегментов сети

В этом случае адресное пространство распределяется в соответствии с географическим расположением сегментов сети. Так, сеть Японии получила адрес 10.2.0.0/16, сеть Соединенных Штатов – адрес 10.4.0.0/16 и т. д. Проведение такой стратегии, в большинстве случаев, положительно отражается на реализации суммирования. Возможно провести суммирование двух сетей Соединенных Штатов – 10.4.1.0/24 и 10.4.2.0/24 – в сеть 10.4.0.0/16 с тем, чтобы маршрутизатор А предоставил устройствам ядра информацию об одном маршруте вместо двух. Аналогичным образом можно провести суммирование двух сетей

Японии – 10.2.1.0/24 и 10.2.2.0/24 – в сеть 10.2.0.0/16 с тем, чтобы маршрутизатор D предоставил устройствам ядра информацию об одном суммированном маршруте. Сети Лондона, однако, представляют собой некоторую проблему. Так, Лондонский отдел исследований – 10.1.2.0/24 – подключен к маршрутизатору С, в то время как остальная часть лондонских офисов подключена к маршрутизатору В. Вследствие подключения этих сетей к различным маршрутизаторам суммирование адресов 10.1.x.x не представляется возможным.

Таким образом, несмотря на то, что в некоторых случаях географический принцип распределения адресов позволяет провести суммирование маршрутов, в целом этот метод адресации не годится для проведения эффективной оптимизации маршрутной информации, передаваемой на высшие уровни иерархической сети.

4. **Топологическая схема адресации.** Наиболее эффективный способ адресации, гарантирующий возможность суммирования маршрутов, заключается в распределении адресов в зависимости от маршрутизатора, к которому подключена сеть, или, что гораздо предпочтительнее, основываясь на топологии сети (рис. 3.13).

В результате распределения адресов сети по топологическому принципу суммирование маршрутов можно легко провести на маршрутизаторах А, В, С и D, вследствие чего число маршрутов, объявляемых этими маршрутизаторами верхним уровням сети, будет сведено к минимуму. Стратегия распределения адресов, основанная на топологии, является наиболее оптимальным способом обеспечения стабильности функционирования сети.

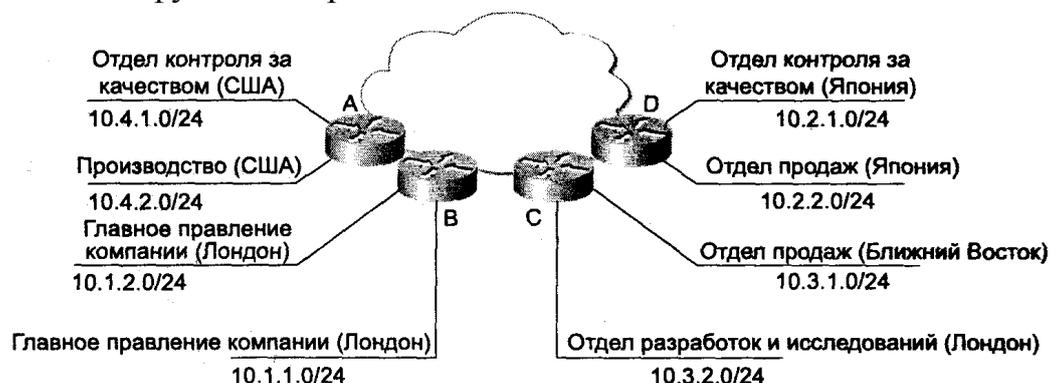


Рис. 3.13. Стратегия распределения адресов, основанная на топологии сети

Единственный недостаток топологического принципа распределения адресов заключается в несколько затрудненном способе определения какого-либо контекста без предварительного знания структуры или базы данных, например подразделения компании, к которому относится определенная сеть. Такой недостаток можно минимизировать, сочетая топологический принцип с другими принципами распределения адресов, например, с структурным принципом.

Поскольку IP-адрес состоит из четырех октетов, существует возможность использования двух левых октетов для географической адресации сети и третьего октета для адресации по подразделениями компании (или какой-нибудь другой комбинации). Предположим, что различным подразделениям компании были назначены следующие адреса:

- Администрация: 0-31.
- Отдел исследований: 32-63.
- Отдел продаж: 64-95.
- Производство: 96-127.

Тогда нумерация точек подключения к маршрутизаторам может иметь следующую нумерацию:

- Маршрутизатор A: 4.
- Маршрутизатор B: 1.
- Маршрутизатор C: 3.
- Маршрутизатор D: 2.

С учетом вышеизложенного выделенные адреса могут иметь следующий вид:

- Сеть администрации, подключенная к маршрутизатору A: с 10.4.0.0/24 по 10.4.31.0/24.
- Сеть отдела исследований, подключенная к маршрутизатору A: с 10.4.32.0/24 по 10.4.63.0/24.
- Сеть отдела продаж, подключенная к маршрутизатору B: с 10.1.64.0/24 по 10.1.95.0/24.
- Производственная сеть, подключенная к маршрутизатору C: с 10.3.96.0/24 по 10.3.127.0/24.

3.4. Избыточность в иерархических сетях

Одиночная точка отказа – это любое устройство, интерфейс устройства или канал передачи информации, который, будучи поврежденным, приводит к изоляции пользователей от необходимых им в повседневной работе служб. Для сетей, спроектированных в соответствии со строгой иерархической моделью, характерно наличие достаточно большого числа одиночных точек отказа, что связано, в первую очередь, с их ориентацией на проведение суммирования и четкой разграниченностью между различными уровнями сети. Например, в иерархической сети, схематически изображенной на рис. 3.14, каждое устройство и каждый канал передачи информации представляет собой одиночную точку отказа.

Избыточность предполагает наличие альтернативных способов передачи информации, позволяющих снизить вероятность выхода из строя той или иной службы сети вследствие повреждения одиночной точки отказа. В то же время, каждый избыточный канал передачи информации и каждая избыточная точка подключения ослабляют иерархию и приводят к снижению стабильности функционирования сети.

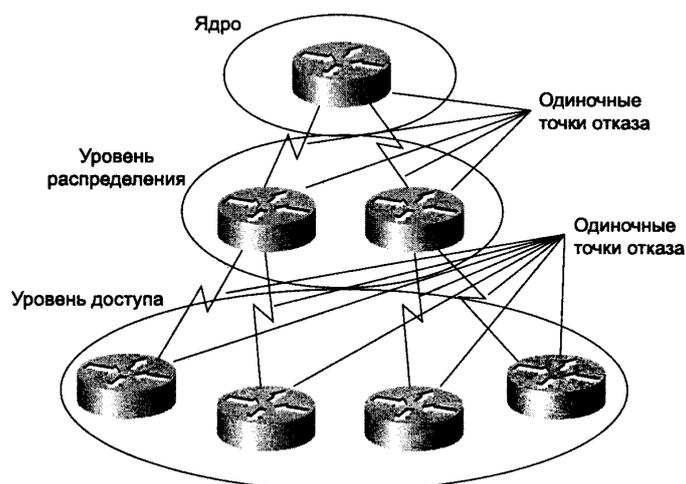


Рис. 3.14. Каждое устройство и каждый канал передачи информации иерархической сети представляет собой одиночную точку отказа

Поэтому при реализации избыточности в иерархической сети необходимо придерживаться следующих стратегий.

- Избыточные каналы передачи информации должны использоваться только в случае повреждения основных каналов.
- Трафик сети не должен проходить через устройства или каналы передачи информации, не предназначенные для обработки транзитного трафика. Предотвращение использования резервных линий связи для передачи трафика в условиях нормального функционирования сети подразумевает их «сокрытие» до момента повреждения основных каналов.

3.4.1. Избыточность, реализуемая на уровне ядра сети

В общем случае задача реализации избыточности на уровне ядра сети считается несколько упрощенной за счет того, что все устройства ядра должны обладать полной информацией о маршрутах (принцип полной достижимости). Единственным исключением из этого правила является стандартный маршрут, используемый для достижения внешних маршрутных доменов (таких, как Internet или сеть корпоративного бизнес-партнера).

3.4.1.1. Структура ядра с резервными связями

Избыточность структур ядра с резервными связями может быть достигнута за счет подключения каждого маршрутизатора к двум высокоскоростным локальным сетям (LAN), таким, как Ethernet или волоконно-оптическое кольцо (рис. 3.15).

Если базовые маршрутизаторы расположены в различных зданиях, то это значительно уменьшает возможности по реализации избыточности на уровне ядра. Для того чтобы добиться приемлемых результатов при реализации избыточности в ядре крупномасштабных сетей, следует придерживаться трех конкурирующих между собой принципов:

- уменьшение счетчика пройденных маршрутизаторов;

- уменьшение числа доступных каналов передачи информации;
- увеличение числа отказов, которые может выдержать ядро.



Рис. 3.15. Реализация избыточности на уровне ядра сети за счет подключения базовых маршрутизаторов к двум высокоскоростным локальным сетям

3.4.1.2. Структура ядра типа «кольцо»

Структура ядра, относящиеся к типу «кольцо» (рис. 3.16), образована путем использования нескольких двухточечных каналов передачи информации, объединяющих между собой маршрутизаторы ядра.

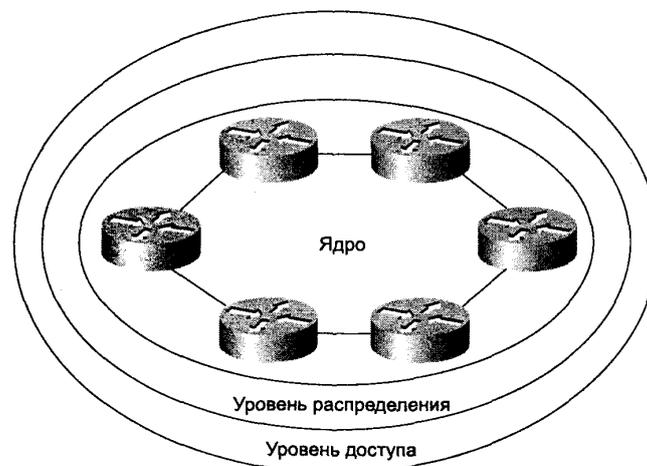


Рис. 3.16. Структура ядра типа «кольцо»

Основными характеристиками структуры ядра типа «кольцо» являются:

- от каждого устройства ядра до любой точки назначения сети существует два различных маршрута;
- при условии неповрежденного ядра пакет проходит максимум четыре базовых маршрутизатора;
- в случае повреждения одного канала передачи информации максимальное число базовых маршрутизаторов, которые проходит пакет, увеличивается до шести.

- повреждение двух произвольных каналов передачи информации ведет к изоляции по крайней мере одного участка сети.

В то время как число проходящих через сеть маршрутов остается на достаточно низком уровне в условиях нормального режима функционирования ядра, повреждение одного канала передачи информации приводит к абсолютно неприемлемому увеличению числа базовых маршрутизаторов, через которые может пройти пакет в худшем случае.

3.4.1.3. Структура ядра с полным объединением

Структура ядра с полным объединением (рис. 3.17), подразумевающая соединение с помощью резервного канала передачи информации каждой пары базовых маршрутизаторов, позволяет достичь максимально возможного уровня избыточности. Ниже перечислены характеристики этой структуры.

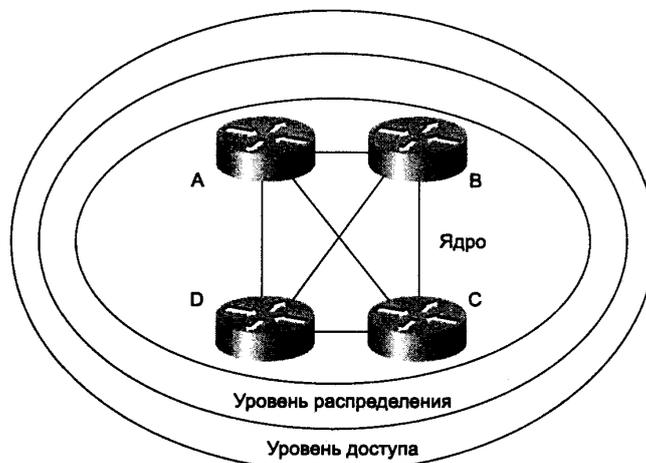


Рис. 3.17. Структура ядра с полным объединением

- Огромное число различных маршрутов к любой точке назначения сети.
- При условии неповрежденного ядра пакет проходит через два базовых маршрутизатора.
 - В самом худшем случае пакет проходит через четыре базовых маршрутизатора.
 - Чрезвычайно высокая степень обеспечения избыточности. Поскольку каждый маршрутизатор соединен каналами связи со всеми другими маршрутизаторами, ядро такой сети должно потерять по крайней мере три канала передачи информации, прежде чем какая-либо точка назначения станет недоступной.

Сильные стороны структуры ядра с полным объединением проявляются в приемлемом значении счетчика пройденных маршрутизаторов и максимально возможном уровне обеспечения избыточности. К сожалению, подобная структура может привести к возникновению «излишней» избыточности в крупных сетях, в результате

чего базовые маршрутизаторы будут вынуждены обрабатывать информацию об очень большом количестве маршрутов, что скажется негативным образом на времени, требуемом для завершения процесса сходимости.

Согласно схеме ядра, представленной на рис. 3.17, маршрутизатор А располагает пятью маршрутами, связывающими его с маршрутизатором С.

- Маршрутизатор А → маршрутизатор С.
- Маршрутизатор А → маршрутизатор В → маршрутизатор С.
- Маршрутизатор А → маршрутизатор D → маршрутизатор С.
- Маршрутизатор А → маршрутизатор В → маршрутизатор D → маршрутизатор С.
- Маршрутизатор А → маршрутизатор D → маршрутизатор В → маршрутизатор С.

Добавление еще одного маршрутизатора к ядру данной сети приведет к возрастанию количества маршрутов между маршрутизатором А и маршрутизатором С до девяти; добавление шестого маршрутизатора увеличит это значение до четырнадцати. В целом же сеть из n узлов со структурой с полным объединением должна иметь $(n(n-1))/2$ каналов передачи информации.

3.4.1.4. Структура ядра с частичным объединением

Структура ядра с частичным объединением позволяет достичь вполне приемлемого компромисса между значением счетчика пройденных маршрутизаторов, уровнем избыточности и числом различных маршрутов. Например, в сети, схематически представленной на рис. 3.18, существует четыре различных маршрута между двумя произвольными точками назначения.

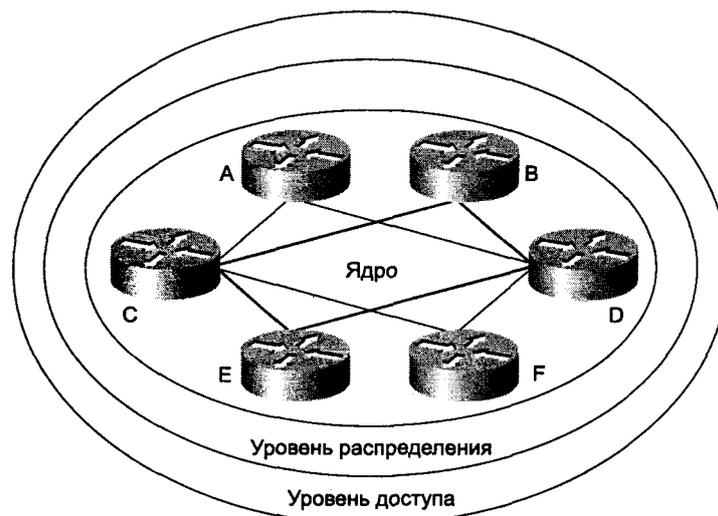


Рис. 3.18. Структура ядра с частичным объединением

Для определенности рассмотрим все маршруты, которые связывают маршрутизаторы А и F.

- Маршрутизатор А → маршрутизатор D → маршрутизатор F.

- Маршрутизатор А → маршрутизатор С → маршрутизатор F.
- Маршрутизатор А → маршрутизатор D → маршрутизатор E → маршрутизатор С → маршрутизатор F.
- Маршрутизатор А → маршрутизатор С → маршрутизатор В → маршрутизатор D → маршрутизатор F.

Ввиду существенного различия в длине четырех доступных маршрутов для передачи трафика в обычных условиях целесообразно использовать только два наиболее коротких из них. Для того чтобы обойти подобную сеть в условиях ее нормального функционирования, пакету понадобится пройти не более трех базовых маршрутизаторов. При повреждении произвольного канала передачи информации максимальное число базовых маршрутизаторов, которое потребуется пройти пакету для полного обхода сети, увеличивается до четырех.

3.4.2. Избыточность, реализуемая на уровне распределения сети

Существует два наиболее распространенных метода реализации избыточности на уровне распределения сети: двойное подключение устройств и использование резервных каналов передачи информации для соединения маршрутизаторов уровня распределения. Главный акцент при разработке структуры, обеспечивающей избыточность на уровне распределения сети, должен быть сделан на непредусмотренных схемах движения трафика.

3.4.2.1. Двойное подключение маршрутизаторов уровня распределения к маршрутизаторам ядра

При таком методе маршрутизатор на уровне распределения имеет два подключения к ядру с использованием двух различных базовых маршрутизаторов (рис. 3.19).

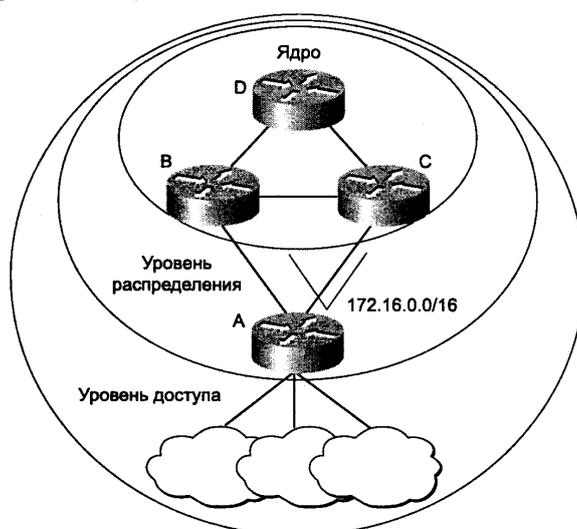


Рис. 3.19. Двойное подключение маршрутизатора уровня распределения к ядру сети

Подобное решение обеспечивает хороший уровень избыточности. Выход из строя одного базового маршрутизатора или повреждение одного канала передачи данных не приводит к изоляции точек назначения, находящихся за маршрутизатором А. Однако оно способно стать причиной возникновения целого ряда проблем.

Если бы маршрутизатор А был подключен только к одному базовому маршрутизатору, то маршрутизатор D имел бы два маршрута к сети 172.16.0.0/16:

- Маршрутизатор D → маршрутизатор В → маршрутизатор А.
- Маршрутизатор D → маршрутизатор С → маршрутизатор В → маршрутизатор А.

В случае двойного подключения маршрутизатора А к ядру маршрутизатор D будет располагать четырьмя различными маршрутами к сети, находящейся за маршрутизатором А:

- Маршрутизатор D → маршрутизатор С → маршрутизатор А.
- Маршрутизатор D → маршрутизатор В → маршрутизатор А.
- Маршрутизатор D → маршрутизатор С → маршрутизатор В → маршрутизатор А.
- Маршрутизатор D → маршрутизатор В → маршрутизатор С → маршрутизатор А.

Двойное подключение маршрутизатора А к ядру сети в два раза увеличивает число маршрутов, доступных изнутри ядра к сети 172.16.0.0/16. Следует отметить, что двукратное увеличение возможных маршрутов для каждого маршрутизатора уровня доступа замедляет процесс сходимости во всей сети.

Использование двойного подключения имеет еще один недостаток: в случае повреждения канала передачи информации между маршрутизатором В и маршрутизатором С маршрутизатор А может оказаться «втянутым» в ядро, передавая транзитный трафик между маршрутизатором В и маршрутизатором С. Несмотря на то что иногда подобная роль маршрутизатора А может быть заранее предусмотрена или запланирована, в общем случае это, конечно же, не так. Наиболее легкий способ решения подобной проблемы заключается в специальной настройке маршрутизатора D, запрещающей ему объявлять маршрутизатору В те маршруты, которые были получены от маршрутизатора С, и наоборот.

3.4.2.2. Использование резервных каналов передачи информации для соединения маршрутизаторов уровня распределения

Прокладывание резервных каналов передачи информации между маршрутизаторами уровня распределения (рис. 3.20) имеет ряд недостатков:

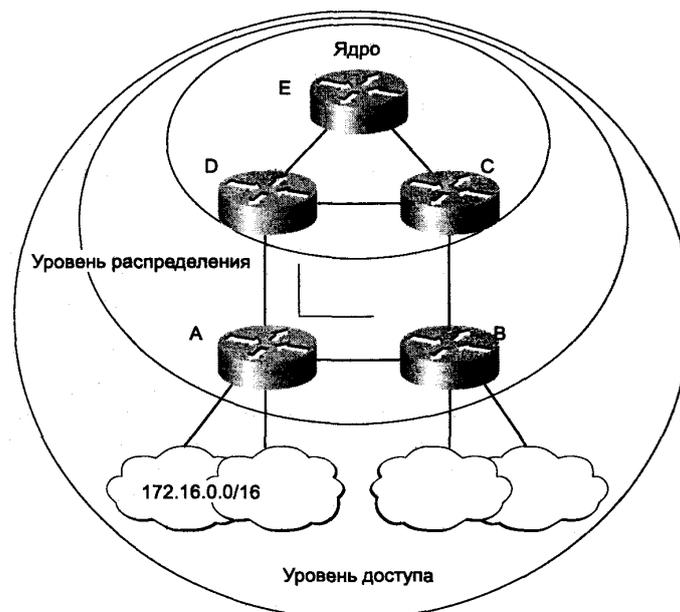


Рис. 3.20. Использование резервных каналов передачи информации для соединения маршрутизаторов уровня распределения

- Увеличение вдвое размера базовой таблицы маршрутизации. Добавление резервного канала передачи информации, соединяющего маршрутизатор А и маршрутизатор В (рис. 5.19), приведет к увеличению вдвое размера базовой таблицы маршрутизации, так как в этом случае маршрутизатор D будет обладать информацией о двух различных маршрутах к сети 172.16.0.0/16 – проходящем через маршрутизатор А и проходящем через маршрутизатор С.

- Возможность использования резервной линии связи для передачи транзитного трафика между устройствами ядра. Если канал связи, соединяющий маршрутизатор D и маршрутизатор С (рис. 3.20), окажется поврежденным, то существует вероятность того, что маршрутизатор D начнет передавать трафик, предназначенный для каких-либо точек назначения за пределами маршрутизатора С, например, маршрутизатору А вместо маршрутизатора Е.

- Предпочтение резервного канала передачи информации маршруту, проходящему через ядро сети. Маршрутизаторы уровня распределения могут предпочесть резервный канал передачи информации маршруту, проходящему через ядро сети. Например, в сети, изображенной на рис. 3.20, маршрутизатор В может предпочесть резервный канал связи маршруту, проходящему через ядро, при передаче трафика, предназначенного для сети 172.16.0.0/16.

3.4.3. Избыточность, реализуемая на уровне доступа сети

Задачи реализации избыточности на уровне доступа и на уровне распределения сети имеют много общего в аспекте как основных проблем, так и стратегий, использующихся для их эффективного решения. Несмотря

на то, что наиболее распространенным способом обеспечения избыточности каналов связи с удаленными сегментами сети является двойное подключение устройств уровня доступа к маршрутизаторам уровня распределения, избыточность данного участка сети также может быть реализована за счет взаимного соединения маршрутизаторов уровня доступа.

Маршрутизаторы G и H (рис. 3.21) являются устройствами уровня доступа, которые подключены к различным ветвям уровня распределения с помощью двух резервных каналов передачи информации.

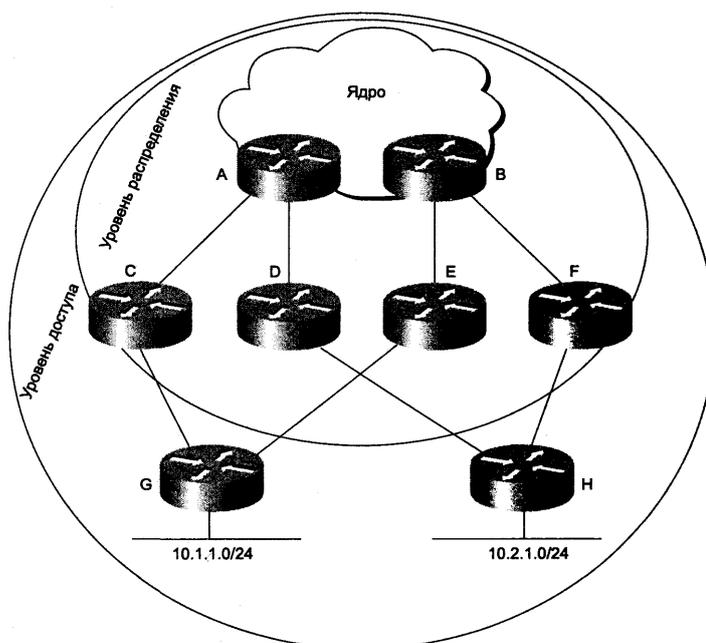


Рис. 3.21. Избыточность, реализуемая на уровне доступа сети: двойное подключение устройств к различным ветвям уровня распределения

При условии полноценного функционирования резервных каналов передачи информации число маршрутов между сетями 10.2.1.0/24 и 10.1.1.0/24 оказывается чрезмерно большим:

- Маршрутизатор H → маршрутизатор F → маршрутизатор B → маршрутизатор A → маршрутизатор C → маршрутизатор G.
- Маршрутизатор H → маршрутизатор F → маршрутизатор B → маршрутизатор E → маршрутизатор G.
- Маршрутизатор H → маршрутизатор D → маршрутизатор A → маршрутизатор C → маршрутизатор G.

Альтернативный способ обеспечения избыточности на уровне доступа сети заключается во взаимном соединении маршрутизаторов уровня доступа с помощью резервных каналов связи. Как показано на рис. 3.22, это позволяет сэкономить на одном канале передачи информации и сократить число маршрутов между сетями 10.1.1.0/24 и 10.2.1.0/24 до двух.

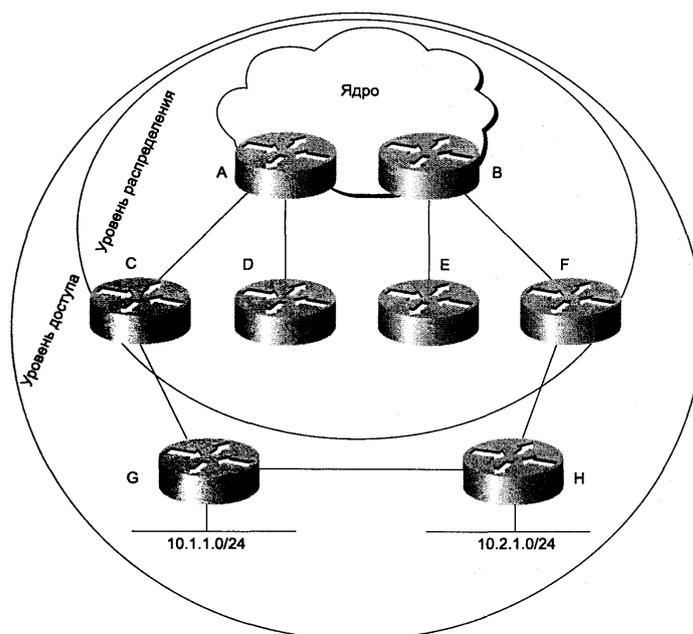


Рис. 3.22. Избыточность, реализуемая на уровне доступа сети: взаимное соединение устройств

Реализуя избыточность на уровне доступа сети за счет взаимного соединения устройств, следует обратить внимание на необходимость обеспечения достаточной пропускной способности каналов, передающих трафик от удаленных сегментов сети по направлению к ядру.

Контрольные вопросы

1. Охарактеризуйте составляющие корпоративных ИС.
2. Назовите три подхода к формированию структур и механизмов взаимодействия корпоративных ИС. Охарактеризуйте их.
3. Почему иерархические сети состоят из нескольких уровней?
4. Какова первичная цель ядра сети? Какие стратегии используются для достижения этой цели?
5. Назовите первичные цели уровня распределения сети.
6. Какие стратегии используются на уровне распределения сети для достижения первичных целей этого уровня?
7. Назовите первичные цели уровня доступа сети.
8. В чем заключается смысл уменьшения длины префикса?
9. Перечислите стратегии адресации. Охарактеризуйте их.
10. Назовите стратегии реализации избыточности в иерархических сетях.
11. Охарактеризуйте избыточность, реализуемую на уровне распределения сети.
12. Охарактеризуйте избыточность, реализуемую на уровне доступа сети.

4. Безопасность корпоративных сетей

Безопасность информационного взаимодействия корпоративных локальных сетей и отдельных компьютеров через открытые сети, например, через глобальную сеть Internet, требует качественного решения двух базовых задач:

- защиты подключенных к публичным каналам связи локальных сетей и отдельных компьютеров от несанкционированных действий со стороны внешней среды;
- защиты информации в процессе передачи по открытым каналам связи.

Решение первой задачи основано на использовании межсетевых экранов (МЭ), поддерживающих безопасность информационного взаимодействия путем фильтрации двустороннего потока сообщений, а также выполнения функций посредничества при обмене информацией.

Защита информации в процессе передачи по открытым каналам связи основана на выполнении следующих функций:

- аутентификации (установление подлинности) взаимодействующих сторон;
- шифровании передаваемых данных;
- подтверждении подлинности и целостности доставленной информации;
- защите от повтора, задержки и удаления сообщений;
- защите от отрицания фактов отправления и приема сообщений.

Перечисленные функции во многом связаны друг с другом, и их реализация основана на криптографической защите передаваемых данных. Высокая эффективность такой защиты обеспечивается за счет совместного использования симметричных и асимметричных криптографических систем.

4.1. Виртуальная частная сеть как средство защиты информации в корпоративных сетях

Благодаря развитию криптографических технологий появился способ решить задачи защиты информации в современной сетевой среде за счет использования технологии защищенных виртуальных частных сетей (Virtual Private Network – VPN), надежно шифрующих информацию, передаваемую по дешевым открытым сетям, включая Internet. Открытая сеть может служить основой для одновременного сосуществования множества виртуальных сетей, количество которых определяется пропускной способностью открытых каналов связи.

Под VPN понимают потоки данных одного предприятия, которые существуют в публичной сети с коммутацией пакетов и в достаточной степени защищены от влияния потоков данных других пользователей этой публичной сети. Другими словами, VPN – это некоторая имитация сети,

построенной на выделенных каналах. Если публичная сеть предоставляет такой сервис, то в ней одновременно сосуществуют несколько VPN, разделяющих общие коммутаторы и физические каналы связи.

Сети VPN решают задачи подключения корпоративного пользователя к удаленной сети и соединения нескольких сетей (рис. 4.1).

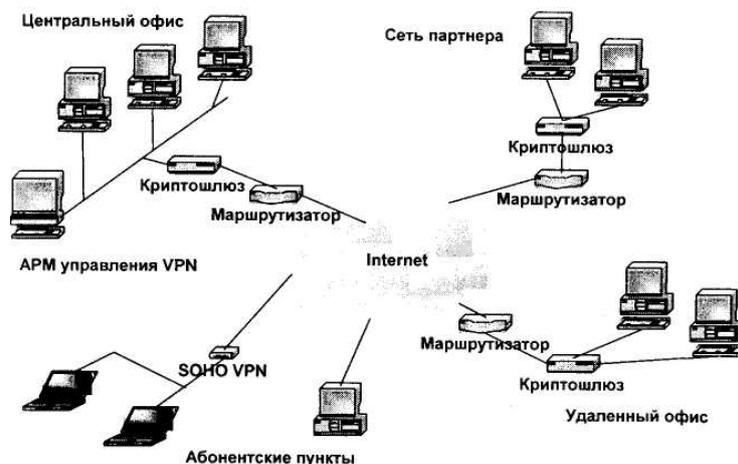


Рис. 4.1. Пример VPN

Цель VPN-технологий состоит в максимальной степени обособления потоков данных одного предприятия от потоков данных всех других пользователей публичной сети.

VPN-технологии обеспечивают:

- защиту (конфиденциальность, подлинность и целостность) передаваемой по сетям информации;
- защиту внутренних сегментов сети от НСД со стороны сетей общего пользования;
- контроль доступа в защищаемый периметр сети;
- сокрытие внутренней структуры защищаемых сегментов сети;
- идентификацию и аутентификацию пользователей сетевых объектов;
- централизованное управление политикой корпоративной сетевой безопасности и настройками VPN-сети;
- криптографическую защиту данных, передаваемых по каналам связи сетей общего пользования между защищаемыми сегментами сети;
- безопасный доступ пользователей VPN к ресурсам сетей общего пользования.

4.2. Схема виртуальной частной сети

На все компьютеры, имеющие выход в Internet, устанавливается средство, реализующее VPN (VPN-агент) (рис. 4.2).

VPN-агенты автоматически шифруют всю исходящую информацию (и соответственно расшифровывают всю входящую). Они также следят за ее целостностью с помощью электронной цифровой подписи (ЭЦП). Поскольку информация, циркулирующая в Internet, представляет собой

множество пакетов протокола IP, VPN-агенты работают именно с ними.

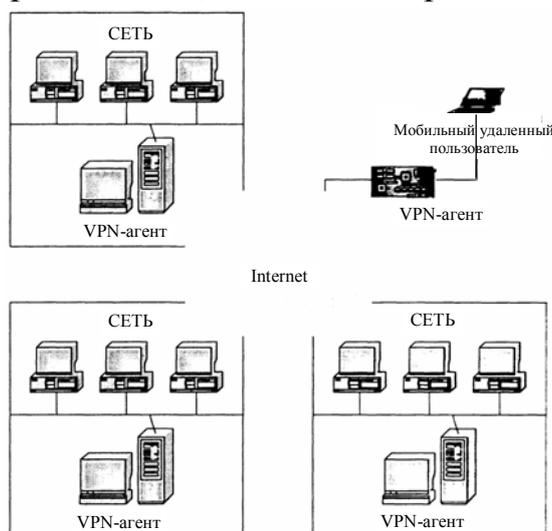


Рис. 4.2. Схема VPN

Перед отправкой IP-пакета VPN-агент действует следующим образом.

- Из нескольких поддерживаемых им алгоритмов шифрования и ЭЦП по IP-адресу получателя выбирает нужный для защиты данного пакета, а также ключи. Если же в его настройках такого получателя нет, то информация не отправляется.

- Генерирует и добавляет в пакет ЭЦП отправителя.
- Шифрует пакет (целиком, включая заголовок).
- Проводит инкапсуляцию, т. е. формирует новый заголовок, где указывается адрес вовсе не получателя, а его VPN-агента. Эта полезная дополнительная функция позволяет представить обмен между двумя сетями как обмен между двумя компьютерами, на которых установлены VPN-агенты. Всякая полезная для злоумышленника информация, например, внутренние IP-адреса, ему уже недоступна.

При получении IP-пакета выполняются обратные действия.

- Заголовок содержит сведения о VPN-агенте отправителя. Если таковой не входит в список разрешенных в настройках, то информация просто отбрасывается. То же самое происходит при приеме пакета с намеренно или случайно поврежденным заголовком.

- Согласно настройкам выбираются алгоритмы шифрования и ЭЦП, а также необходимые криптографические ключи.

- Пакет расшифровывается, затем проверяется его целостность. Если ЭЦП неверна, то он отбрасывается.

- Пакет в его исходном виде отправляется настоящему адресату по внутренней сети.

Все операции выполняются автоматически. Сложной в технологии VPN является только настройка VPN-агентов.

VPN-агент может находиться непосредственно на защищаемом ПК, что полезно для мобильных пользователей, подключающихся к Internet из

разных мест. В этом случае он обезопасит обмен данными только того компьютера, на котором установлен.

4.3. Управление криптографическими ключами в виртуальных частных сетях

В основе большинства средств защиты, применяемых в виртуальных частных сетях, лежат криптографические алгоритмы и протоколы или механизмы защиты, использующие такие алгоритмы. Обычно этот вопрос рассматривается в предположении, что участники VPN уже имеют секретные и открытые ключи, необходимые для выполнения криптографических операций, и им необходимо только грамотно использовать средства защиты, в которых реализованы соответствующие криптографические алгоритмы и протоколы. Однако на практике применение криптографических средств защиты информации сталкивается с целым рядом задач, связанных с необходимостью надлежащим образом распоряжаться криптографическими ключами участников VPN.

4.3.1. Жизненный цикл криптографических ключей

Актуальность задачи по управлению ключами, т. е. их генерации, распространению и самое главное сохранению в тайне от третьих лиц была еще определена в конце XIX в. и получила название «правило Керкхоффа». Согласно этому правилу стойкость криптосистемы определяется только степенью безопасности используемых в ней ключей и не зависит от секретности используемых алгоритмов шифрования.

Для обеспечения безопасности криптографических ключей нужно использовать любые доступные методы:

- технические средства охраны (изолированные помещения, сигнализация и т. п.);
- защищенную от взлома аппаратуру (при этом, однако, важно обеспечить надежную аутентификацию пользователя при работе с нею);
- концентрацию ключевого материала в небольшом количестве легко наблюдаемых и внушающих доверие компонентов системы.

Управление ключами (key management) является совокупность технологий и процедур, посредством которых устанавливаются и поддерживаются ключевые отношения между участниками криптографического протокола.

Ключевым отношением называется состояние, в котором взаимодействующие стороны разделяют общие данные – ключевой материал, – необходимые для выполнения криптографических алгоритмов и протоколов.

Таким образом, основными задачами управления ключами являются:

- обеспечение секретности, подлинности и целостности для секретных криптографических ключей;

- обеспечение подлинности и целостности для открытых криптографических ключей.

Здесь под секретными криптографическими ключами понимаются как общие секретные ключи симметричных криптосистем, так и секретные ключи асимметричных криптосистем (которые, в отличие от первых, не являются общими для двух или более абонентов, а известны исключительно своим владельцам).

Компрометация ключа – это событие, в результате которого произошла или могла произойти утрата одного из свойств криптографического ключа, обеспечивающего безопасность криптосистемы.

На практике дополнительной целью управления ключами является согласие с поддерживаемой политикой безопасности системы.

Основным международным стандартом в области управления криптографическими ключами является стандарт Международной организации по стандартизации и Международной электротехнической комиссии ISO/IEC 11770, состоящий из трех частей:

- ISO/IEC 11770-1 - Key management- Introduction.
- ISO/IEC 11770-2 - Key management - Symmetric techniques.
- ISO/IEC 11770-3 - Key management - Asymmetric techniques.

В указанном стандарте задачи управления ключами выводятся из задачи обеспечения безопасности криптографических ключей на всех этапах их жизненного цикла.

Жизненный цикл криптографического ключа – это последовательность состояний, в которых пребывает ключевой материал за время своего существования в криптосистеме.

Как известно, любой объект (например, любое промышленное изделие) имеет определенное, конечное время жизни, за которое он проходит определенные стадии своего развития от «рождения» до «гибели». Не являются в этом смысле исключением и криптографические ключи. Для любого объекта стандартизации стандартами ISO определяются четыре стадии жизненного цикла: предоперационная, операционная, постоперационная стадии и стадия выхода из эксплуатации.

Применительно к криптографическим ключам эти стадии означают следующее:

- на предоперационной стадии ключ еще не доступен для штатного использования в криптосистеме;
- находясь в операционной стадии жизненного цикла, ключ доступен пользователям криптосистемы и используется ими в штатном режиме;
- на постоперационной стадии ключ более не используется в штатном режиме, но он доступен в особом режиме для специальных целей;
- на стадии выхода из эксплуатации ключ более не доступен, а все

производит установку программного и аппаратного обеспечения, включая установку на него начального ключевого материала, полученного во время регистрации.

3. Генерация ключа: производится таким образом, чтобы гарантировать необходимые свойства для приложения или алгоритма и случайность (в смысле возможности предсказания его противником с пренебрежимо малой вероятностью). Субъект может генерировать свои собственные ключи или приобретать ключи от доверенного компонента криптосистемы.

4. Установка ключа: ключевой материал устанавливается для функционального использования в программном или аппаратном обеспечении субъекта, включая один из следующих способов: ручной ввод пароля или персонального кода, запись на магнитный диск, в постоянную память, микроэлектронную схему или другую аппаратуру. Начальный ключевой материал может служить для установления сеанса с доверенным компонентом криптосистемы, во время которого в реальном масштабе времени согласуются рабочие ключи. Во время последовательных обновлений новый ключевой материал заменяет используемый.

5. Регистрация ключа: одновременно с установкой ключа ключевой материал может быть публично записан как ассоциированный с уникальным именем субъекта системы. Для открытых ключей регистрация может выполняться специально выделенной третьей стороной, при помощи которой они становятся доступными для остальных субъектов через открытые директории или другие средства.

6. Штатное использование ключа: при нормальных обстоятельствах это состояние продолжается, пока не истечет период, предусмотренный регламентом системы (так называемый криптопериод ключа), но он может быть разделен, например, для пар секретного и открытого ключей шифрования (когда открытый ключ более не является действительным для шифрования, а секретный ключ остается в нормальном использовании для расшифрования). Цель жизненного цикла – содействовать операционной доступности ключевого материала для стандартных криптографических целей. Иными словами, фаза жизненного цикла является самой основной, центральной, ради которой поддерживаются все остальные фазы и инфраструктура криптосистемы.

7. Резервирование ключа: копирование ключевого материала на независимом, безопасном носителе обеспечивает источник данных для восстановления ключа. Резервирование подразумевает кратковременное хранение во время операционного использования.

8. Обновление ключа: по истечении периода, предусмотренного регламентом системы, операционный ключевой материал заменяется на новый. Это может включать процедуры генерации ключа, наследования ключа, выполнение протоколов распределения ключей. Для открытых

ключей обновление и регистрация новых ключей обычно включает безопасные коммуникационные протоколы с доверенной третьей стороной.

9. Архивация ключа: ключевой материал, более не используемый в штатном режиме, может быть заархивирован, чтобы обеспечить источник восстановления ключа при специальных обстоятельствах (например, в случае возникновения конфликтов о принадлежности ЭЦП). Архивация подразумевает долговременное хранение ключей в постоперационной стадии, при этом могут применяться алгоритмы сжатия данных с целью сокращения объема хранимых ключей.

10. Вывод из эксплуатации и уничтожение: когда более нет необходимости поддерживать ассоциацию ключа с субъектом, ключ выводится из эксплуатации, т. е. удаляется из всех публичных записей, и все копии ключа уничтожаются. В случае секретных ключей должны быть безопасно стерты все «следы» ключа.

11. Восстановление ключа: если ключевой материал потерян, но при этом не случилось компрометации (сбой оборудования, забыт пароль), должно быть возможно восстановить ключевой материал с безопасной резервной копии.

12. Аннулирование ключа: может быть необходимо удалить ключи из операционного использования до истечения предполагаемого срока по различным причинам, включая компрометацию ключей или выбытие владельца ключа из системы.

4.3.2. Особенности управления ключевой системой асимметричных криптосистем. Концепция инфраструктуры открытых ключей

Изображенный на рис. 4.3 жизненный цикл ключа более соответствует асимметричным криптосистемам. В симметричных он, как правило, проще (например, сеансовые ключи не регистрируются, не резервируются, не восстанавливаются, не архивируются).

С точки зрения обеспечения безопасности криптографических ключей ни одна из указанных стадий жизненного цикла и ни одно из состояний ключа не является более или менее важным по сравнению с другими, т. е. если безопасность ключа нарушена хотя бы на одной из стадий, хотя бы в одном из состояний или при переходе из одного состояния в другое, то это приведет к нарушению безопасности криптосистемы в целом. Таким образом, при разработке и эксплуатации систем криптографической защиты информации большое внимание должно уделяться вопросам грамотной организации управления криптографическими ключами на всех этапах их жизненного цикла.

Если рассматривать жизненный цикл ключей с точки зрения сложности реализации мер обеспечения безопасности ключей, выделяют

четыре наиболее сложные фазы, на которых необходимо решать задачи управления ключами: генерация, распространение, хранение и уничтожение ключей. Из них самой сложной для реализации и самой потенциально опасной является фаза распространения ключей, включающая транспортировку ключей между участниками криптосистемы.

В последнее время преимущественное распространение получили такие средства и системы защиты информации, в которых ведущую роль играют методы криптографической защиты информации, основанные на асимметричных криптосистемах, а симметричные криптосистемы играют по отношению к ним подчиненную роль и используются как необходимое средство в тех случаях, когда асимметричные методы не удовлетворяют требованиям производительности криптосистемы. Ярким примером такого подхода являются, в частности, виртуальные частные сети. В связи с этим ведущую роль приобретает именно организация управления ключами асимметричных криптосистем. Она, в свою очередь, включает две подзадачи: управление частными секретными ключами участников и управление их открытыми ключами.

Задача управления секретными ключами здесь проще, чем в симметричных криптосистемах, так как секретные ключи никогда не выходят за пределы собственности их владельцев: нет необходимости передавать их по каким-либо каналам связи, распространять среди других участников.

Задача управления открытыми ключами является новой по сравнению с симметричными криптосистемами. Идея решения задачи по управлению открытыми ключами выражается в создании специальной инфраструктуры, получившей название *инфраструктуры открытых ключей* (Public Key Infrastructure).

Под инфраструктурой понимаются составные части общего устройства системы, носящие вспомогательный, подчиненный характер и обеспечивающие нормальную деятельность системы в целом.

Инфраструктура открытых ключей (ИОК) – это универсальная концепция организованной поддержки криптографических средств защиты информации в крупномасштабных информационных системах в соответствии с принятыми в них политиками безопасности, которая реализует управление криптографическими ключами на всех этапах их жизненного цикла, обеспечивая взаимодействие всех средств защиты распределенной системы.

Логически ИОК объединяет механизмы, субъекты, правила и взаимосвязи, которые необходимы для доступа к криптографическим ключам и для ассоциирования открытых криптографических ключей со своими владельцами.

Физически ИОК состоит из программ, форматов данных,

коммуникационных протоколов, политик и процедур, требуемых для использования в организации криптосистем с открытым ключом.

Важно иметь в виду, что создание инфраструктуры открытых ключей имеет целью комплексную поддержку всего *жизненного цикла открытых криптографических ключей* в целом, а не только каких-либо отдельных его фаз, например, распространения ключей (хотя последняя задача и является технически наиболее сложной).

4.3.3. Метод сертификации открытых ключей

Наиболее сложным для обеспечения безопасности открытых ключей является этап распространения их среди участников криптосистемы.

Среди известных методов наиболее распространенным является метод сертификации открытых ключей. Рассмотрим кратко основное содержание этого метода. Пусть имеется криптосистема, включающая большое число участников (абонентов), например, VPN (рис. 4.4). Среди участников криптосистемы выделяется специальный участник, которому доверяют все остальные, получивший название «центр сертификации ключей», или «агентство сертификации» (Certification Authority), или «удостоверяющий центр» (УЦ). Его функции может выполнять, например, администратор системы, оснащенный соответствующим аппаратным и программным обеспечением (сервер регистрации и сертификации ключей). Все остальные участники являются обычными, «рядовыми» абонентами криптосистемы, например, узлами VPN, выполняющими функции криптографической защиты данных.

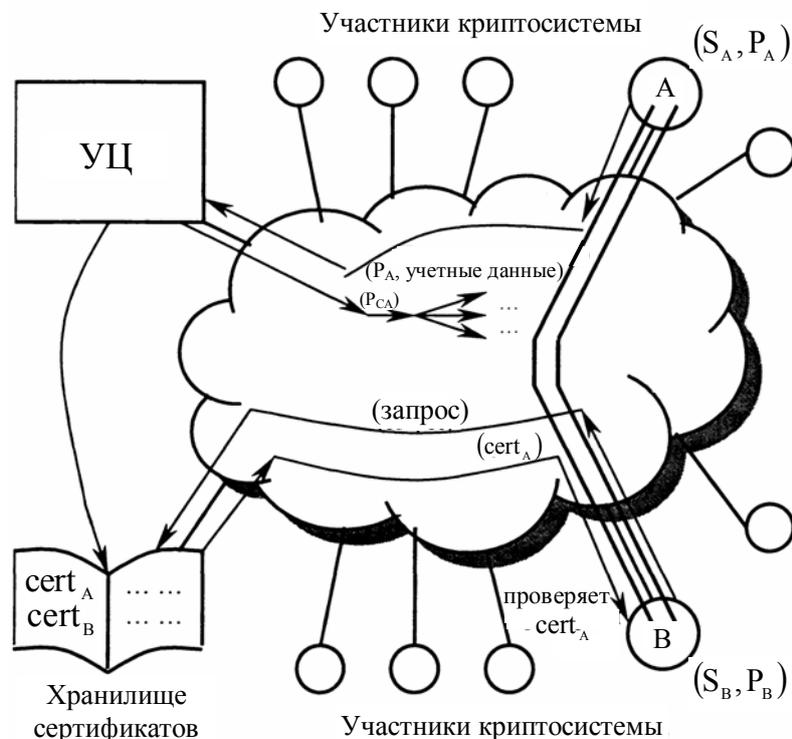


Рис. 4.4. Процессы получения и использования сертификатов участниками криптосистемы

При введении в систему каждого из этих участников – возьмем для примера участника A – он должен пройти процедуру регистрации в криптосистеме (что соответствует первой фазе жизненного цикла его криптографического ключа). Для этого он контактирует с УЦ, чтобы зарегистрировать свой открытый ключ и получить от него так называемый сертификат своего открытого ключа. УЦ должен проверить представленные ему учетные данные, а также (что очень важно!) *знание секретного ключа, соответствующего представленному для регистрации открытому ключу*. Решить эту задачу можно различными способами: в самом простом случае УЦ может попросить A зашифровать на своем секретном ключе текст заданного формата и проверить правильность его расшифрования при помощи представленного открытого ключа или воспользоваться протоколом доказательства знания ключа с нулевым разглашением знаний.

Сертификат открытого ключа – специальная структура данных, состоящая из полей данных и поля подписи. Поле данных содержит, как минимум, какие-либо признаки абонента (идентификатор, атрибуты) и его открытый ключ. Поле подписи – это ЭЦП УЦ под полем данных, логически связывающая признаки абонента с его открытым ключом.

Все абоненты, заинтересованные в связи с абонентом A , получают впоследствии его сертификат либо путем обмена с абонентом A , либо извлекая его из открытого общедоступного справочника, который заводится в криптосистеме. Сертификат, таким образом, является средством для хранения, распространения и передачи через небезопасные каналы связи открытых ключей без опасения их необнаружимого изменения.

Различают две формы сертификатов открытых ключей: идентификационные и атрибутивные.

В идентификационном сертификате обязательно присутствует идентификатор субъекта – владельца ключа, по которому можно однозначно установить его личность. Основным стандартом по идентификационным сертификатам является стандарт Международного телекоммуникационного Союза ITU X.509. Сертификат состоит из двух полей: поля данных и поля подписи. Поле данных содержат необходимые данные, а поле подписи содержит ЭЦП УЦ открытых ключей под полем данных. Наиболее разработанными и широко применяемыми на практике логическими моделями инфраструктуры открытых ключей на базе идентификационных сертификатов являются:

- X9.55 – американский стандарт для финансовой индустрии;
- PKIX – проект стандарта IETF на базе стандарта X.509v3, адаптирующий положения этого стандарта для использования в сети Internet;
- APKI – архитектура для ИОК, описанная в документах The Open Group.

Однако использование идентификационных сертификатов не всегда желательно для пользователей, т. к. при этом может происходить доступ к информации, не имеющей отношения к тому случаю, по которому необходим данный конкретный факт доступа к сертификату. Возможность однозначно установить личность владельца по сертификату может привести к установлению «тотального контроля» над действиями участников криптосистемы в информационной системе. В этой связи было предложено использовать атрибутивную форму сертификатов.

Атрибутивные сертификаты связывают открытый ключ с одним или более «атрибутов», которые в соответствии со стандартом Международного телекоммуникационного Союза X.501 (ISO/IEC 9594-2) определяются как «информация любого типа». Таким образом, один и тот же участник в зависимости от ситуации и используемой прикладной программы может предстать в разных «ипостасях», между которыми невозможно установить однозначную связь. К примеру, атрибутом может быть роль пользователя в информационной системе, например, путем указания его должности. Тогда можно реализовать модель управления доступа «по ролям», т. е. участники системы, занимающие одну и ту же должность, имеют абсолютно одинаковые права в системе, и невозможно установить, кто именно из них совершил конкретное действие с применением данного конкретного сертификата.

Наиболее разработанными и широко применяемыми на практике логическими моделями инфраструктуры открытых ключей на основе атрибутивных сертификатов являются:

- X9.57 – американский стандарт для финансовой индустрии;
- SPKI – проект стандарта IETF для использования в сети Internet.

УЦ открытых ключей – это специально выделенный участник криптосистемы, которому доверяют все остальные участники («центр доверия»), чья подпись служит гарантией подлинности ключей и который выполняет следующие функции:

- сбор сведений об участниках системы, необходимых для сертификации: имя, почтовый адрес, права доступа, должность, номер кредитной карты и т. п. (зависит от конкретного приложения);
- генерация и рассылка (либо помещение в общедоступное хранение) сертификатов открытых ключей;
- уничтожение сертификатов с истекшим сроком годности;
- обновление сертификатов; аннулирование сертификатов.

Аннулирование сертификата может потребоваться в случаях, когда срок санкционированного использования открытого ключа участника системы прерывается досрочно, ранее, чем это предусмотрено принятым в системе регламентом, например:

- при компрометации секретного ключа участника криптосистемы, соответствующего данному открытому ключу;

- при удалении (выбытии) пользователя из системы;
- при смене роли пользователя в системе (перемещении пользователя).

Аннулирование сертификата – это чрезвычайное обстоятельство, в котором необходимо оповестить всех участников криптосистемы. Существуют два способа решения этой задачи.

- *Проверка статуса сертификата в режиме реального времени.* Для этого требуется выполнение специального протокола с УЦ открытых ключей, который отвечает на вопрос, не был ли запрошенный сертификат аннулирован.

- *Периодическое создание и рассылка списков аннулированных сертификатов (CRL - Certificate Revocation List).* Формат CRL определен в стандарте ITU X.509 v2. Второй способ на практике используется чаще. Однако у него есть один существенный недостаток: между двумя последовательными рассылками списка аннулированных сертификатов всегда существует какой-то временной «зазор», т. е. об аннулировании сертификата какого-либо участника все остальные участники узнают не мгновенно, а только по прошествии некоторого времени. Наличие такого разрыва создает угрозу несанкционированного использования аннулированного ключа.

УЦ рассылает всем участникам системы свой открытый ключ который нужен им для проверки подписи на сертификатах. Считается, что подменить его невозможно в силу трех причин: массовости рассылки, периодического повтора и общедоступности.

В криптосистемах с большим числом участников или с большой интенсивностью потока требований к УЦ функции регистрации участников нередко возлагают на специально выделяемый центр регистрации (Registration Authority – RA).

Сертификаты всех участников криптосистемы могут либо храниться в специальном общедоступном хранилище, либо рассылаться по сети. На практике преимущественно используется первый способ, причем физически хранилище сертификатов чаще всего реализуется либо с использованием директориального сервиса (как директория, к которой открыт доступ на чтение всем участникам), либо веб-сервиса (как веб-страница, с которой все участники криптосистемы могут забирать сертификаты).

Преимущество сертификата состоит в том, что два участника системы (к примеру, два узла VPN), доверяющие одному и тому же УЦ, могут не знать и не хранить открытые ключи никаких других абонентов, а при необходимости обратиться в УЦ и получить необходимые ключи. Для этого ему достаточно знать только открытый ключ УЦ. Это позволяет применять метод сертификации открытых ключей в криптосистемах со сколь угодно большим и даже неопределенным числом участников, где все

участники не покрыты непосредственной сетью контактов между собой.

Для того, чтобы узнать открытый ключ любого интересующего его абонента, участнику криптосистемы (обозначим его B) необходимо однократно приобрести аутентичный открытый ключ УЦ открытых ключей (что технически реализовать не сложно). Далее для установления связи с абонентом A ему необходимо:

1) приобрести сертификат открытого ключа абонента A одним из следующих способов: обратившись в хранилище сертификатов, непосредственно получив его от УЦ или от абонента A (зависит от порядка, установленного в системе);

2) выполнить процедуру проверки сертификата, состоящую из следующих действий:

- проверки текущей даты и времени и сравнения с периодом действия сертификата;
- проверки действительности в данный момент времени открытого ключа самого УЦ;
- проверки подписи УЦ на сертификате открытого ключа абонента A , используя открытый ключ УЦ;
- проверки, не был ли сертификат аннулирован к текущему моменту времени;

3) в случае, если все проверки окончились с положительным результатом, принять открытый ключ, извлеченный из сертификата абонента A как аутентичный ключ.

Далее абонент B может использовать открытый ключ абонента A для выполнения любых необходимых ему криптографических алгоритмов или протоколов. К примеру, участники A и B , приобретя таким образом открытые ключи друг друга, могут выполнить протокол открытого распределения Диффи-Хеллмана, выработать с его помощью общий секретный ключ для симметричной криптосистемы и использовать далее для шифрования трафика VPN какой-либо симметричный алгоритм: DES, ГОСТ 28147-89 и др.

Хотя сертификат является вспомогательным средством для транспортировки и обеспечения подлинности открытого ключа, он, как и сам ключ, имеет свой жизненный цикл. В его жизненном цикле можно выделить те же стадии и ряд состояний, в которых пребывает сертификат: создание, рассылка, штатное использование, обновление или аннулирование, уничтожение сертификата.

Недостаток метода сертификации открытых ключей заключается в том, что часто участникам криптосистемы после проверки сертификата все равно необходим доступ в базу данных об участниках криптосистемы в реальном масштабе времени (например, для проверки аннулирования сертификата либо чтобы получить какие-то дополнительные данные об этом участнике, необходимые для работы прикладных программ). Это не

всегда удобно, поэтому сейчас наблюдается «откат» к модели ИОК с УЦ ключей, работающих в реальном масштабе времени.

Контрольные вопросы

1. Какие задачи решает построение VPN, а какие – установка МЭ?
2. Какие услуги по защите данных обеспечивают VPN?
3. Что поднимается под термином управление криптографическими ключами? Какова основная цель и основные задачи управления ключами?
4. Что такое жизненный цикл ключа? Какие его основные стадии?
5. В каких состояниях пребывают криптографические ключи за время своего жизненного цикла? При каких условиях происходят переходы из одного состояния в другое?
6. Что такое инфраструктура открытых ключей? Какова ее логическая и физическая структура?
7. Каковы основные способы распространения открытых ключей в криптосистемах?

Литература

1. Столингс В. Современные компьютерные сети / В. Столингс. – 2-е изд. – СПб. : Питер, 2003. – 783 с.
2. Ретана А. Принципы проектирования IP-сетей / А. Ретана, Д. Слайс, Р. Уайт. – М. : Вильямс, 2002. – 368 с.
3. Проектирование корпоративных информационных систем / А.А. Рындин и др.; под ред. А.А. Рындина. – Воронеж : Кварта, 2003. – 447 с.
4. Вегешна Ш. Качество обслуживания в сетях IP / Ш. Вегешна. – М. : Вильямс, 2002. – 367 с.
5. Запечников С.В. Основы построения частных сетей / С.В. Запечников, Н.Г. Милославская, А.И. Толстой. – М. : Горячая линия – Телеком, 2003. – 249 с.
6. Норткатт С. Защита сетевого периметра / С. Норткатт. – К. : ТИД «ДС», 2004. – 672 с.
7. Милославская Н.Г. Интрасети: доступ в Internet, защита / Н.Г. Милославская, А.И. Толстой. – М. : ЮНИТИ-ДАНА, 2000. – 527 с.
8. Андрейчиков А.В. Интеллектуальные информационные системы / А.В. Андрейчиков, О.В. Андрейчикова. – М. : Финансы и статистика, 2004. – 424 с.

Содержание

Введение.....	3
1 Корпоративные информационные системы.....	3
1.1 Основные задачи корпоративной ИС.....	3
1.2 Классификация информационных систем.....	6
1.3 Технологии корпоративных ИС.....	8
1.3.1 OLTP-системы (On-Line Transaction Processing).....	8
1.3.2 Хранилища данных (Data Warehouse).....	10
1.3.2.1 Очистка данных в Хранилище данных.....	10
1.3.2.2 Структура Хранилища данных.....	12
1.3.2.3 Классификация данных в Хранилище данных.....	16
1.3.2.4 Моделирование хранилищ данных.....	17
1.3.2.5 Моделирование витрин данных.....	19
1.3.3 Способы аналитической обработки данных.....	21
1.3.3.1 Системы поддержки принятия решений.....	21
1.3.3.1.1 Алгоритмы СППР.....	22
1.3.3.1.2 Варианты архитектуры СППР.....	23
1.3.3.2 Оперативная аналитическая обработка (On-Line Analytical Processing, OLAP).....	27
1.3.3.2.1 Многомерная модель данных в OLAP.....	28
1.3.3.2.2 OLAP на клиенте и на сервере.....	34
1.3.3.2.3 Технические аспекты многомерного хранения данных.....	35
1.3.3.3 Интеллектуальный анализ данных (Data Mining).....	37
1.3.3.4 Интеграция OLAP и Data Mining.....	42
2 Стандарты корпоративных ИС.....	44
2.1 MRP (Material Requirements Planning).....	44
2.2 MRP II (Manufactory Resource Planning).....	45
2.3 ERP (Enterprise Requirements Planning).....	47
2.4 MPC (Management Planning and Control).....	48
2.5 CSRP (Customer Synchronized Resource Planning).....	50
2.6 Системы автоматизации документооборота.....	52
2.6.1 Типы носителей информации в документообороте.....	53
2.6.2 Подходы к организации хранения электронных документов.....	54
2.6.3 Требования, предъявляемые к системе документооборота.....	54
2.6.4 Задачи, решаемые системами документооборота.....	55
2.6.4.1 Задачи статических архивов.....	55
2.6.4.2 Задачи динамических архивов.....	57
3 Принципы иерархического проектирования корпоративных сетей.....	59
3.1 Структура корпоративных информационных систем и их элементов.....	59

3.2	Топология иерархических сетей	61
3.2.1	Ядро сети.....	62
3.2.2	Уровень распределения	63
3.2.3	Уровень доступа.....	63
3.3	Адресация и суммирование.....	63
3.3.1	Суммирование в сетях IP	63
3.3.2	Стратегии адресации	67
3.4	Избыточность в иерархических сетях	71
3.4.1	Избыточность, реализуемая на уровне ядра сети	72
3.4.1.1	Структура ядра с резервными связями.....	72
3.4.1.2	Структура ядра типа «кольцо».....	73
3.4.1.3	Структура ядра с полным объединением.....	74
3.4.1.4	Структура ядра с частичным объединением	75
3.4.2	Избыточность, реализуемая на уровне распределения сети ..	76
3.4.2.1	Двойное подключение маршрутизаторов уровня распределения к маршрутизаторам ядра	76
3.4.2.2	Использование резервных каналов передачи информации для соединения маршрутизаторов уровня распределения.....	77
3.4.3	Избыточность, реализуемая на уровне доступа сети	78
4	Безопасность корпоративных сетей	81
4.1	Виртуальная частная сеть как средство защиты информации в корпоративных сетях	81
4.2	Схема виртуальной частной сети	82
4.3	Управление криптографическими ключами в виртуальных частных сетях.....	84
4.3.1	Жизненный цикл криптографических ключей	84
4.3.2	Особенности управления ключевой системой асимметричных криптосистем. Концепция инфраструктуры открытых ключей.....	88
4.3.3	Метод сертификации открытых ключей	90
	Литература	96

Учебное издание

Борисов Дмитрий Николаевич

КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Учебное пособие для вузов

Редактор А.П. Воронина

Подписано в печать 23.05.07. Формат 60×84/16. Усл. печ. л. 5,75.
Тираж 50 экз. Заказ 1019.

Издательско-полиграфический центр
Воронежского государственного университета.
394000, г. Воронеж, пл. им. Ленина, 10. Тел. 208-298, 598-026 (факс)
<http://www.ppc.vsu.ru>; e-mail: pp_center@typ.vsu.ru

Отпечатано в типографии Издательско-полиграфического центра
Воронежского государственного университета.
394000, г. Воронеж, ул. Пушкинская, 3. Тел. 204-133.