

Федеральное агентство по образованию Российской Федерации
Самарский государственный медицинский университет
Самарская государственная экономическая академия
Нижевартовский государственный гуманитарный университет

**Н.В.Абрамов, Н.В.Мотовилов,
Н.Д.Наумов, С.Н.Черкасов**

ИНФОРМАЦИОННЫЕ СИСТЕМЫ В МЕДИЦИНЕ

Учебное пособие

*Рекомендовано Учебно-методическим объединением
по методическому и фармацевтическому образованию вузов России
в качестве учебного пособия для студентов медицинских вузов*



**Издательство
Нижевартовского государственного
гуманитарного университета
2008**

**ББК 5я73
И 74**

Рецензенты:

доктор медицинских наук, профессор, зав. кафедрой
общественного здоровья, здравоохранения и общей гигиены
Ульяновского государственного университета *В.И.Горбунов*;
доктор технических наук, профессор, зав. кафедрой
«Информационные системы» Самарского государственного
технического университета *В.И.Батищев*

И 74 Информационные системы в медицине: Учебное пособие /
Н.В.Абрамов, Н.В.Мотовилов, Н.Д.Наумов, С.Н.Черкасов. —
Нижевартовск: Изд-во Нижневарт. гуманит. ун-та, 2008. — 171 с.

ISBN 5–89988–459–2

Учебное пособие «Информационные системы в медицине» посвящено вопросам разработки и эффективного применения информационных систем, сетей и технологий в решении различных классов медицинских задач. Представлена систематизация концепций информационных систем в медицине, рассмотрены вопросы телемедицины, работы в Интернете. Приведено краткое описание принципов работы систем на основе баз данных, экспертных систем и систем на основе нейронов.

Для студентов, аспирантов и преподавателей, занимающихся вопросами применения информационных систем в медицине.

ББК 5я73

ISBN 5–89988–459–2

© Издательство НГГУ, 2008

Введение

Администрации, врачам и другим сотрудникам медицинских учреждений приходится сталкиваться с огромными объемами информации. С появлением персональных компьютеров началось стихийное развитие информационного пространства практически во всех медучреждениях. Каждый владелец ПК, используя простейшие средства, начал вести свой учет и делать отчеты, зачастую просто внося цифры в готовый шаблон. Персональные компьютеры (далее — ПК), объединенные в локальные сети, позволяют пользоваться и обмениваться информацией сразу несколькими сотрудниками.

Следующим шагом в развитии медицинских систем стало создание мест хранения общей информации. В информационных технологиях такие возможности реализуются базами данных. Сервер базы данных организует работу с пользователями и данными, а именно: создание мест хранения, обеспечение доступа к данным, сохранность информации и многое другое. В первых разработках, разумеется, использовались доступные базы данных, такие как FoxPro, Paradox, Access (либо такого же уровня). Хранилища наполнялись данными, расширялся список необходимой информации, повышались требования к самим базам данных. Такой же путь прошли и другие области человеческой деятельности. Так сложилось, что наибольшее развитие информационные системы получили в финансовой и производственной деятельности человека. Были разработаны информационные инфраструктуры для больших предприятий с сотнями пользователей и едиными базами данных. Наступил момент использовать то лучшее в информационных технологиях, что было накоплено в производстве во благо здоровья человека.

Место информационной системы на предприятии такое же, как у нервной системы в организме человека. Подобно тому, как здоровье человека в значительной степени зависит от состояния его нервной системы (известно, что все болезни — от нервов), так и жизнеспособность предприятия во многом зависит от его информационной системы.

Как происходит выбор медицинской информационной системы (МИС). Какую систему купить? Что хотят получить от системы

специалисты и администрация? Пользователи смотрят прежде всего на функциональность системы.

Если суммировать пожелания специалистов, можно выделить несколько основных пунктов:

- стремление иметь быстрый доступ к актуальной, наиболее полной и достоверной информации (имеются в виду все данные о пациенте, его клиническая история, своевременное поступление результатов анализов, просмотр рентгеновских снимков и т.д.);

- врачебная этика требует обеспечения конфиденциальности информации;

- врачи и администрация должны быть освобождены от трудоемкого процесса составления отчетов;

- необходимо прекратить зависимость от почерка врача и от многократного, повторного ввода данных;

- следует избегать потерь информации;

- иметь возможность удаленного доступа к информации по компьютерной сети (в частности, через Интернет), что порой необходимо для принятия оперативных решений при оказании скорой помощи и при вызовах врача на дом;

- необходимо обеспечить механизм консультации со специалистами из других учреждений.

- рабочее место медицинского работника должно отвечать современным требованиям: удобный и интуитивно понятный пользовательский интерфейс, показ только необходимых для работы форм.

Требования управляющего персонала медицинского учреждения к информационной системе подобны требованиям руководства любого предприятия. В числе этих требований следующие:

- иметь быстрый доступ к актуальной, полной информации своего учреждения;

- располагать отчетной информацией за любой выбранный промежуток времени или по выбранному критерию;

- контролировать и анализировать работу персонала;

- обладать эффективным средством анализа стекающейся информации;

- качественно изменить уровень медицинского обслуживания в лечебных учреждениях и повысить эффективность их работы.

Чтобы выполнить вышеперечисленные требования, необходимо создать информационную инфраструктуру, которая объединит всю информацию по учреждению, обеспечит хранение и прямой доступ к данным.

С точки зрения информационных технологий важно правильно выбрать среду разработки внедряемой системы. В век компьютеризации человеческой деятельности рано или поздно перед лечебно-профилактическими учреждениями возникает необходимость автоматизации. При этом следует обратить особое внимание не только на функциональность предлагаемой системы, но и на среду реализации данного продукта. Только тогда, когда выбрана достаточно мощная среда разработки, можно быть уверенным, что вся функциональность реализуется в полной мере.

Функциональность	Полнота информации	Быстрый доступ	Сохранность информации	Удаленный доступ
	Конфиденциальность	Точность информации	Актуальность	Защищенность от сбоев
Пользователи	Администрация	Регистратор	Статистик	Лаборант
	Врач 1	Врач 2	Врач 3	Врач 4
Загрузка системы	Данные	Данные	Данные	Данные
Информационная инфраструктура	База данных	Среда разработки	Прикладные программы	Программы анализа данных
Материально-техническое обеспечение	Сервера	ПК	Локальная сеть	Приборы

Информационное обеспечение любого предприятия постоянно развивается — это закон. Систему можно представить как строение, где в фундамент мы закладываем информационную среду, программные продукты и материально-техническое обеспечение, включающее в себя ПК, сервера, локальную сеть, приборы. На этот фундамент накладывается информация, увеличивающееся количество пользователей и разрастающаяся функциональность.

1. Информационные системы

1.1. Понятие информации

Информация — это совокупность фактов, явлений, событий, представляющих интерес, подлежащих регистрации и обработке, в том числе машинной, кодовой. Рассматриваемое понятие предполагает существование двух партнеров: источника и потребителя информации. Как первым, так и вторым могут быть объекты науки, техники, общества и природы, животные, люди. Во взаимодействии между ними и рождается информация. В зависимости от области знаний и сферы применения различают научную, учебную, патентную, техническую, коммерческую и другие виды информации. Передают информацию устно и письменно, с помощью электрических сигналов и радиоволн. Получают с помощью органов чувств, датчиков, основанных на различных физических, химических или биологических законах. Отдельные данные и сообщения обрабатывают, преобразовывают, систематизируют, сортируют и получают новую информацию, или новые знания. Информация может быть постоянной, переменной, случайной, вероятностной. Наибольший интерес представляет переменная информация, т.к. она позволяет выявлять причинно-следственные связи в процессах и явлениях.

Существует несколько форм представления информации:

- **символьная (вербальная) форма** — основана на использовании символов (например, букв, цифр, знаков, в том числе знаков пунктуации);
- **текстовая (вербальная) форма** — использует образующие тексты символы, но расположенные в определенном порядке;
- **графическая форма** — к ней относятся различные виды изображений, в том числе анимация и видео;
- **звуковая форма** является аналоговой и для своего хранения требует применения аналого-цифровых (АЦП) и цифро-аналоговых преобразователей (ЦАП).

Информация, представленная в удобном для обработки виде, называется **данными**. Определенная структура информационного объекта, подвергаемого обработке, называется **форматом**. Представление информации в технических средствах реализуется

посредством интерпретаторов. В настоящее время большинство из них использует двоичный код, позволяющий обеспечить необходимую лексику для универсального отображения информации на машинных языках.

Виды информации могут быть разделены на две группы:

1. Статическая информация. Числовая, логическая и символьная информация является статической — ее значение не связано со временем.

2. Динамическая информация. Классическим видом является *аудиоинформация*. Она существует только в режиме реального времени, ее нельзя остановить для более подробного изучения. Если изменить масштаб времени (увеличить или уменьшить), аудиоинформация искажается. Это свойство используется для получения звуковых эффектов.

Видеоинформация может быть как статической, так и динамической. Статическая видеоинформация включает текст, рисунки, графики, чертежи, таблицы. Рисунки делятся на плоские (двухмерные) и объемные (трехмерные).

Динамическая видеоинформация — это видео, анимация, слайд-фильмы. В их основе лежит последовательное экспонирование на экране в реальном времени отдельных кадров в соответствии со сценарием.

Динамическая видеоинформация используется либо для передачи движущихся изображений (анимация), либо для последовательной демонстрации отдельных кадров вывода (слайд-фильмы). Для демонстрации анимационных и слайд-фильмов используются различные принципы. Анимационные фильмы демонстрируются так, чтобы зрительный аппарат человека не мог зафиксировать отдельные кадры. В современных высококачественных мониторах и в телевизорах с цифровым управлением электронно-лучевой трубкой кадры сменяются до ста раз в секунду, что позволяет высококачественно передавать движение объектов. При демонстрации слайд-фильмов каждый кадр экспонируется на экране столько времени, сколько необходимо для восприятия его человеком (обычно от 30 секунд до 1 минуты). Слайд-фильмы можно отнести к статической видеоинформации.

По способу формирования видеоизображения бывают растровые, матричные и векторные:

- растровые видеоизображения используются в телевидении, а в ЭВМ практически не применяются;
- матричные изображения получили в ЭВМ наиболее широкое распространение.

Изображение на экране рисуется электронным лучом точками. Векторные изображения состояются из отрезков линий (в простейшем случае — прямых), для которых задаются начальные координаты, угол наклона и длина отрезка (может указываться и код используемой линии). Векторный способ имеет ряд преимуществ перед матричным: изображение легко масштабируется с сохранением формы, является «прозрачным», может быть наложено на любой фон и т.д.

Для того чтобы в материальном мире происходили обмен информацией, ее преобразование и передача, должны быть носитель информации, передатчик, канал связи, приемник и получатель информации. Среда передачи объединяет источник и получатель информации в информационную систему (рис. 1.1).



Рис. 1.1. Информационная система

Подобные информационные системы возникают не только среди людей. Обмен информацией происходит и в животном, и в растительном мире. Если же участником информационной системы является человек, то речь идет о смысловой информации, т.е. информации, выражаемой человеком.

Получатель информации оценивает ее в зависимости от того, для какой задачи информация будет использована. Поэтому информация имеет свойство относительности. Одна и та же информация для одного получателя может иметь глубокий смысл и обладает чрезвычайной ценностью, а для другого она может быть либо уже давно известной, либо бесполезной.

Данные — это информация, представленная в формализованном виде, пригодном для автоматической обработки при возможном участии человека.

Данные в алгоритмическом аспекте — это информация, несущая полезную смысловую нагрузку, представленная в формализованном виде, позволяющем собирать, передавать, вводить и обрабатывать эту информацию с помощью заданных алгоритмов.

Данные формируются в группы, образуя компоненты **банков и баз данных и баз знаний**. Наименьшим компонентом является **элемент данных** — информационный объект, определяемый его наименованием и совокупностью описывающих его значений (величин). Объектом могут быть и процесс, и явление, и предмет, и страна, и область науки и т.д.

Запись — совокупность значений элементов данных, которая описывает рассматриваемый объект. Для передачи данных последние формируются в блоки данных, для хранения — компонуются в файлы, каталоги, массивы, таблицы, списки. При этом в системах и сети может использоваться несколько способов кодирования данных. Часто в интересах государства либо в интересах коммерческих предприятий, организаций, фирм и отдельных граждан возникает необходимость защиты данных от искажения, противозаконного использования. Все это ставит *проблему безопасности данных*.

Экономическая информация отражает процессы производства, распределения, обмена и потребления материальных благ и услуг. В связи с тем, что экономическая информация большей частью связана с общественным производством, ее часто называют **производственной информацией**.

Экономическая информация характеризуется большим объемом, многократным использованием, обновлением и преобразованием, большим числом логических операций и относительно несложных математических расчетов для получения многих **видов результатной информации**.

Структурной единицей экономической информации является **показатель**. Он представляет собой контролируемый параметр экономического объекта и состоит из совокупности реквизитов.

Реквизит имеет законченное смысловое содержание и потребительскую значимость. **Реквизит** — это логически неделимый элемент показателя, отражающий определенные свойства объекта или процесса. Реквизит нельзя разделить на более мелкие единицы без разрушения его смысла.

Каждый показатель состоит из одного реквизита-основания и одного или нескольких реквизитов-признаков. *Реквизит-признак* характеризует смысловое значение показателя и определяет его наименование. *Реквизит-основание* характеризует, как правило, количественное значение показателя.

Реквизиты объединяются в показатели, показатели — в записи, записи — в массивы, из массивов создаются комплексы массивов, а из комплексов — информационные базы. Структурная теория позволяет на логическом уровне построить оптимальную структуру информационной базы, которая затем с помощью определенных средств реализуется на физическом уровне — на уровне технических устройств хранения информации. От выбранной структуры хранения зависит такой важный параметр, как время доступа к данным, т.е. структура влияет на время записи и считывания информации, а значит, и на время создания и реорганизации информационной базы.

1.2. Информационный обмен и информационный процесс

Принятие решений относительно действий или поведения в той или иной ситуации любых субъектов (людей, животных, роботов, сложных систем управления) осуществляется на основе информационного обмена и информационных процессов. Информационный процесс реализует отношение объекта и субъекта (рис. 1.2.) и представляет собой восприятие субъектом объективной реальности в виде данных, переработку этих данных в информацию в соответствии с целевой установкой и имеющимися знаниями о зависимости фактов .

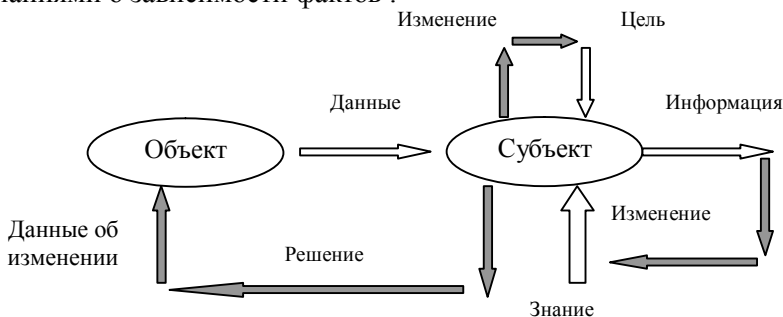


Рис. 1.2. Информационный обмен и процесс

Любая компьютерная информационная система, реализующая информационный процесс, выполняет следующие функции:

1) воспринимает вводимые пользователем информационные запросы и необходимые данные;

2) обрабатывает введенные и хранимые в системе данные в соответствии с известными алгоритмами и формирует требуемую выходную информацию.

3) знания могут субъектов информационного процесса представляются в различной форме:

- а) документированной (учебники, книги, инструкции и т.д.);
- б) недокументированной, неявной (знания конкретного специалиста).

Эти формы представления мало приспособлены для быстрого извлечения при обосновании конкретных решений.

4) знания имеют двоякую природу:

а) фактуальное знание (экстенциональное, детализованное) представляет известные сведения об объектах отражаемой реальности и накапливается в обычных базах данных;

б) операционное знание (интенциональное, обобщенное) отражает зависимость и отношения между объектами, которые позволяют интерпретировать данные или извлекать из них информацию. Операционное знание представляется либо в алгоритмической форме, либо в декларативной (специально сконструированные базы данных).

Информационный процесс с помощью компьютерной информационной системы сводится к адекватному соединению операционного и фактуального знаний и может выполняться по-разному в зависимости от типа информационной системы:

1. В рамках одной прикладной программы:

**Программа = Алгоритм (Правила преобразования данных +
Управляющая структура) + Структура данных.**

В информационных системах данного типа операционное и фактуальное знание неотделимы друг от друга;

2. В системах, основанных на обработке баз данных:

СОБД = Программа ⇔ СУБД ⇔ База данных.

В информационных системах данного типа операционное и фактуальное знание отделены друг от друга. Фактуальное знание

организуется в виде базы данных, операционное — в виде алгоритма программы. В качестве посредника между программой и базой данных выступает программный инструмент доступа к данным — система управления базой данных;

3. В интеллектуальных информационных системах, основанных на обработке баз знаний:

СОЗ = База знаний ⇔ Управляющая структура ⇔ База данных + механизм вывода.

В информационных системах данного типа из операционного знания выделяются правила преобразования данных в отдельную базу — базу знаний. Управляющая структура генерирует алгоритмы решения в зависимости от конкретной поставленной задачи.

4. В интеллектуальных информационных системах, основанных на моделях:

СОМ = Репозиторий ⇔ Механизм генерации или конфигурации ⇔ Информационная система.

В информационных системах данного типа в самостоятельную подсистему (репозиторий метазнания) выделяется описание проблемной области, отражающее ее модель. Программы и структуры данных генерируются из единиц знаний, расположенных в репозитории, и изменяются при модификации модели проблемной области.

1.3. Сети информационного обмена

Компьютерная сеть — это множество компьютеров, соединенных линиями связи и работающих под управлением специального программного обеспечения. Под **линией связи** обычно понимают совокупность технических устройств и физической среды, обеспечивающих передачу сигналов от передатчика к приемнику. В реальной жизни примерами линий связи могут служить участки кабеля и усилители, обеспечивающие передачу сигналов между коммутаторами телефонной сети. На основе линий связи строятся каналы связи. **Каналом связи** обычно называют систему технических устройств и линий связи, обеспечивающую передачу информации между абонентами. Соотношение между понятиями «канал» и «линия» описывается следующим образом: канал связи может включать в себя несколько разнородных линий связи, а одна

линия связи может использоваться несколькими каналами (рис. 1.3).

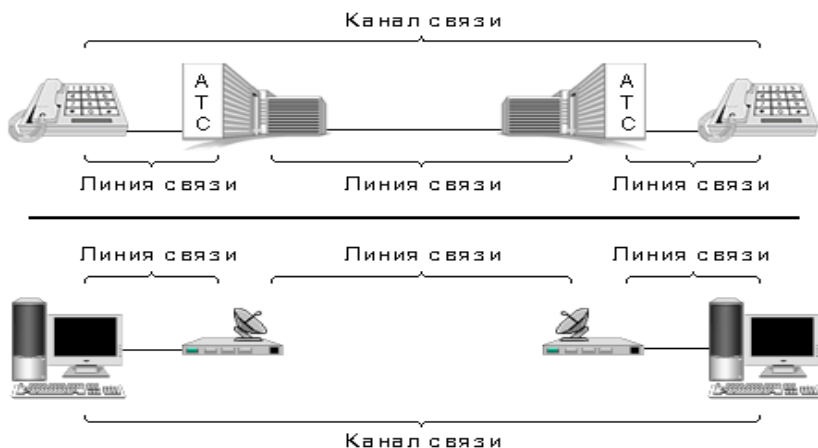


Рис. 1.3. Линии и каналы связи

Главной целью объединения компьютеров в сеть является предоставление пользователям возможности доступа к различным информационным ресурсам (например, к документам, программам, базам данных и т.д.), распределенным по этим компьютерам, а также предоставление им возможности совместного использования этих ресурсов.

Важной характеристикой любой компьютерной сети является широта территории, которую она охватывает. Широта охвата определяется взаимной удаленностью компьютеров, составляющих сеть, и, следовательно, она влияет на технологические решения, выбираемые при построении сети. Выделяют два типа сетей: локальные сети и глобальные сети. К **локальным сетям** обычно относят сети, компьютеры которых сосредоточены на относительно небольших территориях (как правило, в радиусе до одного—двух километров). Классическим примером локальных сетей является сеть одного предприятия, расположенного в одном или нескольких стоящих рядом зданиях. Небольшой размер локальных сетей позволяет использовать для их построения достаточно дорогие, высококачественные технологии, что обеспечивает высокую скорость обмена информацией между компьютерами (рис. 1.4 а).

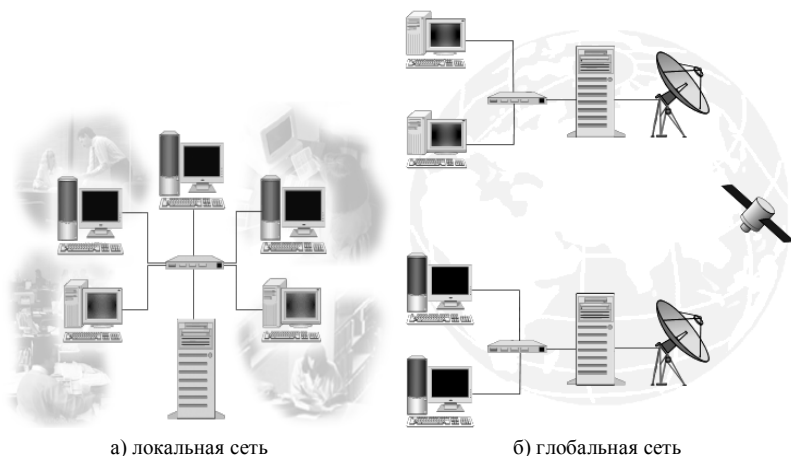


Рис. 1.4. Сети информационного обмена

Глобальные сети — это сети, предназначенные для объединения отдельных компьютеров и локальных сетей, расположенных на значительном удалении (сотни и тысячи километров) друг от друга. Поскольку организация специализированных высококачественных каналов связи большой протяженности является достаточно дорогой, то в глобальных сетях нередко используются уже существующие и изначально не предназначенные для построения компьютерных сетей линии (например, телефонные или телеграфные). В связи с этим скорость передачи данных в таких сетях существенно ниже, чем в локальных (рис. 1.4 б).

Не так давно к двум указанным типам сетей добавился еще один — так называемые **городские сети** (Metropolitan Area Network, MAN). Такие сети предназначены для обеспечения взаимодействия компьютеров и/или локальных сетей, рассредоточенных на территории крупного города (как правило, в радиусе до 100 километров), а также для подключения локальных сетей к глобальным. Для построения таких сетей используются достаточно качественные цифровые линии связи, позволяющие осуществлять взаимодействие на высоких по сравнению с глобальными сетями скоростях (рис. 1.5).



Рис. 1.5. Городская сеть

Независимо от того, какую территорию покрывает сеть, какие технологические решения лежат в основе ее организации, существуют общие принципы сетевого взаимодействия, которым должно подчиняться функционирование сети. Именно выработка таких общих принципов способствовала в свое время появлению Интернета как объединенной сети (иногда даже используется термин «гиперсеть»), собравшей в своем составе локальные, городские и глобальные сети всей планеты (рис. 1.6).

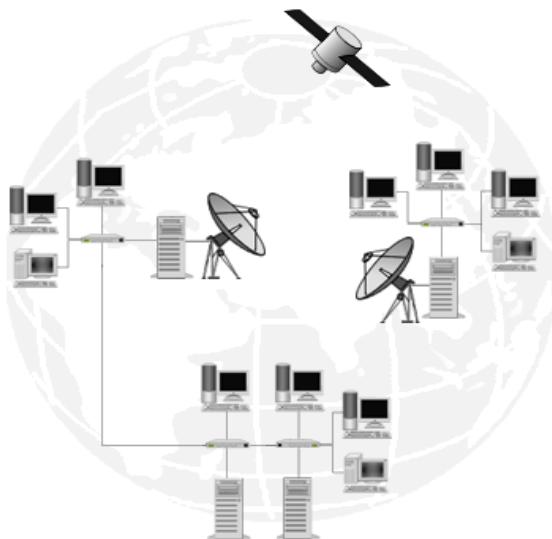


Рис. 1.6. Интернет как объединенная сеть

По характеру реализуемых функций сети делятся на вычислительные (основные функции таких сетей — обработка информации), информационные (для получения справочных данных по запросам пользователей), информационно-вычислительные, или смешанные, в которых в определенном непостоянном соотношении выполняются вычислительные и информационные функции.

По способу управления ИВС делятся на сети с **централизованным** (в сети имеется один или несколько управляющих органов), **децентрализованным** (каждая АС имеет средства для управления сетью) и **смешанным управлением**, в которых в определенном сочетании реализованы принципы централизованного и децентрализованного управления.

По организации передачи информации сети делятся на сети с **селекцией информации** и с **маршрутизацией информации**.

В сетях с **селекцией информации**, строящихся на основе мультимедиа, взаимодействие АС производится выбором (селекцией) адресованных им блоков данных (кадров): всем АС сети доступны все передаваемые в сети кадры, но копию кадра снимают только АС, которым они предназначены.

В сетях с **маршрутизацией информации** для передачи кадров может использоваться несколько маршрутов. Коммуникационные системы сети решают задачи выбора оптимального маршрута.

По типу организации передачи данных **сети с маршрутизацией информации** делятся на:

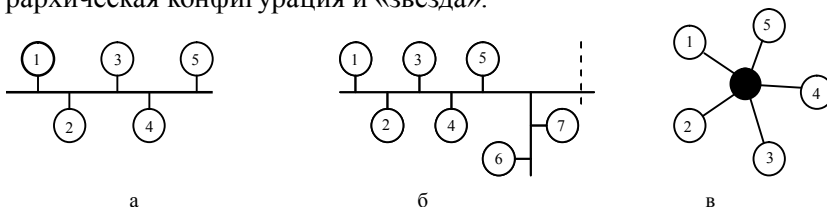
- сети с **коммутацией цепей** (каналов);
- сети с **коммутацией сообщений**;
- сети с **коммутацией пакетов**.

В эксплуатации находятся сети, в которых используются смешанные системы передачи данных.

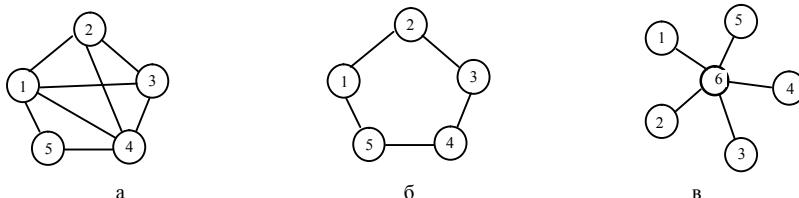
По топологии (конфигурации элементов в ИВС) сети делятся на два класса (рис 1.7): **широковещательные** и **последовательные**.

Широковещательные конфигурации и значительная часть последовательных конфигураций (кольцо, звезда с интеллектуальным центром, иерархическая) характерны для ЛВС. В широковещательных конфигурациях в любой момент времени на передачу кадра может работать только одна рабочая станция (абонентная система). Остальные РС сети могут принимать этот кадр, т.е. такие

конфигурации характерны для ЛВС с селекцией информации. Основные типы широковещательной конфигурации — общая шина, дерево, звезда с пассивным центром. Для глобальных и региональных сетей наиболее распространенной является произвольная (ячеистая) конфигурация. Нашли применение также иерархическая конфигурация и «звезда».



Широковещательные конфигурации сетей:
а) общая шина; б) дерево; в) звезда с пассивным центром.



Последовательные конфигурации сетей:
а) произвольная (ячеистая); б) кольцо; в) звезда с интеллектуальным центром.

Рис. 1.7. Топология вычислительных систем

Контрольные вопросы

1. Дайте определение понятия информации.
2. Что называется данными?
3. Что называется записью?
4. Приведите пример информационного процесса.
5. Приведите пример информационного обмена.
6. Как соединяются операционное и фактуальное знание в рамках прикладной программы?
7. Как соединяются операционное и фактуальное знание в рамках систем, основанных на обработке баз данных?

8. Как соединяются операционное и фактуальное знание в рамках интеллектуальных информационных систем, основанных на обработке баз знаний?

9. Как соединяются операционное и фактуальное знание в рамках интеллектуальных информационных систем, основанных на моделях?

10. Перечислите типы сетей информационного обмена.

11. Укажите отличия локальных, городских и глобальных информационных сетей.

12. Приведите примеры организации передачи информации в сетях.

13. Укажите типы передачи информации в сетях.

14. Приведите примеры топологии информационно-вычислительных сетей.

2. Классификация информационных систем

Информационная система — прикладная программная подсистема, ориентированная на сбор, хранение, поиск и обработку текстовой и/или фактографической информации.

Корпоративной информационной системой называется совокупность специализированного программного обеспечения и вычислительной аппаратной платформы, на которой установлено и настроено программное обеспечение.

Типовые программные компоненты, входящие в состав информационной системы, включают:

- 1) диалоговый ввод—вывод;
- 2) логику диалога;
- 3) прикладную логику обработки данных;
- 4) логику управления данными;
- 5) операции манипулирования файлами или базами данных.

Информационные системы классифицируются по разным признакам:

1. По масштабу (рис. 2.1).



Рис. 2.1. Деление информационных систем по масштабу

Одиночные информационные системы реализуются на автономном ПК.

Содержат несколько простых приложений, связанных общим информационным фондом. Рассчитаны на одного или на группу пользователей, использующих ПК с разделением по времени. Подобные приложения создаются с помощью настольных или локальных систем управления базами данных. Наиболее известны Clarion, Clipper, FoxPro, Paradox, dBase, Microsoft Access.

Групповые информационные системы ориентированы на коллективное использование информации членами рабочей группы и чаще всего строятся на базе локальных вычислительных сетей. При разработке таких приложений используются серверы баз данных (SQL-серверы) для рабочих групп. Среди СУБД наиболее известны Oracle, DB2, Microsoft SQL Server, Inter Base, Informix.

Корпоративные информационные системы являются развитием систем для рабочих групп и могут поддерживать территориально разнесенные узлы или сети. Для них характерна архитектура «клиент/сервер» со специализацией серверов или же многоуровневая архитектура. Наибольшее распространение при разработке корпоративных информационных систем получили Oracle, DB2, Microsoft SQL Server.

2. По сфере применения (рис. 2.2).

Системы обработки транзакций используются для отражения актуального состояния предметной области. Используются в информационных системах организационного управления. Транзакция — единица работы в СУБД. Формируется так, чтобы, начав работу с целостной БД, оставить ее после своего завершения также целостной. Это свойство обеспечивается атомарностью. Единичная операция либо доводится до завершения, либо аннулируется, отменив все начавшиеся действия.

Транзакция — последовательность операций над БД, рассматриваемых СУБД как единое целое. Если операция выполнена успешно, то транзакция также считается успешно выполненной, и СУБД фиксирует (COMMIT) все изменения данных, то есть заносит изменения во внешнюю память. Если хотя бы одна операция заканчивается неудачно, то транзакция считается невыполненной и СУБД производит откат (Roll Back) — отмену всех изменений данных и возврат к первоначальному состоянию.



Рис. 2.2. Деление информационных систем по сфере применения

Транзакция начинается при целостном состоянии базы данных и оставляет это состояние целостным после своего завершения, что позволяет характеризовать транзакцию как *единицу пользовательской активности* по отношению к БД.

Наибольшее значение имеет оперативная обработка транзакций (On Line Transaction Processing), позволяющая отражать предметную область в любой момент времени. Пакетная обработка занимает незначительный объем. В таких системах необходимо выполнять следующие требования:

- высокая производительность обработки транзакций;
- гарантированная доставка информации при удалении доступа к БД по телекоммуникациям.

Для систем OLTP характерен поток довольно простых транзакций, играющих роль заказов, платежей, запросов.

Системы поддержки принятия решений — (Decision Support System) информационные системы, в которых с помощью довольно сложных запросов производится отбор и анализ данных в различных разрезах: временных, географических, стоимостных, технических и т.д.

Информационно-справочные системы в настоящее время основаны на гипертекстовых документах и мультимедиа. Наибольшее развитие получили в Internet.

Офисные информационные системы нацелены на перевод бумажных документов в электронный вид, на автоматизацию делопроизводства и управление документооборотом.

3) По способу организации (рис. 2.3).

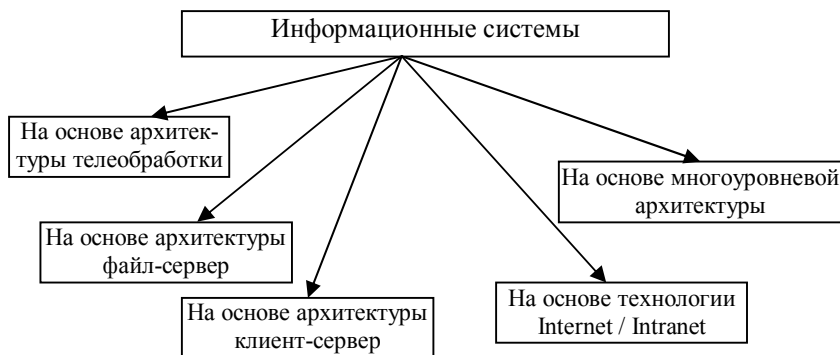


Рис. 2.3. Деление информационных систем по способу организации

Ограничения в применении различных архитектур информационных систем можно понять, выделив и проанализировав их функциональные компоненты (табл. 2.1).

Таблица 2.1

Обозначение	Наименование	Характеристика
PS	Presentation Services (средства представления)	Обеспечивается устройствами, принимающими ввод от пользователя и отображающими то, что сообщает ему компонент логики представления PL с использованием соответствующей программной поддержки.
PL	Presentation logic (логика представления)	Управляет взаимодействием между пользователем и ЭВМ. Обрабатывает действия пользователя при выборе команд в меню, при нажатии кнопок или при выборе элементов из списка.
BL	Business or Application logic (прикладная логика)	Набор правил для принятия решений, вычислений и операций, которое должно выполнять приложение
DL	Data logic (логика управления данными)	Операции с базой данных (SQL-операторы) которые нужно выполнить для реализации прикладной логики управления данными.
DS	Data Services (операции с базой данных)	Действия СУБД, вызываемые для выполнения логики управления данными, определения данных, фиксация или откат транзакций. СУБД обычно компилирует SQL-предложения.
FS	File services (файловые операции)	Дисковые операции чтения и записи данных для СУБД и других компонентов. Обычно являются функциями операционной системы.

Архитектура **телеобработки** (рис. 2.4) использует мощную вычислительную систему с единственным процессором, терминалы неинтеллектуальные и неавтономные.

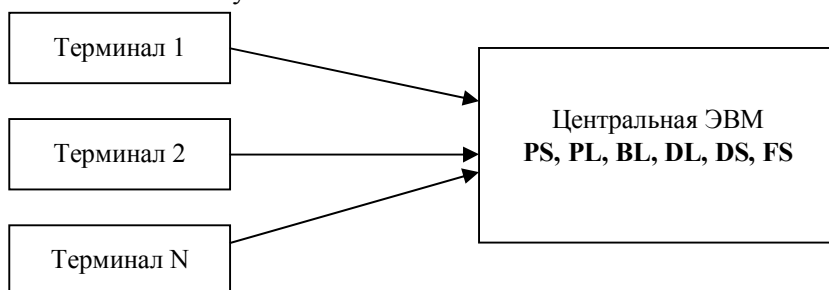


Рис. 2.4. Архитектура телеобработки

Обработку данных осуществляет центральный процессор мейнфрейма. Пользователи при доступе к центральной ЭВМ зачастую дискриминировались системой приоритетов. На ЭВМ возлагалась работа не только по обслуживанию прикладных программ, баз данных, СУБД, но и по обслуживанию терминалов. Все компоненты диалогов и обработки данных располагаются на центральной ЭВМ.

Тенденция к децентрализации и появление высокопроизводительных ПК, локальных вычислительных сетей привели к появлению новых типов архитектуры.

Архитектура **файл-сервер** (рис. 2.5) не имеет сетевого разделения компонентов диалога **PS** и **PL** и использует компьютер для функций отображения, что облегчает построение графического интерфейса. Файл сервер только извлекает данные из файлов. Дополнительные пользователи и приложения добавляют незначительную нагрузку на центральный процессор.

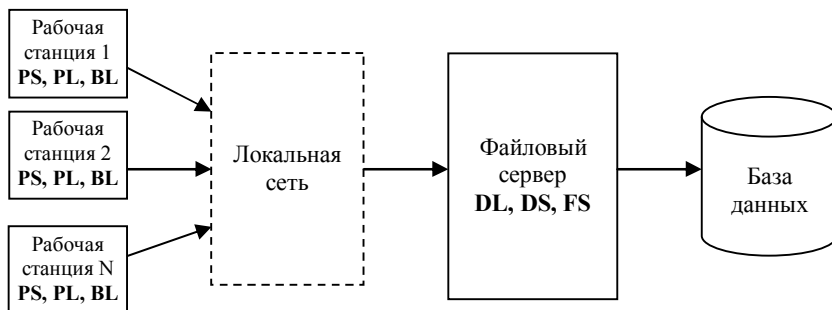


Рис. 2.5. Архитектура файловый сервер

Объектами разработки в файл-серверном приложении являются компоненты приложения, определяющие логику диалога **PL**, а также логику обработки **BL** и управления данными **DL**. Разработанное приложение реализуется либо в виде законченного модуля, либо в виде специального кода для интерпретации. Существенный недостаток такой архитектуры заключается в том, что при выполнении некоторых запросов к базе данных пользователю могут передаваться большие объемы данных, загружая сеть и приводя к непредсказуемому времени реакции.

Устранить указанный недостаток позволяет удаленное управление файл серверным приложением в сети. При этом в локальной

сети размещается сервер приложений, совмещенный с телекоммуникационным сервером (сервер доступа), в среде которого выполняются обычные файл серверные приложения. Особенность состоит в том, что диалоговый ввод—вывод осуществляется через телекоммуникации. Приложение не должно быть слишком сложным.

Архитектура «**клиент/сервер**» предназначена для разрешения проблем файл-серверных приложений путем разделения компонентов приложения и размещения их там, где они будут функционировать наиболее эффективно. Большинство конфигураций клиент/сервер используют двухуровневую модель, в которой клиент обращается к услугам сервера (рис. 2.6).

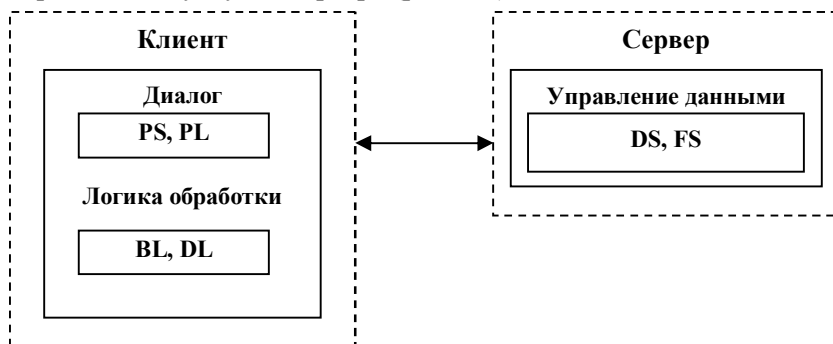


Рис. 2.6. Архитектура «клиент/сервер»

Диалоговые компоненты **PS** и **PL**, логика представлений и прикладная (**BL**, **DL**) размещаются на клиенте, что обеспечивает графический интерфейс. Компоненты управления данными **DS** и **FS** размещаются на сервере.

Клиент управляет пользовательским интерфейсом и логикой приложения, действуя как сложная рабочая станция, на которой выполняются приложения баз данных. Клиент принимает от пользователя запрос, проверяет синтаксис и генерирует запрос к базе данных на языке SQL или на другом языке баз данных. Затем он передает сообщение серверу, ожидает поступление ответа и форматирует полученные данные для представления их пользователю. Сервер принимает и обрабатывает запросы к базе данных, а затем передает полученные результаты клиенту. Выполняемые клиентом и сервером действия приведены в таблице 2.2.

Таблица 2.2

Клиент	Сервер
управляет пользовательским интерфейсом	принимает и обрабатывает запросы к базе данных со стороны клиентов
принимает и проверяет синтаксис введенного пользователем запроса; выполняет приложение	проверяет полномочия пользователей; гарантирует соблюдение ограничений целостности
генерирует запрос к базе данных и передает его серверу	выполняет запросы/обновления и возвращает результаты клиенту
отображает полученные данные пользователю	поддерживает системный каталог; обеспечивает параллельный доступ к базе данных; обеспечивает управление восстановлением

Преимущества такой архитектуры заключаются в следующем:

1. Обеспечивается более широкий доступ к существующим базам данным.

2. Повышается общая производительность системы, т.к. клиент и сервер находятся на разных компьютерах и их процессоры способны выполнять процессы параллельно.

3. Настройка производительности компьютера-сервера упрощается, если на нем выполняется только работа с базой данных.

4. Стоимость аппаратного и программного обеспечения снижается, т.к. только сервер должен быть мощным и на нем должно быть установлено программное обеспечение по обслуживанию базы данных.

5. Сокращаются коммуникационные расходы, т.к. часть операций выполняется на клиентских компьютерах, по сети отправляются запросы только к базе данных.

6. Повышается уровень непротиворечивости данных. Сервер может самостоятельно управлять проверкой целостности данных, поскольку все ограничения определяются и проверяются в одном месте.

Эта архитектура весьма естественно отображается на архитектуру открытых систем.

Такая схема предъявляет наименьшие требования к серверу, однако сложные приложения, вызывающие большое взаимодействие с базой данных могут жестко загружать как клиента, так и сервер. Сократить нагрузку на сеть позволяет размещение компонента BL на сервере. Логика принятия решений оформляется

в виде **хранимых процедур** — процедур с операторами языка SQL для доступа к БД и выполняющихся на сервере БД. (Язык **SQL** — структурированный язык запросов (Structured Query Language)).

2.1. Архитектура «клиент/сервер»

В архитектуре «клиент/сервер» функции приложения распределены между двумя (или более) компьютерами. В соответствии с тем, каким образом это сделано, выделяются три модели архитектуры «клиент/сервер»:

1. Модель доступа к удаленным данным (Remote Data Access — RDA).
2. Модель сервера базы данных (DataBase Server — DBS);
3. Модель сервера приложений (Application Server — AS).

2.1.1. RDA-модель

В RDA-модели (рис. 2.7) коды компонента представления и прикладного компонента совмещены и выполняются на компьютере-клиенте. Последний поддерживает как функции ввода и отображения данных, так и прикладные функции («толстый» клиент).

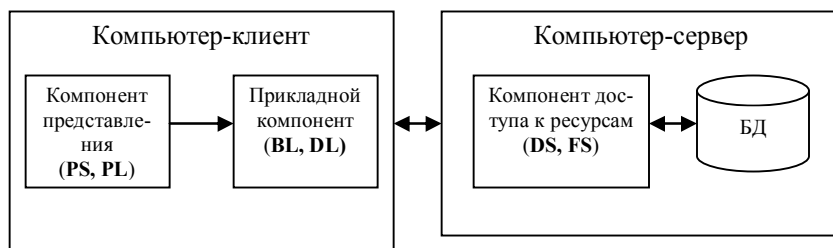


Рис. 2.7. RDA-модель

Доступ к информационным ресурсам обеспечивается, как правило, операторами специального языка (например, SQL) или вызовами функций специальной библиотеки (если имеется соответствующий API). Запросы к информационным ресурсам направляются по сети удаленному компьютеру-серверу базы данных. Последний обрабатывает и выполняет запросы и возвращает клиенту блоки данных. Говоря об архитектуре «клиент/сервер», в большинстве случаев имеют в виду именно эту модель.

Основным преимуществом RDA-модели является широкий выбор средств быстрой разработки приложений различных фирм. Существует множество инструментальных средств, обеспечивающих быстрое создание приложений, работающих с SQL-ориентированными СУБД. Большинство из них поддерживают графический интерфейс пользователя в MS Windows, стандарт интерфейса ODBC, содержат средства автоматической генерации кода. Подавляющее большинство этих средств разработки на языках четвертого поколения (включая и средства автоматизации программирования) как раз и создают коды, в которых смешаны прикладные функции и функции представления.

В то же время RDA-модель имеет ряд ограничений.

1. Очень большая нагрузка сети. Приложение является нераспределенным, и вся его логика локализована на компьютере-клиенте, поэтому взаимодействие его с сервером посредством SQL-запросов приводит к передаче по сети данных большого объема, возможно, избыточных. Как только число клиентов возрастает, сеть становится узким местом, ограничивая быстродействие всей информационной системы.

2. Сложность ведения больших проектов. Очевидно, что если различные по своей природе функции (функции представления и прикладные функции) смешаны в одной и той же программе, написанной на языке SGL, то при необходимости изменения прикладных функций приходится переписывать всю программу целиком. При коллективной работе над проектом, как правило, каждому разработчику поручается реализация отдельных прикладных функций, что делает невозможным контроль за их взаимной непротиворечивостью. Каждому из разработчиков приходится программировать интерфейс с пользователем, что ставит под вопрос единый стиль интерфейса и его целостность.

3. Сложность обновления программного обеспечения, т.к. его замену необходимо производить одновременно на всех компьютерах-клиентах.

4. Низкий уровень безопасности, т.к. реализация разграничения доступа по функциям возможна только на стороне клиента, а на стороне сервера разграничение выполняется только по таблицам базы данных, что снижает уровень защищенности.

2.1.2. DBS-модель

В DBS-модели (рис. 2.8) процесс, выполняемый на компьютере-клиенте, ограничивается функциями представления («тонкий» клиент), а прикладные функции реализованы в хранимых процедурах (stored procedure), которые также называют компилируемыми резидентными процедурами, или процедурами базы данных. Они хранятся непосредственно в базе данных и выполняются на компьютере-сервере базы данных, где функционирует и компонент, управляющий доступом к данным, т.е. ядро СУБД.

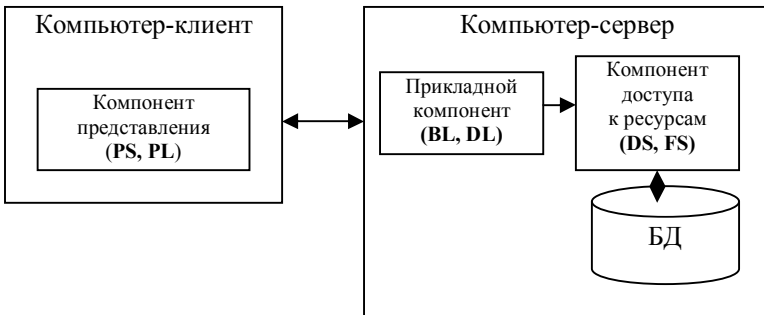


Рис. 2.8. DBS-модель

DBS-модель реализована в некоторых реляционных СУБД (Ingres, Sybase, Oracle). Ее основу составляет механизм хранимых процедур — средство программирования ядра СУБД. Процедуры хранятся в словаре базы данных, разделяются между несколькими клиентами и выполняются на том же компьютере, где функционирует ядро СУБД. Язык, на котором разрабатываются хранимые процедуры, представляет собой процедурное расширение языка запросов SQL.

Преимущества DBS-модели очевидны:

- 1) возможность централизованного администрирования бизнес-функций, размещенных на сервере;
- 2) снижение трафика в сети;
- 3) возможность разделения процедуры между несколькими приложениями и экономия ресурсов компьютера за счет использования единожды созданного плана выполнения процедуры.

Однако есть и недостатки:

1. Средства, используемые для написания хранимых процедур, строго говоря, не являются языками программирования в полном смысле слова. Это разнообразные процедурные расширения SQL, не выдерживающие сравнения по изобразительным средствам и функциональным возможностям с языками третьего поколения (C или Pascal) и тем более четвертого поколения. Они встроены в конкретные СУБД, и, естественно, рамки их использования ограничены. Следовательно, система, в которой прикладной компонент реализован при помощи хранимых процедур, не является мобильной относительно СУБД. В большинстве СУБД отсутствуют возможности отладки и тестирования хранимых процедур, что превращает последние в весьма опасный механизм. Во многих реализациях процедуры являются интерпретируемыми, что замедляет их выполнение.

2. Не обеспечивается требуемая эффективность использования вычислительных ресурсов. Объективные ограничения в ядре СУБД не позволяют пока организовать в его рамках эффективный баланс загрузки, миграцию процедур на другие компьютеры-серверы БД и реализовать другие полезные функции. Попытки разработчиков СУБД предусмотреть в своих системах эти возможности (распределенные хранимые процедуры, запросы с приоритетами и т.д.) пока не позволяют добиться желаемого эффекта.

3. Децентрализация приложений — один из ключевых факторов современных информационных технологий — требует существенного разнообразия вариантов взаимодействия клиента и сервера. При реализации прикладной системы могут понадобиться такие механизмы взаимодействия, как хранимые очереди, асинхронные вызовы и т.д., которые в DBS-модели не поддерживаются.

На практике часто используются смешанные модели, когда поддержка целостности базы данных и некоторые простейшие прикладные функции поддерживаются хранимыми процедурами (DBS-модель), а более сложные функции реализуются непосредственно в прикладной программе, которая выполняется на компьютере-клиенте (RDA-модель).

2.1.3. AS-модель

В AS-модели (рис. 2.9) процесс, выполняющийся на компьютере-клиенте, отвечает, как обычно, за ввод и отображение данных (т.е. реализует функции первой группы). Прикладные функции выполняются группой процессов (серверов приложений), функционирующих на удаленном компьютере (или на нескольких компьютерах). Доступ к информационным ресурсам, необходимым для решения прикладных задач, обеспечивается таким же способом, что и в RDA-модели. Серверы приложений выполняются, как правило, на том же компьютере, где функционирует менеджер ресурсов, однако могут выполняться и на других компьютерах.

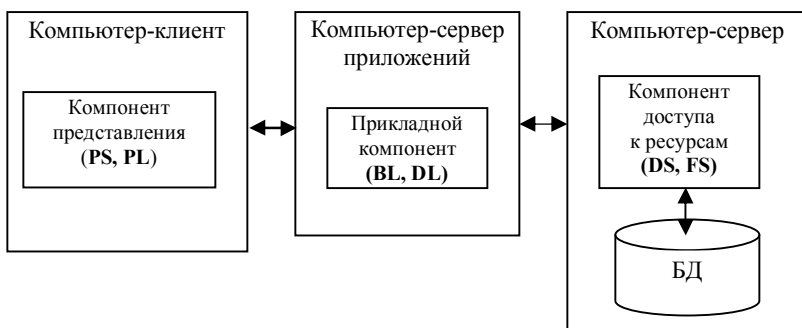


Рис. 2.9. AS-модель

Основным элементом принятой в AS-модели трехзвенной схемы является сервер приложения. В его рамках реализовано несколько прикладных функций, каждая из которых оформлена как служба (service) и предоставляет некоторые услуги всем программам, которые могут ими воспользоваться. Серверов приложений может быть несколько, и каждый из них предоставляет определенный набор услуг. Любая программа, которая пользуется ими, рассматривается как клиент приложения (Application Client — AC). Детали реализации прикладных функций в сервере приложений полностью скрыты от клиента приложения. AC обращается с запросом к конкретной службе, но не к AS, т.е. серверы приложений обезличены и служат лишь своего рода «рамкой» для оформления служб, что позволяет эффективно управлять балансом загрузки. Запросы, поступающие от AC, выстраиваются

в очередь к AS-процессу, который извлекает и передает их для обработки службе в соответствии с приоритетами.

АС трактуется более широко, чем компонент представления. Он может поддерживать интерфейс с конечным пользователем (тогда он является компонентом представления), может обеспечивать поступление данных от некоторых устройств (например, датчиков), может, наконец, сам по себе быть AS. Последнее позволяет реализовать прикладную систему, содержащую AS нескольких уровней. Архитектура такой системы может выглядеть как ядро, окруженное концентрическими кольцами. Ядро состоит из серверов приложений, в которых реализованы базовые прикладные функции. Кольца символизируют наборы AS, являющихся клиентами по отношению к серверам нижнего уровня. Число уровней серверов в AS-модели, вообще говоря, не ограничено.

AS-модель в наибольшей степени отражает сильные стороны технологии «клиент/сервер»:

- 1) четкое разграничение логических компонентов приложения;
- 2) возможность баланса загрузки между несколькими серверами;
- 3) значительное снижение трафика между клиентом и сервером приложений, дающее возможность работы по медленным линиям связи;
- 4) высокий уровень защиты данных, т.к. они являются «спрятанными» за сервисами приложения, в которые можно встроить проверку полномочий клиента;
- 5) возможность использования в качестве клиентской части приложения стандартного браузера;
- 6) упрощение процесса обновления ПО.

Фундаментальное различие между моделями архитектуры «клиент/сервер» заключается в следующем. RDA- и DBS-модели опираются на двухзвенную схему разделения функций. В RDA-моделях прикладные функции приданы программе-клиенту, в DBS-моделях ответственность за их выполнение берет на себя ядро СУБД. В первом случае прикладной компонент сливается с компонентом представления, во втором — интегрируется в компонент доступа к информационным ресурсам. Напротив, в AS-модели реализована классическая трехзвенная схема разделения функций, где прикладной компонент выделен как важнейший

элемент приложения, для его определения используются универсальные механизмы многозадачной операционной системы, и стандартизованы интерфейсы с двумя другими компонентами. Собственно, из этой особенности AS-модели и вытекают ее преимущества.

Многоуровневая архитектура — развитие архитектуры «клиент/сервер». В своей классической форме она состоит из трех уровней (рис. 2.10):

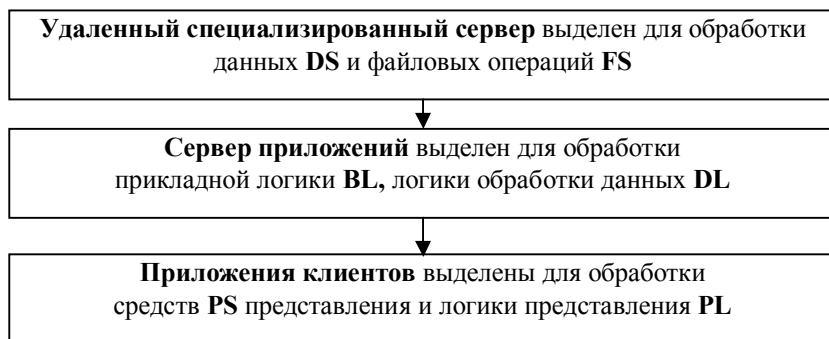


Рис. 2.10. Многоуровневая архитектура

Трехуровневая архитектура позволяет сбалансировать нагрузку на узлы и сеть. Четко разделяются платформы и инструменты для реализации интерфейса и прикладной логики. Оптимизируется распределение программно-аппаратных ресурсов. Средства удаленного вызова процедур в наибольшей степени соответствуют идее распределенных вычислений, обеспечивая:

- 1) вызов прикладной процедуры из любого узла сети;
- 2) передачу параметров;
- 3) удаленную обработку и возврат данных.

Интернет/Инtranет-технологии в сочетании с многоуровневой архитектурой позволяют создать простые и удобные информационные системы. Структура приложения приобретает следующий вид (рис. 2.11):

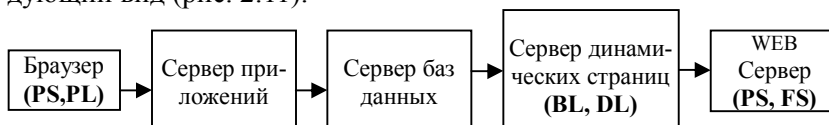


Рис. 2.11. Архитектура на основе Интернет/Инtranет-технологии

2.2. Области применения и примеры реализации информационных систем

Современный бизнес крайне чувствителен к ошибкам в управлении, и для принятия грамотного управленческого решения в условиях неопределенности и риска необходимо постоянно держать под контролем различные аспекты финансово-хозяйственной деятельности лечебно-профилактического учреждения. Рассмотрим наиболее важные задачи, решаемые с помощью информационных систем.

Бухгалтерский учет — наиболее часто реализуемая на сегодняшний день задача. Бухгалтерские ошибки стоят дорого, а задача бухучета наиболее просто реализуема, т.к. легко формализуется.

Управление финансовыми потоками — внедрение информационных технологий также обусловлено критичностью этой области к ошибкам. Неправильно построив систему расчетов с поставщиками и потребителями, можно спровоцировать кризис наличности даже при налаженной сети закупки, сбыта и маркетинга (аптечная сеть).

Управление складом, ассортиментом, закупками — внедрение информационных технологий в эту область позволяет избежать замораживания оборотных средств в чрезмерном складском запасе, выявить перспективные товары (медицинское оборудование, лекарственные препараты).

Управление производственным процессом — трудоемкая задача, включающая планирование и оптимальное управление производственным процессом. Решение этой задачи дает возможность грамотно планировать, учитывать затраты, проводить техническую подготовку производства, оперативно управлять процессом выпуска продукции в соответствии с производственной программой и технологией. В области медицины это полная автоматизация бизнес-процессов и формирование единого информационного поля здравоохранения на государственном и территориальном уровнях, страховая медицина. Информационные системы управляют бизнес-процессами, включая планирование, взаимоотношения с контрагентами, организацию логистики, финансовый анализ, управление персоналом и т.д.

Управление маркетингом предполагает сбор и анализ данных о фирмах, их продукции и ценовой политике, прогнозирование рекламных кампаний, моделирования внешнего окружения для определения оптимального уровня цен (лекарственные препараты, медицинское оборудование).

Документооборот является важным процессом деятельности любого предприятия. Отлаженная система учета документооборота отражает реальную текущую производственную деятельность предприятия. В области медицины это электронные истории болезни, электронные базы данных, справочники.

Оперативное управление предприятием — это корпоративная информационная система, решающая задачи текущего управления лечебно-профилактическим учреждением. Содержит большое число баз данных, фиксирующих всю возможную информацию о предприятии. Содержимое баз данных непрерывно пополняется и модифицируется.

Представление информации о фирме — каждое уважающее себя предприятие имеет свой Web-сервер, который решает следующие задачи:

- создание имиджа медицинского учреждения;
- максимальная разгрузка справочной службы путем представления необходимой информации о фирме, товарах, услугах, ценах и т.д.

Контрольные вопросы

1. Дайте определение понятия «корпоративная информационная система».
2. Укажите существенное отличие корпоративной информационной системы от других типов информационных систем.
3. Дайте определение понятия «транзакция».
4. Приведите пример классификации информационных систем по сфере применения.
5. Чем отличается Internet от Intranet?
6. Перечислите функциональные компоненты информационной системы.
7. Приведите пример архитектуры телеобработки.

8. Поясните архитектуру информационной системы на основе файл-сервера.
9. Поясните архитектуру информационной системы на основе «клиент/сервер».
10. Перечислите модели архитектуры «клиент/сервер».
11. Поясните особенности архитектуры «клиент/сервер» на основе RDA-модели.
12. Поясните особенности архитектуры «клиент-сервер» на основе DBS-модели.
13. Поясните особенности архитектуры «клиент-сервер» на основе AS-модели.
14. Поясните особенности многоуровневой архитектуры.
15. Приведите примеры и назовите области применения информационных систем в медицине.

3. Жизненный цикл информационных систем

3.1. Классификация проектов информационных систем

Информационные системы в основном развиваются как проект. Проекты информационных систем могут сильно отличаться по сфере приложения, составу, предметной области, масштабу, длительности, составу участников, степени сложности, значимости результатов.

В общем случае проекты классифицируются (рис. 3.1) по следующим основным признакам:

1) класс:

— **монопроект** (отдельный проект, который может отличаться масштабом, любым типом, значимостью);

— **мультипроект** (комплексный проект, состоящий из ряда монопроектов и требующий многопроектного управления);

— **мегапроект** — целевые программы развития регионов, отраслей и других образований, включающие в свой состав ряд моно- и мультипроектов.

2) тип:

— технические;

— организационные;

— экономические;

— социальные;

— смешанные.

3) характер предметной области:

— учебно-образовательные проекты;

— проекты исследования и развития;

— инновационные проекты (цель которых — разработка и применение инноваций и т.п.);

— инвестиционные проекты, (цель которых — создание и реновации основных фондов, требующих вложения инвестиций);

— комбинированные проекты, сочетающие в себе комбинацию различных видов проектов.

4) длительность:

— краткосрочные (до 3 лет);

— среднесрочные (от 3 до 5 лет);

— долгосрочные (свыше 5 лет).

- 5) сложность:
- простые;
 - сложные;
 - очень сложные.
- 6) масштаб:
- мелкие;
 - малые;
 - средние;
 - крупные;
 - очень крупные.

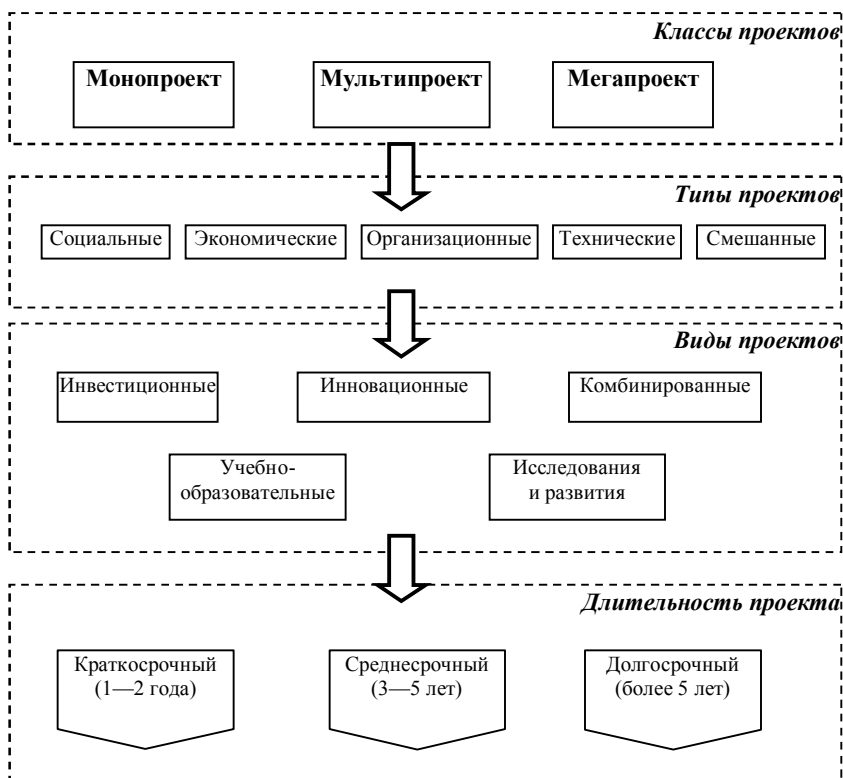


Рис. 3.1. Классификация проектов

Дополнительно следует различать терминальные, развивающиеся и открытые проекты.

Терминальный (конечный) проект имеет конечную цель и четко ограниченный жизненный цикл.

Развивающийся проект на момент начала (инициации) не имеет четких конечных целей и ограниченного жизненного цикла. Конечные цели проекта возникают в зависимости от эффективности ранее осуществлявшейся деятельности и внешних условий. Развивающийся проект завершится тогда, когда исчерпается набор гипотез и концепций, заложенный в проект при его инициации (разработка версий программного продукта).

Открытый проект не имеет четко заданных и неизменных целей, и потенциально отсутствует его временное ограничение. Целевые показатели управляемой системы меняются в силу ее саморазвития и внутренних тенденций, а также в силу внешних случайных факторов (проекты социально-экономического развития региона).

3.2. Разработка, эксплуатация

Разработка информационной системы — это в основном технический проект. Каждый проект, независимо от сложности, проходит в своем развитии от состояния «проекта еще нет» до состояния «проекта уже нет». Можно выделить следующие фазы развития информационной системы:

- 1) формирование концепции;
- 2) разработка технического задания;
- 3) проектирование;
- 4) изготовление;
- 5) ввод системы в эксплуатацию.

Концептуальная фаза включает:

- формирование идей и постановку целей;
- формирование ключевой команды проекта (группы основных исполнителей, включающей руководителя, главного инженера, главного специалиста и т.д.);
- изучение мотивации и требований заказчика и других участников;
- сбор исходных данных и анализ существующего состояния;
- определение основных требований и ограничений, требуемых материальных, финансовых и трудовых ресурсов;

- сравнительную оценку альтернатив;
- представление предложений, их экспертизу и утверждение.

Техническое предложение включает:

- разработку основного содержания проекта, базовой структуры;
- разработку технического задания;
- планирование, декомпозицию структурной базовой модели проекта;
- составление сметы и бюджета проекта, определение потребности в ресурсах;
- разработку календарных планов и укрупненных графиков работ (сетевых графиков);
- подписание контракта с заказчиком;
- ввод в действие средств коммуникации и мер контроля за ходом работ для участников проекта.

Проектирование — на этой фазе осуществляется:

- выполнение базовых проектных работ;
- разработка частных технических заданий;
- выполнение концептуального проектирования;
- составление технических спецификаций и инструкций;
- представление проектной разработки, ее экспертиза и утверждение.

Изготовление — на этой фазе выполняется:

- разработка программного обеспечения;
- подготовка к внедрению системы;
- контроль и регулирование основных показателей проекта.

Ввод системы в эксплуатацию включает в себя следующие работы:

- 1) комплексные испытания;
- 2) подготовку кадров для эксплуатации создаваемой системы;
- 3) подготовку рабочей документации, сдачу системы заказчику и ввод ее в эксплуатацию;
- 4) сопровождение, поддержку, сервисное обслуживание;
- 5) оценку результатов проекта и подготовку итоговых документов;
- 6) разрешение конфликтных ситуаций и закрытие работ по проекту;

7) накопление опытных данных по эксплуатации системы, исправление ошибок, анализ опыта и определение направления развития.

Наиболее часто встречаются следующие ошибки в проектах информационных систем:

- ошибки в определении интересов заказчика;
- концентрация на маловажных вторичных интересах;
- неправильная интерпретация исходной постановки задачи;
- неправильное или недостаточное понимание деталей;
- неполнота функциональных спецификаций (системных требований);
- ошибки в определении требуемых ресурсов и сроков;
- редкая проверка на согласованность этапов и отсутствие контроля со стороны заказчика.

3.3. Основные процессы жизненного цикла

Жизненный цикл информационной системы регламентируется стандартом ISO/IEC 12207 (ISO — International standardization organization, IEC — International electrotechnical commission). Согласно этому стандарту структура жизненного цикла основывается на трех группах процессов:

- 1) основные процессы (приобретение поставка, разработка, эксплуатация, сопровождение);
- 2) вспомогательные процессы, сопровождающие основные (документирование, управление конфигурацией, верификация, аттестация и т.д.);
- 3) организационные процессы (управление проектами, создание инфраструктуры проекта, улучшение жизненного цикла, обучение).

Рассмотрим основные процессы жизненного цикла.

Разработка включает в себя:

- 1) оформление проектной документации;
- 2) подготовку материалов, необходимых для проведения тестирования разработанных программных продуктов;
- 3) разработку дидактических материалов для обучения персонала заказчика.

Эксплуатация — на этом этапе работы делятся на подготовительные и основные. *К подготовительным работам* относятся:

- конфигурирование рабочих мест пользователей и базы данных;
- обеспечение пользователей эксплуатационной документацией;
- обучение персонала.

Основные эксплуатационные работы включают:

- непосредственно эксплуатацию;
- локализацию проблем и устранение причин их возникновения;
- модификацию программного обеспечения;
- подготовку предложений по совершенствованию системы;
- развитие и модернизацию системы.

Сопровождение обеспечивается службой технической поддержки, укомплектованной наиболее квалифицированными специалистами, способными решать не только каждодневные задачи администрирования, но и быстро восстанавливать работоспособность системы после сбоев.

Техническое обслуживание и сопровождение включает:

1. Выделение наиболее ответственных узлов системы и определение для них критичности простоя. Это позволяет выделить и оптимизировать распределение ресурсов для технического обслуживания.
2. Разделение задач технического обслуживания на внешние и внутренние. Внешние передаются специализированным сервисным организациям, внутренние решаются силами обслуживающего персонала.
3. Проведение анализа внутренних и внешних ресурсов, необходимых для организации технического обслуживания (играет роль гарантии квалификация персонала, состояние ремонтного фонда).
4. Подготовку плана организации технического обслуживания с указанием действий, сроков исполнения.

Вспомогательные процессы — наибольшее значение имеет управление конфигурацией. В сложных многокомпонентных системах отдельные модули программного обеспечения могут разрабатываться независимо друг от друга, и возникает необходимость учета их связей и функций в единой информационной системе.

Организационные процессы разделяются на технические и организационные. Они включают:

- 1) выбор методов и инструментальных средств реализации проекта;
- 2) определение методов описания промежуточных состояний проекта;
- 3) разработку методов и средств испытаний созданного программного обеспечения;
- 4) обучение персонала.

Качество проекта связано с решением вопросов верификации и проверки.

Верификация — процесс определения соответствия текущего состояния системы требованиям данного этапа.

Проверка — это процесс определения соответствия параметров разработки исходным требованиям. Проверка отчасти совпадает с тестированием, которое производится для определения различий между действительными и ожидаемыми результатами. Кроме того, это оценка соответствия характеристик информационной системы исходным требованиям.

3.4. Структура жизненного цикла

Жизненный цикл информационной системы можно разделить на четыре стадии:

- 1) начало;
- 2) уточнение;
- 3) конструирование;
- 4) переход (передача в эксплуатацию).

На **начальной стадии** устанавливается область применения системы и граничные условия. На начальной стадии идентифицируются все функциональные возможности системы и производится описание наиболее существенных из них.

На **стадии уточнения** анализируется прикладная область, разрабатывается архитектурная основа информационной системы. В конце стадии проводится повторный анализ решений информационной системы, определяются способы устранения главных элементов риска, содержащихся в проекте.

На **стадии конструирования** разрабатывается законченное изделие, определяется работоспособность разработанного программного обеспечения.

На **стадии перехода** программное обеспечение передается пользователям. Обнаруженные ошибки и недоработки устраняются. В конце этапа определяют, достигнута цель разработки или нет.

3.5. Каскадная и спиральная модели жизненного цикла

Моделью жизненного цикла информационной системы будем называть некоторую структуру, определяющую последовательность осуществления процессов, действий и задач, выполняемых на протяжении жизненного цикла информационной системы, а также взаимосвязь между этими процессами, действиями и задачами. Наибольшее распространение имеют каскадная и спиральная модели.

Каскадная модель применяется во всех сферах человеческой деятельности, описана в нормативной, технической, экономической и методической литературе. Каскадная модель подразумевает последовательную организацию работ: вся работа разбивается на этапы: переход с этапа на этап происходит только после того, как завершены все работы предыдущего этапа. Каждый этап завершается выпуском полного комплекта документации, достаточной для того, чтобы другая команда разработчиков могла продолжить работу над новым этапом с «нуля». Вне зависимости от предметной области можно выделить ряд устойчивых этапов разработки, практически не зависящих от предметной области (рис. 3.2).

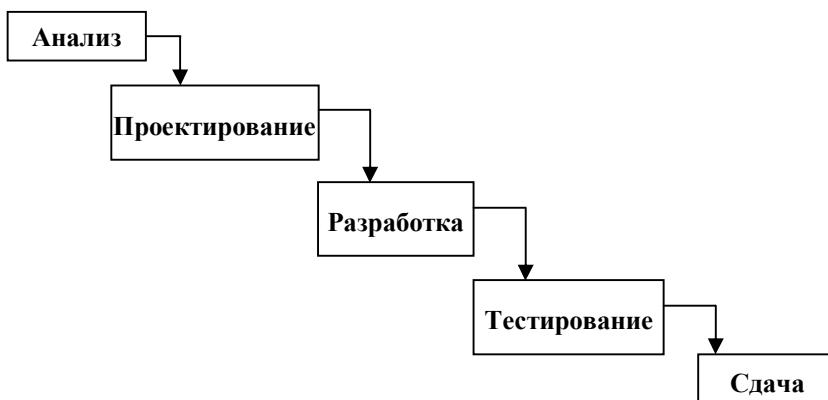


Рис. 3.2. Каскадная модель разработки

На этапе **анализа** производится изучение требований заказчика, исследуются проблемы и четко формулируются задачи, которые должны быть решены. Результатом этих работ является техническое задание, которое согласуется со всеми заинтересованными сторонами.

На этапе **проектирования** разрабатываются проектные решения, удовлетворяющие всем требованиям, сформулированным в техническом задании. Результатом этих работ является комплект проектной документации, содержащий все данные, необходимые для реализации проекта.

На этапе **разработки** создается программное обеспечение в соответствии с проектной документацией. Методы, используемые для реализации, принципиального значения не имеют. Результатом этих работ является готовый программный продукт.

На этапе **тестирования** производится проверка полученного программного обеспечения на предмет соответствия требованиям, заявленным в техническом задании. Опытная эксплуатация позволяет выявить скрытые дефекты и недоработки, проявляющиеся при реальной эксплуатации любой системы.

На этапе **сдачи** необходимо убедить заказчика, что его требования удовлетворены в полном объеме.

Каскадная модель имеет ряд существенных достоинств:

1. На каждом этапе имеется полный комплект документации, отвечающий критериям полноты и согласованности.
2. Работы выполняются в логической последовательности, что позволяет планировать сроки завершения.
3. Хорошо зарекомендовала себя при создании инженерно-технических и некоторого класса информационных систем (реального времени, расчетные системы).

При разработке информационных систем других типов проявляются недостатки каскадной модели:

1. Задержка получения результатов возникает из-за последовательности выполнения работ. Только на конечном этапе можно выяснить, удовлетворяет ли информационная система требованиям пользователя или нет. Искажения могут вноситься как аналитиками-проектировщиками, так и программистами из-за незнания в совершенстве предметной области.

2. Ошибки и недоработки, возникшие на этом этапе, проявляются, как правило, на следующем, что приводит к необходимости возврата на предыдущие стадии.

3. Сложность параллельного ведения работ, даже если разработку подсистем можно вести параллельно. Это связано с необходимостью постоянного согласования различных частей проекта.

4. Информационная перенасыщенность возникает вследствие сильной зависимости между различными группами разработчиков. Внесение изменений в проект или в его часть требует оповещения всех разработчиков.

Реальный процесс разработки по каскадной модели представлен на (рис. 3.3).

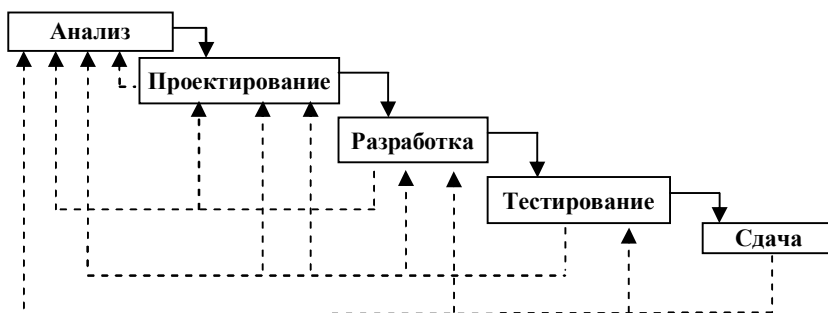


Рис. 3.3. Реальный процесс разработки по каскадной модели

Спиральная модель предполагает итерационный процесс разработки информационной системы; при этом возрастает значение начальных этапов жизненного цикла, таких как анализ и проектирование. На этих этапах проверяется и обосновывается реализуемость технических решений путем создания прототипов. Каждая итерация представляет собой законченный цикл разработки, приводящий к выпуску внутренней или внешней версии изделия, которое совершенствуется от итерации к итерации (рис. 3.4).

Каждый виток спирали соответствует созданию фрагмента или версии программного продукта, переход на следующий этап можно осуществить, не дожидаясь полного завершения, т.к. недоделки можно устранить на следующей итерации. Главная задача итерации — создать готовый продукт как можно быстрее, для его демонстрации заказчику.

Итерационный подход имеет следующие преимущества:

1) упрощает внесение изменений и дополнений при изменении требований заказчика;

2) отдельные элементы интегрируются в систему постепенно практически на каждом этапе;

уменьшается риск и обеспечивается большая гибкость в управлении проектом;

процесс устранения ошибок или недочетов упрощается, т.к. выполняется непрерывно на каждом этапе (итерации).

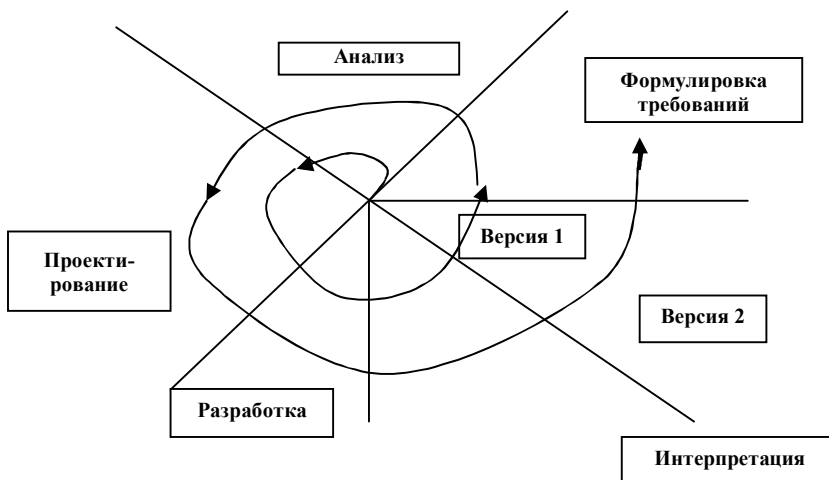


Рис. 3.4. Спиральная модель жизненного цикла информационной системы

Существенный недостаток такой модели — невозможность точного определения момента окончания итерации. Стремление к совершенству порождает непрерывность итераций. Здесь требуется помнить, что лучшее — враг хорошего.

Итерационные модели практически не имеют смысла в технической и промышленной сферах (трудно представить плотину, здание, возведение которых производится таким образом).

Контрольные вопросы

1. Приведите классификацию информационных систем как проектов.
2. Перечислите фазы развития информационной системы.

3. Что включает концептуальная фаза развития информационной системы?
4. Что включает фаза технического предложения?
5. Что включает фаза проектирования?
6. Что включает фаза изготовления?
7. Что включает фаза ввода системы в эксплуатацию?
8. Какими стандартами регламентируется жизненный цикл информационной системы?
9. Из каких групп процессов состоит жизненный цикл информационной системы?
10. Перечислите основные процессы жизненного цикла информационной системы.
11. Какие процессы жизненного цикла информационной системы относятся к вспомогательным?
12. Дайте определение верификации и проверки.
13. Поясните каскадную модель жизненного цикла информационной системы.
14. Поясните спиральную модель жизненного цикла информационной системы.

4. Информационные системы документального поиска

На современном этапе развития здравоохранения России появились врачи общей практики, медицинские знания которых должны быть обширны и должны соответствовать современному состоянию медицинской науки. Это возможно только с применением Интернет-медицины и телемедицины.

4.1. Телемедицина

Термин «телемедицина» был введен в медицинскую литературу в 1974 г. В настоящее время можно найти множество ее определений. Приведем некоторые из них.

Телемедицина — это быстрое обеспечение медицинскими знаниями на расстоянии с помощью телесвязи и информационной технологии независимо от того, где находится пациент или где требуется нужная информация.

Телемедицина — это совокупность внедряемых, «встраиваемых» в медицинские информационные системы принципиально новых средств и методов обработки данных, объединяемых в целостные технологические системы, обеспечивающие создание, передачу, хранение и отображение информационного продукта (данных, знаний) с наименьшими затратами с целью проведения необходимых и достаточных лечебно-диагностических мероприятий, а также обучения для всех нуждающихся в них в нужном месте и в нужное время.

Внедрение информационных технологий для нужд телемедицины — это, пожалуй, основное направление в работе прикладных математиков, специалистов по медицинской информатике. Даже в нашей стране телемедицина получила значительное развитие. На сегодня основными областями ее применения являются:

- диагностика и обеспечение консультаций на расстоянии (при этом в основном между специалистом и другим медиком (врачом или медсестрой));
- дистанционное обучение.

Понятно, что применение телемедицинских технологий особенно эффективно там, где диагностика основана на визуальной

информации. К таким специальностям можно отнести телеэндоскопию, теледерматологию, телерадиологию, телеофтальмологию, ультразвуковую диагностику, телепатологию. Последняя составляет около 11% от всех телемедицинских проектов, причем 58% из них носят междисциплинарный характер.

Поддержка медицинской информационной системой телемедицинских возможностей — обоснованная необходимость. Сама по себе телемедицина вряд ли способна качественно улучшить лечебно-диагностический процесс, т.к. востребована только в определенных ситуациях. Ее можно и нужно встраивать в информационную систему, но не следует считать телемедицину основой информатизации лечебного учреждения.

Отметим основные преимущества от внедрения телемедицинских функций в информационные системы:

- снижение стоимости лечения или диагностики; сокращается необходимость в поездках на консультации и обследования, в покупке дорогостоящего оборудования и расходных материалов;
- расширение информационных связей между медицинскими работниками; проведение виртуальных консилиумов, видеоконференций — общение через Интернет расширяет информационные возможности каждого сотрудника;
- улучшение качества медицинской помощи, расширение возможностей ее оказания. Так, пациент, имеющий домашний компьютер с подключением к Интернет может отправить электронное сообщение своему лечащему врачу или семейному доктору, записаться на прием, получить консультацию или результаты обследования.

Выделим основные телемедицинские задачи информационных систем:

1. Возможность проведения консультаций, которые могут быть двух видов — отложенные и в режиме реального времени.

Отложенная консультация — это процесс обмена информацией, когда пациент записывается на нее, при необходимости формируется электронное описание его жалоб, анамнеза заболевания, проведенного лечения и обследования. Вся эта информация переносится в специальную базу данных удаленного консультанта. Время от времени консультант читает сформированный для него список записей, выполняет консультации.

При консультации в режиме реального времени консультанту по системе обмена сообщений (например, по электронной почте) отправляются необходимые данные о пациенте. Как только он получает их, немедленно отвечает на запрос. Такие возможности достаточно просто выполнить. Для этого в программу планирования рабочего времени необходимо добавить возможности работы через Интернет, записи пациента с указанием анамнеза и других данных. В подсистемы врачей необходимо встроить поддержку отправки документов по электронной почте.

2. Консилиумы и видеоконференции. В рамках информационных систем для этих целей должно применяться специальное программное обеспечение, т.к. существует реальная необходимость организации связи между врачами (с возможным участием пациента) и передачей видеоинформации. Примером может служить программа Microsoft NetMeeting, входящая в комплект поставки операционной системы Windows.

3. Дистанционное обучение. Использование ресурсов сети Интернет, таких как электронные библиотеки полнотекстовых статей, доступ к архивам интересных клинических случаев, записей редких видов патологии и т.д.

4.2. Интернет и информационно-поисковые системы

Уверенное применение Интернет невозможно без общего представления об *информационно-поисковых системах*, которые с течением времени также начнут включать аналитические блоки, что позволит перейти к индивидуально-групповому управлению процессом профилактики и лечения на основе использования накапливаемых в их базах данных сведений о диспансеризуемых контингентах.

Информационно-поисковые системы — это упорядоченная совокупность документов (массивов документов) и информационных технологий, предназначенных для хранения и поиска информации — текстов (документов) или данных (фактов). Информационно-поисковыми системами (далее — ИПС) являются любые определенным образом организованные хранилища информации. Причем ИПС могут быть и неавтоматизированными. Главное — это целевая функция: хранение и поиск информации.

В зависимости от объекта хранения и типа запроса различают три вида информационного поиска: документальный, фактографический и информационно-логический.

Документальными называются ИПС, в которых реализуется поиск по запросам в массиве документов или текстов с последующим представлением пользователю подмножества всех документов (текстов), которые удовлетворяют его запросу. На запрос «скорость света» будут выданы все документы, в которых имеется словосочетание «скорость света».

Фактографические ИПС реализуют хранение, поиск и выдачу фактических данных (научные, технические, экономические и т.д.), и на запрос «скорость света» будет выдано численное значение скорости (300 000 км/сек).

Таким образом, документальные и фактографические ИПС отличаются степенью предварительной интеллектуальной обработки материала.

В информационно-логических ИПС возможны ответы на запросы, для которых в базах данных нет ответа в явном виде. Для этого используется экстралингвистическая база знаний, порождающая ответ на запрос алгоритмическим образом из имеющейся информации. Ответ может использоваться непосредственно либо для дальнейшего поиска. Системы такого типа разрабатываются давно, но практического успеха пока не имеют.

4.3. Содержание документального поиска.

Релевантность и пертинентность

Важное значение в ИПС имеют **запросы**, представляющие собой информационную потребность, сформулированную на естественном языке. В результате перевода информационного запроса на информационно-поисковый язык возникает поисковый образ запроса (далее — ПОЗ), или поисковое предписание. Для составления адекватного ПОЗ пользователь должен знать язык запросов и информационно-поисковый язык в минимальном объеме для интерпретации действий поисковой системы.

Разновидностью документального поиска является предметный (или тематический) поиск. Документ, центральная тема которого в целом соответствует смысловому содержанию информационного запроса, называется релевантным. В ИПС формируются

формализованные представления содержания документов и запросов. Эти формализованные представления принято обозначать терминами «поисковый образ документа» (ПОД) и, соответственно, «поисковый образ запроса» (ПОЗ). Часто используется термин «поисковое предписание».

В соответствии с этими терминами различают:

- 1) смысловую релевантность (документ соответствует содержанию информационного запроса);
- 2) формальную релевантность (поисковый образ документа соответствует поисковому образу запроса);
- 3) истинную релевантность (документ соответствует информационной потребности пользователя; ее часто называют пертинентностью).

Информационная потребность возникает у людей и порождает запрос на поиск информации, однако далеко не все пользователи могут правильно сформулировать и точно выразить свои потребности (например: запрос «информационные системы в медицине» порождает массу ответов о фирмах-разработчиках, о существующем программном обеспечении и т.д., хотя пользователь хотел получить учебные планы по дисциплине «Информационные системы в медицине»).

Каждый вид релевантности может рассматриваться как точка в четырехмерном пространстве:

- 1) поисковый образ, документ, информация;
- 2) поисковое предписание, пользовательский запрос, информационная потребность, проблема;
- 3) пользователь, тема, задача, контекст и их всевозможные комбинации;
- 4) различные временные интервалы: от возникновения проблемы до ее решения.

4.4. Функциональные операционные схемы информационно-поисковых систем

ИПС делятся на составные части (подсистемы) по функциональному признаку, когда каждая подсистема выполняет определенную функцию в технологическом процессе (рис. 4.1):

- 1) ввод документов;

- 2) индексирование документов;
- 3) ввод и корректировка запросов;
- 4) индексирование запросов;
- 5) поиск;
- 6) ведение словарей;
- 7) ведение статистики;
- 8) обработка результатов поиска;
- 9) выдача документов.

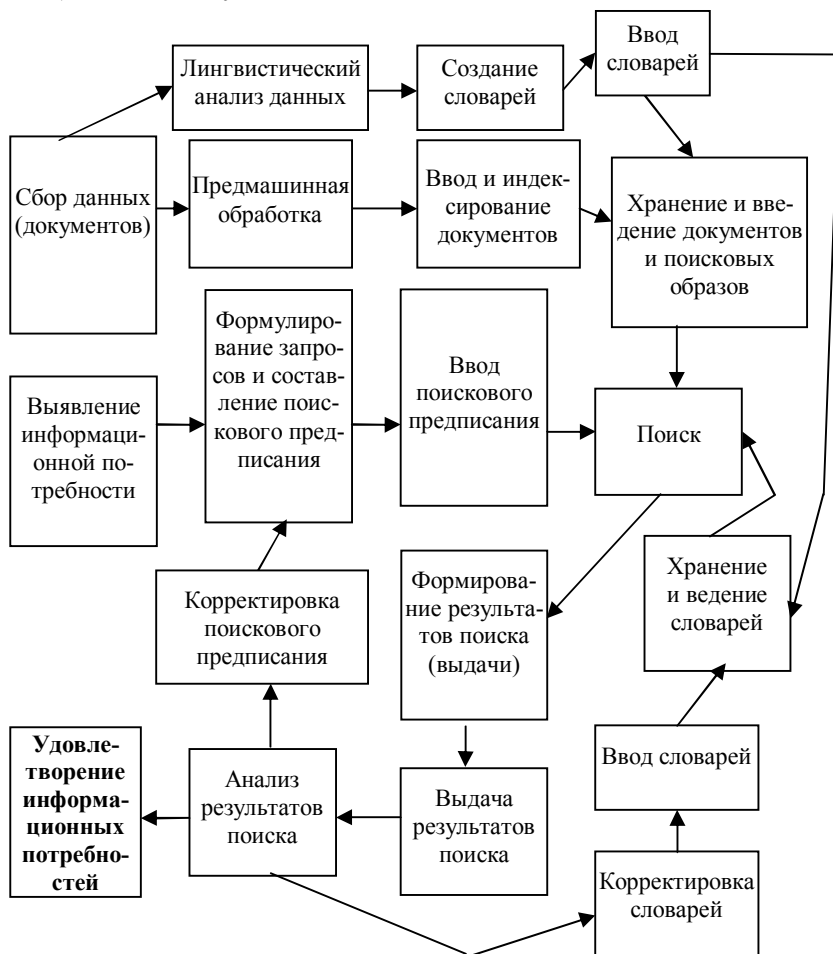


Рис. 4.1. Функционально-операционная схема документальной ИПС

4.5. Документы, информационно-поисковый язык, поисковые массивы

Понятие документа меняется с течением времени, постоянно появляются новые формы документов, изменяются старые. Российский государственный стандарт (РСТ) определяет **документ** как средство накопления любым способом на специальном материале любой информации о фактах, событиях, явлениях объективной действительности и мыслительной деятельности человека. Существуют различные классификации документов, наиболее существенной является деление документов на первичные и вторичные. К первичным документам относятся документы, исходящие от автора. К вторичным — документы, являющиеся обработкой одного или нескольких первичных. Документы имеют различные формы представления. В автоматизированных поисковых системах это прежде всего текстовая информация на естественных языках в машиночитаемой форме. Для поиска документов существенное значение имеет его внутренняя структура. Такая структура документов может разрабатываться специально для конкретной ИПС или задаваться стандартами.

Важнейшим компонентом ИПС является **информационно-поисковый язык**. Во всех случаях описания документов используются сокращения (аннотации, рефераты, каталоги), производимые на естественных языках. Отбор документов в информационных системах осуществляется с помощью специальных информационных языков. Основные требования к таким языкам сводятся:

- к однозначности между планом выражения и планом содержания (каждая лексическая единица соотносится с одним понятием и наоборот);

- к достаточной семантической силе (к способности фиксировать с достаточной полнотой и точностью все существенное в содержании документов и запросов);

- к открытости (к возможности пополнения содержания языка).

Информационные массивы в ИПС организуются по двум схемам — прямой и инвертированной.

Прямая схема реализуется следующим образом: каждая запись массива содержит поисковый образ очередного документа, представленный набором дескрипторов: ключевых слов, терминов

или других поисковых признаков **kw** (табл. 4.1). Это поддокументная организация массива. Весь массив может быть упорядочен по любому признаку (например, по номеру документа **D**).

Таблица 4.1

Идентификатор документа (ПОД)	Дескрипторы, ключевые слова или другие поисковые элементы				
D1		Kw2		Kw4	Kw5
D2	Kw1	Kw2	Kw3	Kw4	
D3		Kw2			Kw5

Полный ответ на запрос может быть получен лишь при условии сплошного просмотра массива.

В **инвертированных схемах** массивы организованы на принципе обеспечения доступа к документам через их идентификаторы (табл. 4.2):

Таблица 4.2

Поисковые элементы данных	Идентификатор документа (ПОД)				
Kw1		D2			
Kw2	D1	D2	D3		
Kw3		D2			
Kw4	D1	D2			
Kw5	D1				

Инвертированную схему получают путем обработки массива с прямой организацией и создания специальных файлов — точек доступа (индексных файлов, индексов). Просмотр инвертированного массива по поисковым признакам, содержащимся в запросе, значительно упрощает доступ к документам. Недостаток заключается в необходимости постоянного переформирования массивов после ввода новых документов, а также в большом объеме дискового пространства, необходимого для хранения инвертированных массивов.

4.6. Типы информационно-поисковых систем

Рассмотрим классификацию ИПС по их внутренним характеристикам и свойствам:

1. *По степени автоматизации информационных процессов:*
 - 1.1. Ручные.

- 1.2. Механизированные.
- 1.3. Автоматизированные.
- 2. *По типу данных:*
 - 2.1. Документальные.
 - 2.2. Фактографические.
 - 2.3. Информационно-логические.
- 3. *По типу языка:*
 - 3.1. Классификационные.
 - 3.2. Дескрипторные.
 - 3.3. Вербальные.
- 4. *По типу критерия смыслового содержания (критерий выдачи):*
 - 4.1. С количественными критериями (в основе лежат арифметические, алгебраические, статистические функции, определяющие соответствие ПОЗ и ПОД).
 - 4.2. С логическими критериями (применяется аппарат булевой алгебры в процессе поиска).
- 5. *По режиму работы:*
 - 5.1. Ретроспективные (выдают ответ исходя из всей информации, находящейся в базе информации).
 - 5.2. Избирательное распространение информации (выдают информацию только по новым поступлениям или по поступлениям за определенный срок).
- 6. *По способу хранения первичных документов:*
 - 6.1. Двухконтурный (документы хранятся как в машине, так и на различного рода материальных неэлектронных носителях).
 - 6.2. Одноконтурный (документы хранятся в машине в виде поисковых образов ПОД).

Как правило, на практике мы встречаемся с комбинацией этих признаков (например, сводный каталог электронных библиотек, справочники телефонов «Вся Самара и Самарская область»).

4.7. Лингвистическое обеспечение

Под лингвистическим обеспечением понимают комплекс языковых, логических средств и методов, используемых в ИПС для обеспечения основной задачи — сопоставления содержания

документов и запросов. К лингвистическому обеспечению относятся:

1. Информационно-поисковые языки (далее — ИПЯ):
 - 1.1. алфавит и микросинтаксис языка;
 - 1.2. средства представления плана содержания (словари, тезаурусы, рубрикаторы, классификаторы, наполнение форматов);
 - 1.3. средства представления плана выражения:
 - 1.3.1. грамматические средства ИПЯ (указатели связи, указатели роли и т.д.);
 - 1.3.2. структура форматов.
 - 1.4. Средства представления естественно языковой информации в ИПЯ (правила представления текстов на естественных языках).
2. Средства обеспечения совместимости различных языков:
 - 2.1. таблицы соответствия;
 - 2.2. таблицы инвертирования;
 - 2.3. протоколы.
3. Процедурные средства:
 - 3.1. методики индексирования;
 - 3.2. алгоритмы автоматического индексирования;
 - 3.3. средства поиска:
 - 3.3.1. язык запросов;
 - 3.3.2. средства автоматической адресации запросов к базам данных;
 - 3.3.3. единые командные языки.
4. Критерий смыслового соответствия.
5. Язык структурирования (представления) и метаописания данных.
6. Вспомогательные средства обработки текстов на естественных языках:
 - 6.1. программно-алгоритмические средства морфологического и синтаксического анализа;
 - 6.2. средства семантической обработки текстовых данных;
 - 6.3. лингвистические средства и алгоритмы автоматической классификации документов;
 - 6.4. лингвистические средства и алгоритмы автоматизированного составления поисковых предписаний;

- 6.5. средства автоматизированного поиска ошибок;
- 6.6. лингвистические банки данных, машинные словари.
- 7. Средства поддержки лингвистического обеспечения:
 - 7.1. процедуры составления машинных словарей;
 - 7.2. процедуры ведения машинных словарей;
 - 7.3. документация по лингвистическому обеспечению.

4.8. Информационно-поисковые языки с иерархической классификацией

Классификация — это система распределения объектов (предметов, явлений, процессов, понятий) по классам в соответствии с определенными признаками. Классификация объектов — это процедура группировки на качественном уровне, направленная на выделение однородных свойств. Иерархическая классификация строится следующим образом (рис. 4.2):

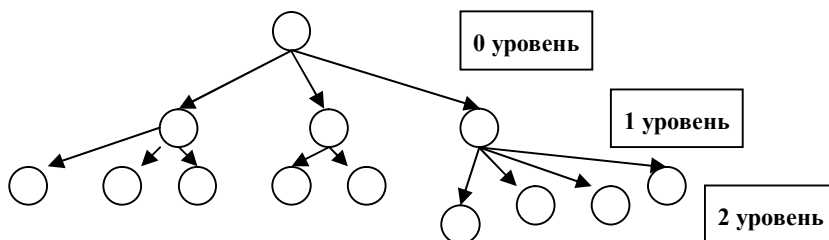


Рис. 4.2. Иерархическая система классификации

1. Исходное множество элементов составляет «0» уровень и делится в зависимости от выбранного классификационного признака на классы (группировки), которые образуют первый уровень.
2. Каждый класс первого уровня в соответствии с характерным для него классификационным признаком делится на подклассы, которые образуют второй уровень.
3. Каждый второй уровень соответственно делится на группы, образуя третий уровень, и т.д.

Наиболее известные и распространенные иерархические классификаторы:

1. DDC — десятичная классификация Дьюи.
2. УДК — универсальный десятичный классификатор.

3. ББК — библиотечно-библиографическая классификация.
4. МКИ — международная классификация изобретений.
5. ОКПО — Общероссийский классификатор продукции.
6. ИНН — идентификационный номер налогоплательщика.

В иерархической классификации обычно каждый класс, подкласс называется индексом и обозначается цифровым или буквенно-цифровым кодом, который сопровождается понятием на естественном языке, например (табл. 4.3):

Таблица 4.3

Универсальный десятичный классификатор

5	Математика. Естественные науки
53	Физика
531	Общая механика
532	Гидромеханика
536	Термодинамика
536.2	Теплопроводность, теплопередача
536.22	Теплопроводность жидкостей

Иерархические классификации очень широко распространены в различных отраслях человеческой деятельности и применяются в качестве информационно-поисковых языков (далее — ИПЯ) в электронных каталогах и автоматизированных библиотечных системах.

4.9. Вербальные информационно-поисковые языки

Использование дескрипторных языков предполагает процедуру индексирования — перевод содержания документов с естественных языков на дескрипторные (от англ. *describe* — описательные). Перевод может быть ручным или автоматизированным, но в любом случае остаются нерешенными вопросы больших затрат ручного труда и субъективность индексирования, во втором случае — нерешенность вопросов компьютерного моделирования интеллектуальных семантических процессов (индексирования). Вместо ИПЯ с лексическим контролем в настоящее время используются бестезаурусные языки, или *вербальные*. Говорят, что вербальные языки используют координирование на естественном языке. Используются слова и словосочетания естественного

языка. Использование бестезаурусных (не имеющих начальных словарей) языков затрудняет выполнение смыслоотождествления. Эта проблема решается на этапе составления поискового предписания путем избыточного индексирования. Каждому поисковому предписанию приписываются все синонимы и близкие по смыслу термины вплоть до находящихся в родовидовых отношениях.

4.10. Семантические и синтагматические языки

Наиболее близко к задаче полного представления содержания документов подошли разработчики семантических и синтагматических языков, среди которых наиболее известны семантический код Перри—Кента, языка RX—кодов и язык Синтол. Их разработчики видели основную задачу в обеспечении однозначного перевода с естественного языка на ИПЯ.

В качестве примера рассмотрим язык RX—кодов (X — термины, обозначающие предметы и понятия, R — релятемы, обозначающие свойства и бинарные отношения).

Пусть у нас есть термины:

X1 — понятие «луг»;

X2 — участок;

X3 — травянистая растительность;

X4 — пойма реки;

X5 — понятие «лес».

X6 — древесная растительность.

Релятемы:

R1 — быть видом;

R2 — быть покрытым;

R3 — быть расположенным.

Тогда определение «лес — это участок, покрытый деревьями», будет иметь вид: $X5 = R1 X2 R2 X6$. Комбинация RX кодов $X1 = R1 X1 R2 X3 R3 X4$ означает: «луг — это участок, покрытый травянистой растительностью и расположенный в пойме реки».

Следует заключить, что реально работающие документальные ИПС строятся на основе вербальных ИПЯ с простой грамматикой.

4.11. Индексирование

Индексирование — одно из важнейших понятий информационного поиска. Процесс выбора и присвоения документам и запросам или их частям индексов — лексических единиц ИПЯ — называется индексированием. В соответствии со стандартами (ГОСТ 14.407-75, 14.409 «Требования к ИПЯ, требования к ИПС технологического назначения») под индексированием понимают процесс описания их смыслового содержания средствами ИПЯ с целью обеспечения высоких показателей их поиска. Процесс индексирования, т.е. выбор дескрипторов для включения в ПОД и ПОЗ, определяется специальными инструкциями, называемыми методиками индексирования. Процесс индексирования в отечественных разработках базируется на разработках ВИНТИ (Всесоюзного института научной и технической информации), в соответствии с которыми процесс индексирования включает в себя шесть основных операций:

- 1) анализ содержания документа и выбор из текста номинативных лексических единиц, существенных с точки зрения его содержания;
- 2) формирование перечня ключевых слов, используемых в процессе свободного координатного индексирования;
- 3) нормализация ключевых слов по форме и содержанию при помощи словарей ИПЯ, используемых при контролируемом индексировании;
- 4) избыточное индексирование — введение в под дополнительных лексических единиц, связанных по смыслу с исходными ключевыми словами и выбираемых из словаря ИПЯ;
- 5) введение в под грамматических средств (указателей роли и связи);
- 6) заполнение рабочего листа предмашинного формуляра под или его ввод в компьютер.

Практически с самого начала развития ИПС предпринимались попытки автоматизации процедур индексирования. В этой области накоплен достаточно большой теоретический и практический опыт. Существуют и практически работают системы (Скобки, Yandex, АИДОС, CAS).

4.12. Языки запросов, структура запросов

Языки запросов, служащие для построения поисковых предписаний, объединяют собственно язык и критерий смыслового соответствия, а также содержат требования к интерфейсу выдачи. В общем случае язык запросов содержит следующие компоненты:

1. Поисковые элементы (термины, выражающие информационную потребность).
2. Средства морфологической нормализации элементов запроса.
3. Поисковые булевы операторы.
4. Средства линейной грамматики.
5. Дополнительные условия поиска:
 - 5.1. поиск в определенных полях;
 - 5.2. ограничение области поиска по языку, региону и т.д.
6. Средства управления критериями смыслового содержания.
7. Требования на сортировку (ранжирование) выдаваемых результатов поиска.
8. Требования к форме представления результатов поиска (вид выдаваемых результатов, количество выдаваемых документов).

Поисковые системы в настоящее время базируются на булевой (логической) модели поиска. Запрос в этих системах представляет булево выражение — набор поисковых терминов, объединенных логическими операторами с учетом правил поискового синтаксиса. NOT — выполняется первыми, AND — вторым, OR — третьим. Следует помнить, что действия, объединенные в скобки, осуществляются первыми.

4.13. Язык описания данных

Языки описания данных состоят из двух составляющих: языка представления данных и языка описания метаданных. Языки представления задают формальные характеристики документов: форматы, кодировку, тип, структуру документов. Языки метаданных представляют собой данные о данных, вплоть до таких, какие отсутствуют в документе (например, страна, язык, правовые аспекты использования).

К языкам представления относятся прежде всего языки библиографического описания данных, применяемых при организации

каталогов и картотек. На автоматизированный поиск рассчитаны машинные библиографические форматы. Их основу составляют форматы MARC (Machine-Aided Readable Cataloging) и ISO 2709. На современном этапе наиболее распространены международные форматы UNIMARC, американский MARC21 и в России RUS-MARC.

Международный коммуникативный формат предназначен для обмена библиографическими данными в машиночитаемой форме. UNIMARC определяет структуру и наполнение библиографических записей и включает «МАРКЕР» записи, «СПРАВОЧНИК», «ПОЛЯ ДАННЫХ».

Маркер записи 24-х символьный и, кроме всего прочего, содержит элементы, описывающие структуру справочника и полей данных. Справочник представляет собой перечень статей. Каждая статья справочника состоит из трех частей: метки поля, относительного адреса данного поля, длины поля. Например, Вестник Санкт-Петербургского университета будет иметь запись: 011 ##\$a0132-4624.

Для электронных документов разработаны и используются следующие форматы:

- 1) текст, структурированный или нет (ISO 646);
- 2) текст на языке SQLML (ISO 8879);
- 3) текст на языке HTML;
- 4) ODA-текст (ISO 8613);
- 5) CCIT-формат кодировки графических образов страниц;
- 6) Postscript-файлы;
- 7) мультимедийные составные файлы (текст, аудио, видео).

В связи с ростом Интернет найти необходимый ресурс становится все сложнее, использование форматов (UNIMARC, MARC21) в сети нереально, т.к. они слишком детальные.

Работы по стандартизации набора семантических свойств с ориентацией на электронные документы были активизированы в 1995 г. после семинара в Дублине (штат Огайо), в связи с чем вариант языка метаданных получил название Дублинского ядра. Текущая версия спецификаций Дублинского ядра включает 15 элементов или полей, которые могут повторяться. При описании поля кроме имени вводят понятия схемы и подполей.

Схема — это наименование правил, в которых оговаривается содержание полей. Например:

- 1) поле «Дата» — стандарт ввода дат;
- 2) поле «Язык» — стандарт кодировки ASCII, KOI8 и т.д.

Подполе — это информация, уточняющая содержание поля. Перечень подполей еще четко не определен. При использовании стандарта DC описание ресурса может быть прочитано специальным роботом и помещено в каталог с разбиением на поля.

Кратко рассмотрим описание некоторых элементов Дублинского ядра (Dublin Core Metadata Element Set).

1. Название ресурса.

Поле: **Title**.

Определение: имя, данное ресурсу автором или издателем.

Схема: не используется.

Подполя:

DC. **Title** — основное заглавие (подполе по умолчанию);

DC. **Title Alternative** — альтернативное заглавие.

2. Автор или создатель.

Поле: **Creator**.

Определение: лицо или организация, ответственные за содержание ресурса.

Подполя:

DC. **Creator** — автор (подполе по умолчанию);

DC. **Creator. Personal Name** — имя индивидуального автора;

DC. **Creator. Corporate Name** — имя коллективного автора (включая наименование конференций);

DC. **Creator. Personal Name. Address** — любой тип адреса индивидуального автора, включая электронный;

DC. **Creator. Corporate Name. Address** — любой тип адреса коллективного автора, включая электронный.

3. Предмет и ключевые слова.

Поле: **Subject**.

Определение: описание ресурса. Обычно предмет описывается ключевыми словами или фразами. Рекомендуется использовать контролирующие словари или формальные классификационные схемы.

1. Описание.

Поле: **Description**

Определение: описание содержания ресурса. Это может быть краткий реферат, перечень содержимого и т.п.

Остальные поля далее рассмотрим в порядке перечисления:

2. Издатель (пять подполей).

3. Сведения об ответственности (сведения об организации или лице, внесших значительный вклад в создание ресурса, но не указанных в поле «автор» или «создатель», содержит пять подполей).

4. Дата (согласовано только два подполя: «дата создания» и «дата поступления»).

5. Тип ресурса (текст, программы, изображения, перечень которых приведен в Интернете):

<http://sunsite.berkley.edu/Metadata/types.html>

6. Формат используется при определении программного и технического обеспечения (doc, pdf, htm, jpeg). Для его определения рекомендуется использовать контролируемый словарь стандарта **MIME** (Multipurpose Internet Mail Extensions) — Internet Media Types.

7. Идентификатор ресурса (строка или число для однозначного определения ресурса).

8. Источник (строка или число для однозначного определения источника, из которого ресурс был создан).

Схема:

Текст по умолчанию:

URL (Uniform Resource Locator):

ISBN (International Standard Book Number) и т.д.

9. Язык.

10. Отношения (указываются другие ресурсы, на которые имеются ссылки).

11. Охват (пространственная и временная характеристика).

12. Правовые аспекты (источник информации об авторских правах).

Стандарт **DC** полностью соответствует стандарту **HTML**, и поэтому он может непосредственно включаться в сам ресурс. В языке **HTML** для этого используется специальный тег **Meta** с атрибутами **NAME** (название поля) и **CONTENT** (значение). В последних версиях **HTML** введены атрибуты **LANG** и **SCHEME**,

которые позволяют задать язык представления и соответственно схему, уточняющую контекст.

Стандарт **DC** достаточно прост, и для разработчиков ресурсов существуют сайты с формами для описания ресурса и набора метаданных, а также правил кодировки.

4.14. Информационный поиск в сети Интернет

Интернет — это огромная компьютерная сеть (далее — Сеть), состоящая из тысяч меньших сетей, разбросанных по всему миру. Интернет возник из ARPANET сети Министерства обороны США.

Для размещения информации до конца 1980-х годов использовался (используется) сервис, получивший название *FTP* по имени механизма обмена информацией (протокол FTP). В настоящее время FTP-архивы превратились в огромные многопрофильные хранилища данных.

Значительное влияние на развитие сети оказали протоколы транспортного и межсетевого обмена **TCP** (Transfer Control Protocol) и **IP** (Internet Protocol).

Важной вехой в распространении информации в сети стало рождение системы телеконференций **Usenet** (зародилась из электронной доски объявлений и обзавелась собственным протоколом **NNTP — News Network Transfer Protocol**). В настоящее время это огромная электронная система оперативных сообщений, разделенная на части по интересам ее пользователей. Каждая группа новостей имеет свое название с иерархической структурой, например:

- 1) наука (**sci**);
- 2) группа новостей посвященных биологии (**sci.biology**) и т.д.

Другим важным средством обмена информацией в Интернет является **электронная почта**. С появлением стандарта **MIME** электронная почта получила возможность посылать не только простые текстовые сообщения, но и текст с вложениями от программных до аудио, видео и графических файлов.

До появления компьютерных сетей существовали информационные системы и центры (SNT International, ВИНТИ, ВНИИЦ, ВНИИППЭ), накапливающие научную, техническую, юридическую

информацию. Эти ресурсы стали доступны в Интернет после появления механизма доступа Telnet (удаленный терминал).

Основные протоколы, используемые в Интернет, не обеспечены достаточными встроенными функциями поиска, не говоря уже о миллионах серверах, находящихся в ней. Протоколы НТТР и более простой FТР, используемые в Интернет, хороши лишь в отношении навигации, которая рассматривается как средство просмотра страниц, но не их поиска. Из-за быстрого роста информации, доступной в Сети, навигационные методы просмотра достигли предела эффективности и функциональных возможностей.

Нужную информацию невозможно получить сразу, т.к. в Сети сейчас находятся миллиарды документов, их количество возрастает согласно экспоненциальной зависимости. Количество изменений (преобразований, кодировок, индексаций), которым эта информация подвергается при хранении в Сети, огромно и, самое главное, они происходят за очень короткий промежуток времени.

Основная проблема заключается в том, что единой, полной функциональной системы обновления и занесения подобного объема информации с обеспечением одновременного доступа всех пользователей Интернет никогда не было. Для обеспечения пользователей удобными средствами поиска были созданы информационные поисковые системы, структурирующие накопленную в Сети информацию.

4.15. Поисковые системы, механизмы поиска

Архитектура поисковой системы имеет вид (рис. 4.3), а поисковые системы обычно содержат три компонента:

1. **Агент** (или паук, или кроулер, или робот), который перемещается по Сети и собирает информацию.
2. **Поисковая база данных**, которая содержит всю информацию, собираемую пауками.
3. **Поисковый механизм**, который люди используют как интерфейс для взаимодействия с базой данных.

Средства поиска и структурирования информации в Сети, называют поисковыми механизмами.

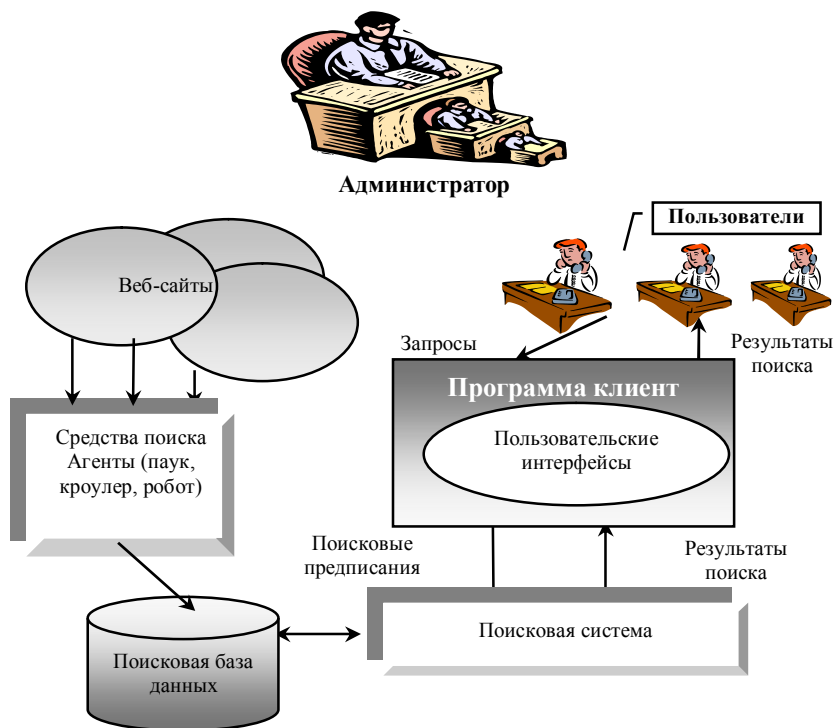


Рис. 4.3. Архитектура поисковой системы

Эти специальные программы занимаются поиском **Веб-сайтов** и страниц в Сети, извлекают гипертекстовые ссылки, автоматически индексируют их и используют для построения базы данных.

Веб-сайты — все информационные ресурсы Интернет, просмотр которых обеспечивается программами. **Агент** (разновидности — **паук, кроулер, робот**) — система, обеспечивающая сканирование Интернет и поддержание инвертированного файла в актуальном состоянии. Этот программный комплекс является основным источником информации о состоянии ресурсов сети. Для обхода веб-сайтов используется теория графов. Обход дерева графов осуществляется тремя способами, а именно:

- 1) случайный выбор URL-адреса;
- 2) обход с наиболее популярных веб-узлов;
- 3) на основе системы имен или кодов стран (в этом случае выделяется дополнительная программа **Робот**).

Поисковая база данных — специальным образом организованная база (индекс), включающая инвертированный файл, состоящий из лексических единиц, проиндексированных веб-документов, лексем, определяющих их позицию в документах и на сайте в целом.

База данных отыскивает предмет запроса, основанный на информации, указанной в заполненной форме, и выводит соответствующие документы, подготовленные базой данных.

Для определения порядка, в котором список документов будет показан, база данных применяет алгоритм ранжирования. В идеальном случае документы, наиболее релевантные пользовательскому запросу, будут помещены первыми в списке.

Различные поисковые системы используют различные алгоритмы ранжирования, однако основные принципы определения релевантности следующие:

1) количество слов запроса в текстовом содержимом документа (т.е. в html-коде);

2) тэги, в которых эти слова располагаются;

3) время — как долго страница находится в базе поискового сервера. Поначалу кажется, что это довольно бессмысленный принцип. Существует много сайтов, живущих максимум месяц. Если сайт существует довольно долго, это означает, что владелец весьма опытен в данной теме и пользователю больше подойдет сайт, который пару лет вещает миру о правилах поведения за столом, чем тот, который появился неделю назад с этой же темой;

4) индекс цитируемости — как много ссылок на данную страницу ведет с других страниц, зарегистрированных в базе поисковика.

База данных выводит ранжированный подобным образом список документов с HTML и возвращает его пользователю, сделавшему запрос. Различные поисковые механизмы также выбирают различные способы показа полученного списка: некоторые показывают только ссылки; другие выводят ссылки на первые предложения, содержащиеся в документе, или заголовок документа вместе со ссылкой. При заполнении базы данных сотрудники базы могут помещать информацию прямо в индекс, заполняя особую форму для того раздела, в который они помещают свою информацию. Эти данные передаются базе данных.

Поисковая система — подсистема поиска, обрабатывающая поисковые предписания пользователя, поиск в базе и выдачу результатов поиска.

Клиент — программа просмотра информационных ресурсов (Opera, Netscape Navigator, Internet Explorer).

Пользовательские интерфейсы — экранные формы общения пользователя с поисковым аппаратом.

Каждый поисковый механизм имеет собственный набор правил сбора документов. Некоторые следуют за каждой ссылкой на каждой найденной странице и затем, в свою очередь, исследуют каждую ссылку на каждой из новых страниц. Другие игнорируют ссылки, которые ведут к графическим и звуковым файлам, файлам мультимедиа. Третьи игнорируют ссылки к ресурсам типа баз данных WAIS. Четвертые проинструктированы, что нужно просматривать прежде всего наиболее популярные страницы.

4.16. Агенты и кроулеры, роботы

Агенты — самые «интеллектуальные» из поисковых средств. Кроме собственно функций поиска, они могут:

- 1) выполнять транзакции от Вашего имени;
- 2) искать сайты специфической тематики и возвращать списки сайтов, отсортированных по индексу цитируемости или посещаемости;
- 3) обрабатывать содержание документов, находить и индексировать другие виды ресурсов, не только страницы;
- 4) программироваться для извлечения информации из уже существующих баз данных.

Независимо от информации, которую агенты индексируют, они передают ее обратно базе данных поискового механизма.

Агенты извлекают и индексируют различные виды информации. Индексация может осуществляться различными способами, в числе которых:

- индексирование отдельных слов во встречающемся документе;
- индексирование только наиболее важных ста слов в каждом;
- индексирование размера документа и число слов в нем;
- индексирование названия, заголовка и подзаголовка.

Вид построенного индекса определяет, какой поиск может быть сделан поисковым механизмом и как полученная информация будет интерпретирована.

Администраторы поисковых систем могут определить, какие сайты или типы сайтов агенты должны посетить и проиндексировать. Проиндексированная информация отсылается базе данных поискового механизма так же, как было описано выше.

Пауки — это программы, осуществляющие общий поиск информации в Сети. Пауки сообщают о содержании найденного документа, индексируют его и извлекают итоговую информацию, просматривают заголовки, некоторые ссылки и посылают проиндексированную информацию базе данных поискового механизма.

Кроулеры — это программы, просматривающие заголовки и возвращающие только первую ссылку.

Роботы могут программироваться так, чтобы можно было переходить по различным ссылкам различной глубины вложенности, выполнять индексацию и даже проверять ссылки в документе. Из-за их природы они могут застревать в циклах, поэтому, проходя по ссылкам, им нужны значительные ресурсы Сети. Однако имеются методы, предназначенные для того, чтобы можно было запретить роботам поиск по сайтам, владельцы которых не желают, чтобы они были проиндексированы.

Роботы могут использоваться для выполнения множества полезных задач, таких как статистический анализ, обслуживание гипертекстов, исследование ресурсов или зазеркаливание страниц.

Зазеркаливание — популярный механизм поддержания FTP-архивов. Зеркало рекурсивно копирует полное дерево каталогов по FTP, а затем регулярно перезапрашивает те документы, которые изменились. Это позволяет распределить загрузку между несколькими серверами, успешно справиться с отказами сервера и обеспечить более быстрый и более дешевый локальный доступ, так же как и автономный доступ к архивам.

4.17. Сравнительный обзор поисковых систем

Когда пользователю необходимо найти информацию, доступную в Интернете, он посещает страницу поисковой системы и

заполняет форму, детализирующую информацию. Здесь могут использоваться ключевые слова, даты и другие критерии. Критерии в форме поиска должны соответствовать критериям, используемым агентами при индексации информации, которую они нашли при перемещении по Сети.

Для сервиса **FTP**, была разработана система поиска и выдачи информации **Archie**. Поиск производится по названию файлов или каталогов. Возможен поиск по словам, содержащимся в кратких описаниях файлов (аналог аннотаций).

Развитие информационных технологий хранения и поиска данных в сети привело к созданию распределенной информационно-справочной системе **Gopher**. В основе системы лежит представление информации в форме иерархического дерева по аналогу с деревом каталогов в операционных системах (разрабатывалась для продвинутых студентов университетов штата Миннесота). Информация хранится в виде электронных каталогов на **Gopher**-серверах, образуя так называемое **Gopher**-пространство. Универсальной поисковой системой считается **Veronica**, сканирующая **Gopher**-пространство, как простую текстовую базу данных. Поиск осуществляется на основе ключевых слов.

Lycos. В Lycos (рис. 4.4) используется следующий механизм индексации:

1. Слова в <title> заголовке имеют высший приоритет.
2. Слова в начале страницы.
3. Слова в ссылках.
4. Если в его базе индекса есть сайты, ссылка с которых указывает на индексируемый документ, то релевантность этого документа возрастает.

Как и большинство систем, Lycos дает возможность применять простой запрос и более изощренный метод поиска. В простом запросе в качестве поискового критерия вводится предложение на естественном языке, после чего Lycos производит нормализацию запроса, удаляя из него так называемые stop-слова, и только после этого приступает к его выполнению. Почти сразу выдается информация о количестве документов на каждое слово, а позже и список ссылок на формально релевантные документы. В списке против каждого документа указывается его мера близости запросу, количество слов из запроса, попавших в документ, и оценочная

мера близости, которая может быть больше или меньше формально вычисленной. Пока нельзя вводить логические операторы в строке вместе с терминами, но использовать логику через систему меню Lycos позволяет. Такая возможность применяется для построения расширенной формы запроса, предназначенной для искушенных пользователей, уже научившихся работать с этим механизмом. Таким образом, видно, что Lycos относится к системе с языком запросов типа «Like this», но намечается его расширение и на другие способы организации поисковых предписаний.



Рис. 4.4. Фрагмент интерфейса Lycos

AltaVista. Индексирование в этой системе (рис. 4.5) осуществляется при помощи робота, который при этом имеет следующие приоритеты:

1. Слова в теге <title> имеют высший приоритет; ключевые фразы в <Meta> тэгах.
2. Ключевые фразы, находящиеся в начале странички.
3. Ключевые фразы в ALT-ссылках

4. Ключевые фразы по количеству вхождений\присутствия слов\фраз.

Если тэгов на странице нет, использует первые тридцать слов, которые индексирует и показывает вместо описания (tag description)

Наиболее интересная возможность AltaVista — это расширенный поиск. Здесь стоит сразу оговориться, что в отличие от многих других систем AltaVista поддерживает одноместный оператор NOT. Кроме этого, имеется еще и оператор NEAR, который реализует возможность контекстного поиска, когда термины должны располагаться рядом в тексте документа. AltaVista разрешает поиск по ключевым фразам, при этом она имеет довольно большой фразеологический словарь. Кроме всего прочего, при поиске в AltaVista можно задать имя поля, где должно встретиться слово: гипертекстовая ссылка, applet, название образа, заголовок и ряд других полей. К сожалению, подробно процедура ранжирования в документации по системе не описана, но видно, что ранжирование применяется как при простом поиске, так и при расширенном запросе. Реально эту систему можно отнести к системе с расширенным булевым поиском.



Рис. 4.5. Фрагмент интерфейса AltaVista

OpenText. Информационная система OpenText представляет собой самый коммерциализированный информационный продукт

в Сети. Все описания больше похожи на рекламу, чем на информативное руководство по работе. Система позволяет провести поиск с использованием логических коннекторов, однако размер запроса ограничен тремя терминами или фразами. В данном случае речь идет о расширенном поиске. При выдаче результатов сообщается степень соответствия документа запросу и размер документа. Система позволяет также улучшить результаты поиска в стиле традиционного булевого поиска. OpenText можно было бы отнести к разряду традиционных информационно-поисковых систем, если бы не механизм ранжирования.

Infoseek. В этой системе индекс создает робот, но он индексирует не весь сайт, а только указанную страницу. При этом робот имеет такие приоритеты:

1. Слова в заголовке <title> имеют наивысший приоритет.
2. Слова в тегах keywords, description и частота вхождений\повторений в самом тексте.
3. При повторении одинаковых слов рядом выбрасывает из индекса.
4. Допускает до 1024 символов для тега keywords, 200 символов для тэга description;
5. Если тэги не использовались, индексирует первые 200 слов на странице и использует как описание.

Система Infoseek обладает довольно развитым информационно-поисковым языком, позволяющим не просто указывать, какие термины должны встречаться в документах, но и своеобразно взвешивать их. Достигается это при помощи специальных знаков: «+» — термин обязан быть в документе, «-» — термин должен отсутствовать в документе. Кроме этого, Infoseek позволяет проводить то, что называется контекстным поиском. Это значит, что, используя специальную форму запроса, можно потребовать последовательной совместной встречаемости слов. Можно также указать, что некоторые слова должны совместно встречаться не только в одном документе, но и в отдельном параграфе или заголовке. Имеется возможность указания ключевых фраз, представляющих собой единое целое, вплоть до порядка слов. Ранжирование при выдаче осуществляется по числу терминов запроса в документе, по числу фраз запроса за вычетом общих слов. Все эти факторы используются как вложенные процедуры. Таким образом,

Infoseek относится к традиционным системам с элементом взвешивания терминов при поиске.

WAIS. Дальнейшее совершенствование сети привело к созданию сервиса **WAIS** — распределенной поисковой, осуществляющей поиск в сети по ключевым словам. Результат поиска представляется в виде списка файлов, расположенных по убывающей на основе частоты встречаемости или местоположения.

WAIS является одной из наиболее изощренных поисковых систем Интернета. В ней не реализованы лишь поиск по нечетким множествам и вероятностный поиск. В отличие от многих поисковых машин, система позволяет строить не только вложенные булевы запросы, считать формальную релевантность по различным мерам близости, взвешивать термины запроса и документа, но и осуществлять коррекцию запроса по релевантности. Система также позволяет использовать усечения терминов, разбиение документов на поля и ведение распределенных индексов. Не случайно именно эта система была выбрана в качестве основной поисковой машины для реализации энциклопедии «Британика» на Internet.

Для использования расположенной в Интернете информации в научных, образовательных, коммерческих целях была разработана технология **World Wide Web**, позволяющая связывать информацию из различных источников и делать ее доступной из любой точки Сети. Характерная особенность **WWW** — использование гипертекстовых технологий. Само понятие гипертекста возникло задолго до появления Сети. Главная идея гипертекста — нелинейная навигация по тексту, заключающаяся в том, что элементы документа образуют структурную или ассоциативную сеть, движение по которой определяется пользователем в момент просмотра текста (например: справочная система **Windows** и **Windows** совместимых приложений генерирует справку с контекстной привязкой к текущей ситуации на экране либо с помощью помощника, ключевых слов, дерева каталогов).

Для гипертекстовых систем характерны «меню ориентированные» способы работы.

Приведем несколько определений гипертекста:

1) многомерное текстовое пространство, построенное на ассоциативных связях внутри и между документами;

2) нелинейная последовательность записи и чтения информации, основанная на объединении ассоциативно связанных блоков информации;

3) способ хранения и манипулирования информацией, при котором она представлена в виде сети связанных между собой узлов;

Каждый узел может содержать:

а) текст, графику, видео, аудио информацию;

б) доступ к узлам (просмотр и манипулирование) в интерактивном режиме.

Наиболее распространенные поисковые системы Сети используют как классификационный, так и вербальный принципы построения, но степень развития их в конкретных системах неодинакова.

Yahoo! Данная система появилась в Сети одной из первых, и сегодня Yahoo сотрудничает со многими производителями средств информационного поиска, а на различных ее серверах используется различное программное обеспечение. Язык Yahoo достаточно прост: все слова следует вводить через пробел, они соединяются связкой AND либо OR. При выдаче не указывается степень соответствия документа запросу, а только подчеркиваются слова из запроса, которые встретились в документе. При этом не производится нормализация лексики и не проводится анализ на «общие» слова. Хорошие результаты поиска получаются только тогда, когда пользователь знает, что в базе данных Yahoo информация есть наверняка. Ранжирование производится по числу терминов запроса в документе. Yahoo относится к классу простых традиционных систем с ограниченными возможностями поиска.

На верхнем уровне содержится 13 разделов (категорий). Общее число рубрик (разделов, подразделов и т.д.) составляет более 25 тысяч. Очень подробная и продуманная схема классификации **Yahoo!** оказалась при подробном анализе достаточно близка к УДК (из 50 наиболее популярных категорий только 4 не подпадают под DDC или УДК). **Yahoo!** объективно отражает все отрасли знания без приоритетов, учитывая не только веб-сайты, телеконференции, чаты. Поиск может осуществляться путем ввода ключевых слов в поисковую строку (рис. 4.6) либо путем просмотра поисковых категорий. В верхней части экранного интерфейса располагаются типовые классификационные индексы Интернет.

К ним относятся (рис. 4.7): Интернет; Покупки; Спорт; Знакомства; СМИ.

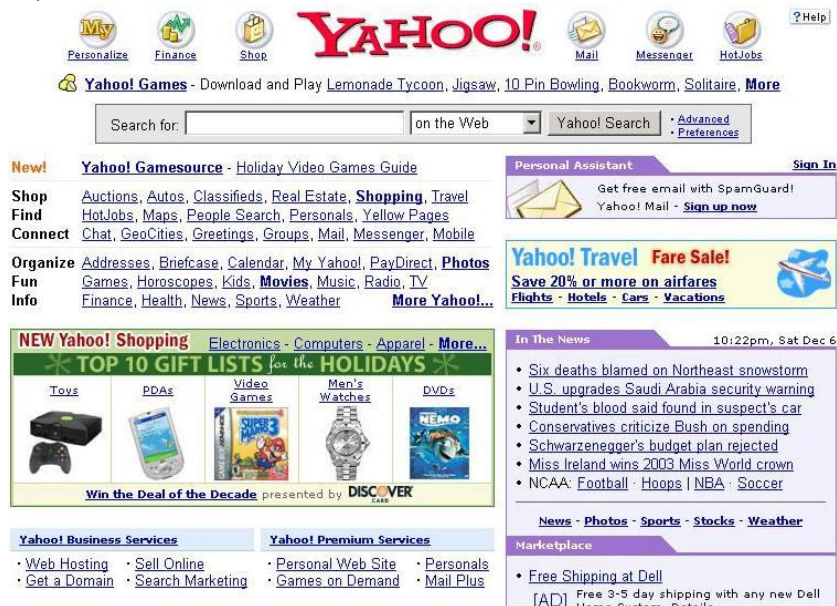


Рис. 4.6. Фрагмент интерфейса ввода запросов в системе Yahoo!



Рис. 4.7. Схема классификационной системы Yahoo!

В средней части экранного интерфейса располагается основной ряд классификационных индексов **Web Site Directory** (рис. 4.7), содержащий рубрики «Экономика и Бизнес», «Образование», «Здравоохранение», «Компьютеры и Интернет» и т.д. Поиск по индексам разворачивает выстроенные в иерархическом порядке пункты меню, постепенно сужающие и конкретизирующие тему. В нижней части экранного интерфейса (рис. 4.8) расположены дополнительные классификационные индексы: географические **Local Yahoo!s** (ограничивают поиск по месту) и расширенные **More Yahoo!** (инструкции, малый бизнес, предпринимательство и т.д.).

Local Yahoo!s

Europe	Asia Pacific	Americas		
• Catalan • Denmark • France • Germany • Italy	• Asia • Australia & NZ • China • Hong Kong • India	• Japan • Korea • Singapore • Taiwan	• Argentina • Brazil • Canada	• Mexico • U.S. in Chinese • U.S. in Spanish

U.S. Cities: [Atlanta](#) - [Boston](#) - [Chicago](#) - [Houston](#) - [LA](#) - [NYC](#) - [SF Bay](#) - [Seattle](#) - [more...](#)

More Yahoo!

Guides	Small Business	Enterprise	Personal Finance
• Education • Health • Lottery • Members • Pets • Yahoo!igans!	• Get a Domain Name • Promote Your Business • Small Biz Center • Sell Online • Web Hosting	• Enterprise Solutions • Business Messenger • Enterprise My Yahoo! • Professional Services • Resumix	• Banking • Bill Pay • Money Manager • Insurance • Loans • Taxes

Рис. 4.8. Схема дополнительной классификационной системы Yahoo!

К недостатку системы **Yahoo!** следует отнести высокий процент ссылок на устаревшие источники, что свидетельствует об отсутствии механизмов автоматической проверки актуальности отраженных материалов, а огромные объемы документов затрудняют поддержание справочника в актуальном состоянии.

Yandex. Основным инструментом автоматизированного поиска в Интернет следует считать вербальные поисковые системы. Наиболее крупной и развитой вербальной информационно-поисковой системой в Российском Интернете (Рунете) обладает **Yandex**. Это оригинальная разработка фирмы CompuTek.

Яндекс содержит более 600 тыс. уникально проиндексированных серверов, количество документов превышает 70 млн. веб-страниц. Поиск выполняется с учетом русской морфологии на основе очень мощного языка запросов.

Поиск осуществляется путем ввода ключевых слов в поисковую строку (рис. 4.9).



Рис. 4.9. Интерфейс ввода запросов в Yandex

Результаты запросов отражаются в виде перечня страниц и сайтов (рис. 4.10). Дополнительно в виде закладок отражаются возможности поиска по рубрикам.

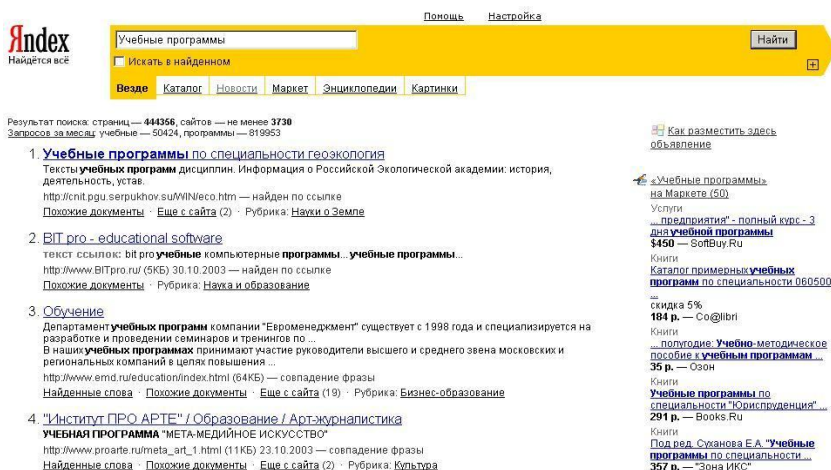


Рис. 4.10. Экранная форма результатов поиска в Yandex

Контрольные вопросы

1. Дайте определение термина *телемедицина*.
2. Перечислите основные телемедицинские задачи информационных систем.
3. Охарактеризуйте назначение информационно-поисковых систем.
4. Какие информационно-поисковые системы относятся к документальным?

5. Какие информационно-поисковые системы относятся к фактографическим?
6. Перечислите типы релевантности.
7. Дайте определение термина *рeртинeнтнocть*.
8. Перечислите основные требования к информационно-поисковым языкам.
9. Поясните функционально-операционную схему документальной ИПС.
10. По каким схемам организуются информационные массивы в ИПС?
11. Поясните различия между прямым и инвертированным информационным массивом.
12. Перечислите типы информационно-поисковых систем.
13. Что понимают под лингвистическим обеспечением ИПС?
14. Приведите пример иерархической классификации.
15. Что понимают под процедурой индексирования?
16. Перечислите основные требования к языку запросов.
17. Перечислите функции агента.
18. Перечислите функции паука или кроулера.
19. Перечислите функции робота.

5. Корпоративные системы документооборота

5.1. Корпоративные информационные системы

Корпоративной информационной системой (КИС) называется совокупность специализированного программного обеспечения и вычислительной аппаратной платформы, на которой установлено и настроено программное обеспечение.

Корпоративные информационные системы являются развитием систем для рабочих групп и могут поддерживать территориально разнесенные узлы или сети. Для них характерны архитектура «клиент — сервер» со специализацией серверов или же многоуровневая архитектура. Наибольшее распространение при разработке корпоративных информационных систем получили Oracle, DB2, SQL Server.

Корпоративные информационные системы — это собирательный, нестрогий термин, в чем-то аналогичный использовавшемуся в 60—70-е годы термину АСУ. Назначение таких систем — это учет, контроль, анализ и оптимизация финансово-хозяйственной деятельности предприятия. Они должны иметь гибко настраиваемый инструмент разработки, что позволяет адаптировать систему для работы на любом объекте учета предприятия. Цель корпоративной информационной системы — обеспечить руководителей и работников предприятия комплексной и качественной информацией для принятия и контроля исполнения управленческих решений.

Корпоративная информационная система должна обеспечивать:

- 1) комплексный подход к составу подсистем и задач автоматизации управления предприятием;
- 2) работу всех программных модулей системы в едином информационном пространстве с предоставлением возможности локальной работы отдельных подсистем, групп пользователей и рабочих мест;
- 3) использование единой системы документооборота, обеспечение принципа однократности ввода данных, возможность использования выходных документов системы в качестве первичных;
- 4) возможность одновременного обслуживания системой нескольких предприятий с получением консолидированных отчетов;

5) возможность работы пользователей, обслуживающих разные предприятия, с едиными или автономными аналитическими справочниками;

6) открытость структур хранения информации;

7) возможности работы в условиях распределенной обработки данных;

8) использование во всех подсистемах общих программных библиотек, с поддержанием единой технологии обработки данных и стандартизованных интерфейсов.

Использование корпоративных систем документооборота в практической деятельности любой организации позволяет:

- значительно облегчить выполнение задач исполнителям, занятым обработкой информации, вовлеченной в документооборот;

- обеспечить руководителям и персоналу, ответственному за организацию технологических процессов обработки и движения документов, достижение качественного уровня в решении следующих задач:

- организация единого централизованного хранилища документов, его оптимальная систематизация, что обеспечит удобный доступ пользователей к самой свежей и актуальной информации;

- улучшение средств контроля за ходом выполнения процессов обработки документов, регулирование и управление системой документооборота, что обеспечит сокращение сроков выполнения работ, существенное повышение производительности труда;

- повышение степени защищенности данных, вовлеченных в документооборот;

- достижение высокой степени независимости работы с документами от личных качеств сотрудников организации, за счет автоматического исполнения большинства формальных действий; заложенная в архитектуру системы широкая функциональность позволяет использовать для решения широкого круга задач.

Практически все системы документооборота работают на Интернет-технологиях и используют средства защиты данных, применяемые западными финансовыми институтами для обеспечения безопасности финансовых транзакций.

Системы функционируют как через закрытую от доступа извне локальную сеть компании, так и через Интернет для организации

единой корпоративной системы документооборота между территориально удаленными подразделениями компании.

Системы документооборота интегрируются с MS Office, позволяя пользователям кардинально упростить работу с документами, обеспечить гибкость и настраиваемость документов для работы с привычными файлами, а сами документы находятся на сервере, а не на компьютерах пользователей, что существенно повышает безопасность.

5.2. Корпоративная система электронного документооборота Lotus Notes

Корпоративная система электронного документооборота Lotus Notes является свободно распространяемой (дистрибутив доступен через Internet). Используя Lotus Notes, возможно в сжатые сроки обеспечить документооборот в ЛПУ. Основные возможности Lotus Notes:

- ✓ регистрация входящих и исходящих документов;
- ✓ сохранение версий документов;
- ✓ создание профилей документов;
- ✓ интуитивная иерархия контейнеров;
- ✓ полномасштабные средства текстового поиска и получения информации.

Lotus Notes 6.0 — многоплатформная (функционирует под многими операционными системами), многопротокольная (может работать по многим сетевым протоколам) программная система для групповой работы с документами. Lotus Notes 6.0 предназначена для организации и управления потоком деловой информации, документооборотом, делопроизводством, работой с базами данных, электронной почты. Работа с Lotus Notes 6.0 представляет полный доступ к Word Wide Web и к другим службам Internet, позволяя просматривать и управлять как внутренними, так и внешними информационными ресурсами. Программное обеспечение Lotus Notes 6.0 относится к классу программ для рабочих групп или корпоративных информационных систем. Возможно индивидуальное, групповое (в небольших организациях) и корпоративное (в международных компаниях) применение.

Программное обеспечение Notes состоит из клиентской части (станции Notes) и серверной части (сервера Notes).

Сервер Notes отвечает:

- ✓ за предоставление разным пользователям в совместное использование расположенных на нем баз данных (как сервер баз данных);

- ✓ хранение пользовательских почтовых ящиков и передача почты между серверами по расписанию (как почтовый сервер);

- ✓ выполнение репликаций баз данных между серверами по расписанию (поддержка распределенных баз данных);

- ✓ взаимодействие с пользователями и серверами, имеющими доступ к серверу только по модему непосредственно (как DialUp-сервер) или с использованием сервиса удаленного доступа к локальной сети;

- ✓ предоставление пользователям и серверам доступа через данный сервер к другим серверам (как Passthru-сервер);

- ✓ обеспечение безопасности.

Администратор Notes устанавливает, контролирует и управляет работой сервера Notes и обеспечивает его безопасность. Управление сервером может выполняться администратором как непосредственно с сервера, так и дистанционно, со станции администратора.

Программное обеспечение станции Notes (клиента) реализует графический интерфейс пользователя. Со своих станций пользователи имеют доступ к почтовым услугам, предоставляемым сервером Notes, и к расположенным на серверах Notes базам данных. Пользователи могут также сохранять персональные базы данных локально, т.е. на своих компьютерах. Станция Notes (клиент) имеет соединение с сервером (серверами) Notes по локальной сети (LAN) или с использованием модема.

DialUp-станция Notes имеет доступ к серверу Notes только посредством модема. Станции Notes не могут соединяться между собой. Сервер Notes тоже не может «по своей инициативе» соединиться со станцией — соединение с сервером всегда инициирует станция. Сервер может соединяться с другими серверами для выполнения репликаций баз, передачи почты или для предоставления «посреднических услуг» станциям или другим серверам.

Lotus Notes 6.0 может входить в состав других информационных систем как самостоятельный функциональный модуль.

Версия **Lotus Notes 6.0** поддерживает файлы и базы, созданные в предыдущих версиях, поэтому сделанные ранее наработки могут быть использованы без каких-либо затруднений.

Для комфортной работы в **Lotus Notes 6.0** на рабочей станции необходимы следующие вычислительные ресурсы:

- ✓ процессор не ниже 486;
- ✓ RAM 32 Мб и более;
- ✓ свободное дисковое пространство 40 Мб и более;
- ✓ CD-ROM 24—54-скоростной;
- ✓ поддержка протоколов SPX, NetBIOS/NetBEUI, TCP/IP, VINES, X.PC;
- ✓ операционные системы Solaris, Windows 95(98), Windows 2000x, Windows NT;
- ✓ мышь или другое устройство указания.

Наиболее существенное усовершенствование с точки зрения пользователя заключается в появлении:

- ✓ настраиваемых рабочих страниц;
- ✓ программных средств создания новых баз данных — Lotus Designer;
- ✓ программных средств управления работой в сетях — Lotus Administrator;
- ✓ закладок, упорядочивающих базы данных;
- ✓ кнопок задач и навигации;
- ✓ календарей групп;
- ✓ поддержки всех стандартов;
- ✓ новых параметров форматирования таблиц.

Программное обеспечение **Lotus Notes 6.0** может устанавливаться локально либо в сети совместно с **Domino**. Следует различать:

- ✓ функции программы **Notes**, переводящие базы данных **Notes** в связанные друг с другом WEB-страницы и обратно. **Domino** также позволяет пользователям Web-браузеров взаимодействовать с базами данных **Notes**;
- ✓ программное обеспечение сервера **Domino**, выполняющего задачи передачи почты и получения Web-страницы из Интернета, а также сохранения общих баз данных.

5.3. Корпоративные сети и соединения

Возможны следующие типы соединений станции с сервером или сервера с сервером:

✓ **Постоянное соединение по локальной сети (LAN)** (рис. 5.1). На станциях и серверах установлены сетевые карты и драйверы, обеспечивающие поддерживаемый сетевой протокол, например, NETBIOS, SPX, TCP/IP.

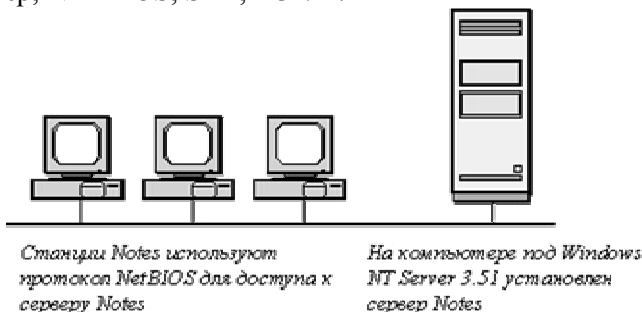


Рис. 5.1. Пример постоянного соединения по LAN

✓ **Временное соединение с другой LAN через Microsoft RAS** (рис. 5.2). Станция или сервер Notes инициируют RAS на установление необходимого соединения с компьютером в другой LAN. Это соединение обеспечивает поддержку протоколов NETBIOS, SPX (IPX) или TCP/IP. После установления соединения станция или сервер Notes работают по одному из сетевых протоколов с каким-то из серверов Notes в этой LAN.

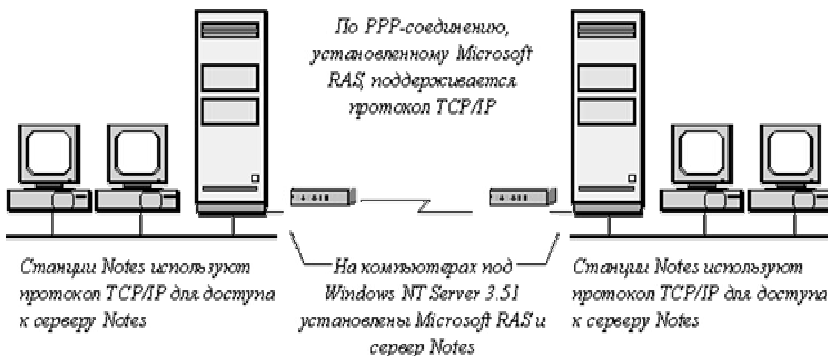


Рис. 5.2. Пример временного соединения с другой LAN через Microsoft RAS

✓ **Постоянное последовательное соединение** (рис. 5.3). Выделенная линия, необходимое оборудование и программное обеспечение (например, нуль-модемное соединение двух серверов или прямое соединение двух серверов по сети X.25.



Рис. 5.3. Пример постоянного последовательного соединения двух серверов

✓ **Временное последовательное соединение** (рис. 5.4). Обычная телефонная линия, модем и драйвер X.PC из поставки Notes.

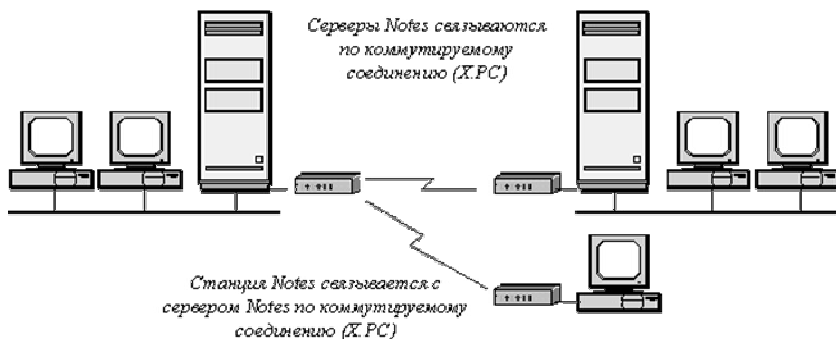


Рис. 5.4. Пример временного последовательного соединения

✓ **Соединение через сервер-посредник (PassThrough)** (рис. 5.5).



Рис. 5.5. Пример соединения с сервером через сервер-посредник

Уточним некоторые термины, которые будут встречаться в дальнейшем.

Файл-сервер — компьютер, на котором функционирует сетевая операционная система (например, Novell NetWare, Windows NT).

Мост/маршрутизатор — устройство для соединения двух или более локальных сетей в одну; осуществляет передачу пакетов между сетями. Обычно прозрачно для Notes.

Сервис удаленного доступа к локальной сети — программное и аппаратное обеспечение для доступа к локальной сети с использованием коммутируемых каналов связи (обычные телефонные линии, сети X.25, ISDN), но в отличие от мостов/маршрутизаторов на «временной основе». Такое соединение прозрачно для Notes, но его установление и разрыв могут инициироваться из Notes.

WAN — глобальная сеть, объединяющая многие локальные сети с использованием мостов, маршрутизаторов, выделенных линий, стандартных телефонных линий или иного специального оборудования для соединения компьютеров, обычно на больших расстояниях.

Для обслуживания Notes формально существуют четыре стандартных должности:

1. **Администратор системы** обслуживает серверы, занимается поддержкой пользователей, загружает и останавливает серверы, регистрирует серверы и сертификаторов верхних уровней, составляет и отлаживает расписание репликаций и передачи почты, распределяет дисковое пространство на сервере, следит

за происходящими в системе событиями по протоколу работы серверов, выполняет регламентные работы.

2. **Инженер поддержки** устанавливает и проверяет Notes, обеспечивает поиск ошибок. В большинстве случаев может поддерживать Notes в дополнение к поддержке других систем.

3. **Сертификатор** (Certifier) имеет и хранит от несанкционированного использования ID сертификатора, создает и сертифицирует ID-пользователей, используя при этом ID-сертификатор, а также предпринимает прочие меры безопасности.

4. **Администратор базы данных** управляет доступом пользователей к конкретной базе данных, размещенной на сервере.

5.4. Автоматизированная система ведения медицинской документации

При разработке программ, как правило, стараются автоматизировать в первую очередь именно деятельность врача и медицинского персонала, т.е. создавать программные продукты, позволяющие вести полную компьютеризированную «историю болезни».

При разработке информационной системы документооборота лечебного учреждения на базе любой системы документооборота (в том числе и Lotus Notes), основную сложность представляет реализация единой истории болезни (оговоримся, что все «административные» функции, так или иначе связанные с историей (расписание приема, списки амбулаторных посещений, госпитализации и т.п.) здесь не рассматриваются).

Понятно, что при принципиальной схожести историй болезни для врачей разных специальностей все же они будут существенно различаться. Более того, разные врачи одной и той же специализации одного отделения ЛПУ часто ведут истории болезни по-разному.

В то же время представляется важным, чтобы в основе МИС ЛПУ лежала единая история болезни.

В настоящее время в результате обобщения накопленного опыта разработан механизм ведения истории болезни (механизм шаблонов), позволяющий:

- изменять структуру истории болезни без изменения структуры базы данных и привлечения разработчика;

- хранить информацию об историях болезни в базе данных в единообразном виде — при существенно разном внешнем представлении истории.

Этот механизм близок механизму шаблонов MS Word по простоте использования и широким возможностям модификации, но при этом все введенные данные доступны для дальнейшего исследования, т.к. сохраняются в базе данных в едином виде (и при этом, при правильной организации шаблона, не в текстовом варианте).

Механизм ввода и хранения данных синтезирует два наиболее распространенных способа ввода информации в медицинских системах: простую диалоговую форму с фиксированными элементами ввода и редактируемый текстовый шаблон.

Ниже приведено краткое описание построенного механизма.

Первым шагом в решении данной проблемы было условное разделение истории болезни на две части: *«официальные данные»* и *«собственно история»*. К *«официальным данным»* можно отнести следующие поля: «дата рождения», «пол», «адресные данные», «образование», «специальность», «наследственный анамнез» и другие (в зависимости от специализации ЛПУ). Отличительной особенностью этих данных является их неизменность для каждого пациента; во всяком случае, эти изменения (например, изменение адреса) несущественны с медицинской точки зрения. К *«собственно истории»* отнесем все остальные данные.

«Официальные данные» вводятся в базу данных и хранятся «стандартно» (т.е. с использованием определенных разработчиком форм ввода и полей базы данных). Для ввода и хранения *«собственно истории»* был разработан описанный ниже **механизм шаблонов**.

В данном механизме используются следующие два приема:

1. Организация иерархического представления информации, хранящейся в истории. Историю болезни рассматривают как совокупность абстрактных понятий (**концепций**), объединенных в иерархическую древовидную структуру (**дерево концепций**). Важной особенностью «дерева концепций» является то, что каждый элемент может находиться в нескольких ветвях дерева, хотя в базе данных хранится только одна его копия. Практически это может быть полезно, например, в следующем случае: концепция

«Диагноз основной» должна входить в концепции «Госпитализация» и «Выписной эпикриз». Если продолжить аналогию с проводником Windows, то приблизительно эту способность можно представить как использование ссылок.

2. Динамическое создание форм для редактирования/просмотра разделов истории болезни.

Внешне окно для визуализации этого механизма выглядит как проводник Windows: в левой части формы отображается «дерево концепций», при перемещении по нему в правой части формы отображаются шаблоны, соответствующие выбранной концепции.

Другой особенностью данного подхода является то, что шаблон может быть сколь угодно подробным, — при вводе конкретной истории болезни достаточно заполнить только разделы, относящиеся к данному пациенту. В распечатку истории попадут только те разделы, которые были заполнены (если пользователь не определит иначе).

С технической точки зрения помимо организации структуры истории болезни и ее визуализации, данное «дерево» организует также разграничение доступа к данным. При входе в систему определяются полномочия пользователя, и при работе с историей он получает доступ только к тем разделам, на просмотр и изменение которых имеет право.

Концепции представляют собой логические понятия и служат для организации всего многообразия информации, хранящейся в истории болезни, в единую четкую структуру. Непосредственно же данные о конкретных пациентах отображаются и редактируются посредством шаблонов, в которых в режиме ввода (просмотра, редактирования) истории отображаются данные, соответствующие конкретной истории болезни конкретного пациента. Здесь же можно и редактировать эти данные (рис. 5.6).

Пациент: Тест Иван Иванович

Файл Записи Вкладки Справочники Справка

Паспортные данные Анализ Контакты

Ф.И.О.
Тест Иван Иванович

Пол ☐ Ж ☐ М

Дата рождения, возраст 01.01.1950 50

Семейное положение одинокий

Образование б/образования

Страховой анамнез
без больничного листа

Паспортные данные
Серия - Номер XXXX - 123456
Дата выдачи 01.01.1980
Выдан выдан 18-м отделением милиции г.

Адрес
Район Адмиралтейский
Улица 1-я Красноармейская ул.
№: 00 Корпус: 0 Квартира 00
Телефон

Организация
Тип орг-ции Здравоохранение Должность
Организация ВМА им.Кирова Лаборант

Условия работы, профпередности
не удовлетворительные
ТЕСТ условия работы, профпередности

Жилищно-бытовые условия
не удовлетворительные

☒ Декретированный
последний мед осмотр

☒ Донор
предыдущая кровосдача:

Рис. 5.6. Экранный шаблон заполнения истории болезни

Контрольные вопросы

1. Дайте определение корпоративных информационных систем.
2. Что дает использование корпоративных систем документооборота в практической деятельности медицинской организации?
3. Из каких частей состоит программное обеспечение Notes?
4. Какие должности существуют для обслуживания Notes?
5. Перечислите типы соединений станции с сервером или сервера с сервером.
6. Функции выполняет мост/маршрутизатор.
7. Поясните принципы ведения медицинской документации с помощью автоматизированной информационной системы.

6. Информационные системы на базах данных

6.1. Банки данных, базы данных, базы знаний

Банк данных — это комплекс информационных, технических, программных, языковых и организационных средств, обеспечивающих сбор, хранение, поиск и обработку данных, без чего невозможны создание, развитие и продуктивное использование информационной среды в любых отраслях деятельности. Банк данных предназначен для хранения больших массивов информации, быстрого поиска нужных сведений и документов. Создаются банки в абонентской системе любой производительности (от ПК до суперкомпьютера) и специализируются на сборе информации в определенных областях науки, технологии, продукции.

Банк поддерживается прикладными процессами, получающими в абонентской системе сервис области взаимодействия, благодаря которой банк взаимодействует с большим числом пользователей сети, а также с другими банками. Лица, работающие в банке данных, делятся на три группы: сотрудники банка (или программисты), администратор банка и пользователи.

Сотрудники банка осуществляют сбор и запись в базу всей первичной информации, определяемой тематикой этой базы, а также удаляют устаревшую информацию. Сотрудники банка и некоторые пользователи составляют программы, позволяющие из первичной информации получать необходимые вторичные сведения, составлять отчеты.

Администратор обеспечивает руководство банком, решает вопросы, связанные с бесперебойной и надежной работой, хранением информации и безопасностью данных, осуществляя копирование содержимого баз и организуя хранение копий.

Пользователи банка взаимодействуют с необходимыми им банками, формулируя запросы на получение информации в соответствии со своими профессиональными потребностями. В некоторых случаях или на контрактной основе им может быть разрешено обновление информации или составление вспомогательных программ.

Защита от несанкционированного доступа к базам создается за счет введения паролей и кодов, обеспечивающих идентификацию

пользователей. Формирование и ведение банков данных связано с большими затратами. Они становятся рентабельными лишь при большом **трафике**, т.е. при больших перерабатываемых и транслируемых в единицу времени объемах информации.

База данных — совокупность взаимосвязанных данных, организованная по определенным правилам. Строго говоря, базой данных являются специальным образом организованные группы файлов.

Файл — совокупность данных, выделенная в отдельный неделимый информационный продукт под собственным идентифицируемым именем, в расширении которого указан также вид используемого приложения.

Для работы с файлами используется **система управления базой данных** (далее — СУБД). При этом подразумевается, что база данных определена по схеме, не зависящей от программ, которые к ней обращаются. База данных характеризуется ее концепцией — совокупностью требований, обусловленных представлениями пользователей о необходимой им информации.

Для целенаправленного поиска информации в базах данных служат специально создаваемые информационно-поисковые системы.

Среди баз данных и СУБД различного назначения следует выделить направление, поддерживающее изучение и освоение тех или иных предметных областей в образовании и науке, так называемые базы знаний.

База знаний — сложная детально моделируемая структура информационных совокупностей, описывающих все особенности предметной области, включая факты (фактические знания), правила (знания условий для принятия решений) и метазнания (знания о знаниях), т.е. знания, касающиеся способов использования знаний и их свойств, а также сведения и инструкции, необходимые для освоения указанных знаний.

Наибольшее распространение получают **распределенные базы данных**, как это имеет место в мировой глобальной сети Интернет, услугами которой пользуются миллионы людей.

Существует много методов доступа к данным, находящимся в базах. Особой популярностью пользуется метод, определяемый **языком структурированных запросов (SQL)**. Все большее

распространение получают **аудио- и видео-базы**: в них размещаются, хранятся и выдаются тексты, звуки, неподвижные и движущиеся изображения.

Увеличение скорости обработки данных, рост объемов памяти, построение коммуникационных сетей с высокой пропускной способностью — все эти причины привели к использованию **полнотекстовых баз данных**. WWW обеспечивает универсальный доступ к большому числу баз данных, расположенных на различных континентах.

Увеличивается распространение **реляционных баз данных**, а также баз данных, имеющих **объектно-ориентированную архитектуру** и **многомерное представление данных** (ГИС — географические информационные системы). В этих базах создаются модули объектов, в том числе прикладных программ, которые управляются внешними событиями с помощью **графического интерфейса пользователя**.

6.2. Синтаксический, семантический, прагматический аспекты информационного процесса

В базе данных информация записывается и воспроизводится с помощью специально созданных лексических средств и на основе принятых синтаксических правил и ограничений.

Синтаксический анализ устанавливает важнейшие параметры информационных потоков, включая необходимые количественные характеристики, для выбора комплекса технических средств сбора, регистрации, передачи, обработки, накопления, хранения и защиты информации. Синтаксический анализ обслуживаемых информационных потоков обязательно предшествует всем этапам проектирования информационных систем.

Семантический анализ позволяет изучить информацию с точки зрения смыслового содержания отдельных элементов, находить способы языкового соответствия (язык человека, язык ЭВМ) при однозначном распознавании вводимых в систему сообщений.

Прагматический анализ проводится с целью определения полезности информации, используемой для управления, выявления практической значимости сообщений, применяемых для выработки управляющих воздействий.

Постоянная информация остается без изменений или же подвергается незначительным корректировкам в течение более или менее длительного периода времени. Это различные справочные сведения, нормативы, расценки и т.п.

Переменная информация отражает результаты выполнения производственно-хозяйственных операций, соответствует их динамизму и, как правило, участвует в одном технологическом цикле машинной обработки.

При вводе и обработке информации используются пакетный и интерактивные режимы.

Пакетный режим был наиболее распространен в практике централизованного решения экономических задач, когда большой удельный вес занимали задачи отчетности о производственно-хозяйственной деятельности экономических объектов разного уровня управления. Организация вычислительного процесса при пакетном режиме строилась без доступа пользователя к ЭВМ. Его функции ограничивались подготовкой исходных данных по комплексу информационно-взаимосвязанных задач и передачей их в центр обработки, где формировался пакет, включающий задание для ЭВМ на обработку, программы, исходные, нормативно-расценочные и справочные данные. Пакет вводился в ЭВМ и реализовывался в автоматическом режиме без участия пользователя и оператора, что позволяло минимизировать время выполнения заданного набора задач. В настоящее время **пакетный режим** реализуется в **электронной почте** или при массированных обновлениях баз данных.

Интерактивный режим предусматривает непосредственное взаимодействие пользователя с информационно-вычислительной системой, может носить характер запроса или диалога с системой.

Запросный режим необходим пользователям для взаимодействия с системой через значительное число абонентских терминальных устройств, в том числе удаленных на значительное расстояние от центра обработки.

Пример: Задача резервирования билетов на транспорте.

Информационная система реализует массовое обслуживание, работает в режиме разделения времени, при котором несколько независимых пользователей с помощью терминалов имеют в процессе решения своих задач непосредственный и практически

одновременный доступ к информационной системе. Этот режим позволяет дифференцированно в строго установленном порядке предоставлять каждому пользователю время для общения с системой, а после окончания сеанса выходить из нее.

Диалоговый режим открывает пользователю возможность непосредственно взаимодействовать с информационно-вычислительной системой в допустимом для него темпе работы, реализуя повторяющийся цикл выдачи задания, получения и анализа ответа.

6.3. Моделирование отношений *сущность/связь*. E/R-модели

Модель данных можно задать с помощью **спецификации вычислительной структуры**. В различных сферах хозяйственной деятельности встречаются однотипные данные. Примерами являются данные о наличии клиентов, товаров или сотрудников предприятия. Для таких больших множеств данных удобнее применять специальные способы описания.

При моделировании информационных систем (ИС) и баз данных (БД) с большим числом данных используется модель *сущность/связь* или **E/R (Entity Relationship Model)** модель (диаграмма). В такой модели данные представляются множествами элементов данных, а связи между основными элементами представляются через отношения.

ER-диаграммы (рис. 6.1) используются для разработки данных и представляют собой стандартный способ определения данных и отношений между ними.

Таким образом, осуществляется детализация хранилищ данных. ER-диаграмма содержит информацию о сущностях системы и способах их взаимодействия, включает идентификацию объектов, важных для предметной области (сущностей), свойств этих объектов (атрибутов) и их отношений с другими объектами (связей). Во многих случаях информационная модель очень сложна и содержит множество объектов.

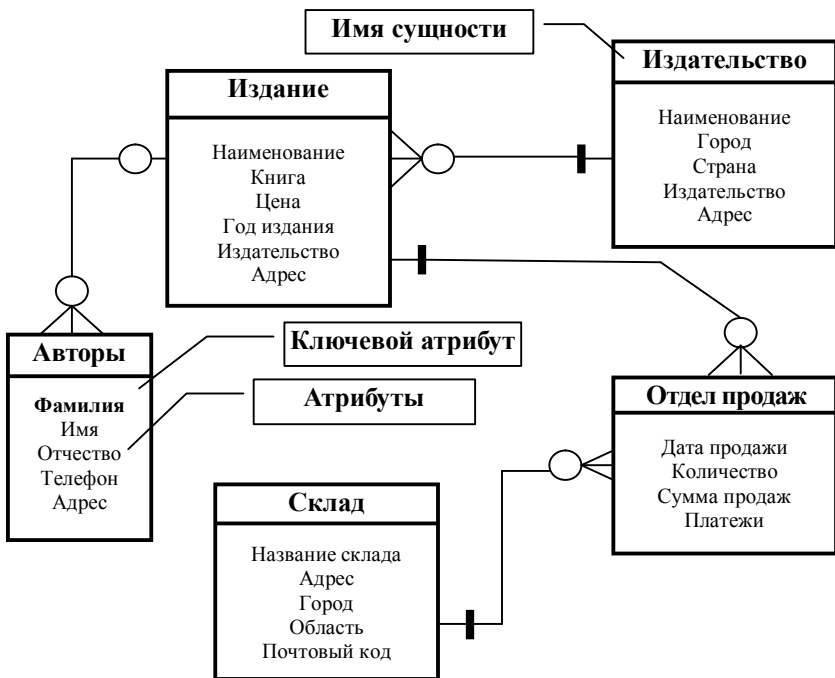


Рис. 6.1. Пример ER-диаграммы

Сущность изображается в виде прямоугольника, вверху которого располагается имя сущности (например, «Издание»), и дополнительно могут быть перечислены атрибуты сущности (ключевые указываются первыми в списке атрибутов). Отношение изображается линией между двумя сущностями с обязательным текстовым пояснением (рис. 6.2):

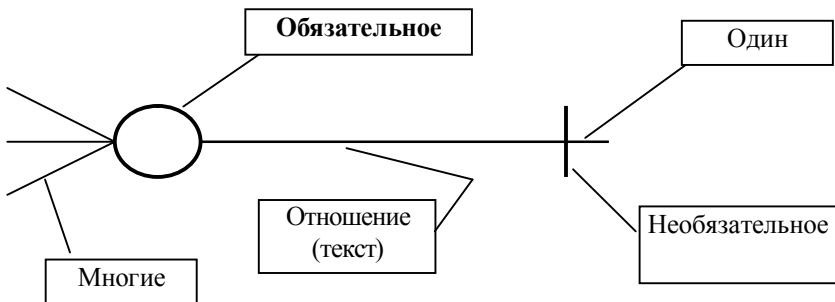


Рис. 6.2. Элемент ER-диаграммы

Например, для каждого «Издания» обязательно должен быть указан издатель в «Издательство», а один издатель в «Издательстве», может выпускать несколько наименований изданий в «Издания». Следует отметить, что связи всегда комментируются (надпись на линии, изображающей связь).

Рассмотрим пример (рис. 6.3) изображения рефлексивного отношения «сотрудник», где один сотрудник может руководить несколькими подчиненными и так далее вниз по иерархии должностей.

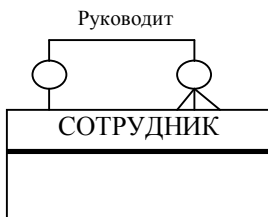


Рис. 6.3. ER-диаграмма рефлексивного отношения

Это отношение всегда является необязательным, в противном случае возникает бесконечная иерархия.

Если сущность имеет набор взаимоисключающих отношений с другими сущностями, то говорят, что такие отношения находятся в дуге. Например, банковский счет может быть оформлен или для юридического лица, или для физического. Фрагмент ER-диаграммы для такого типа отношений приведен на рисунке 6.4.

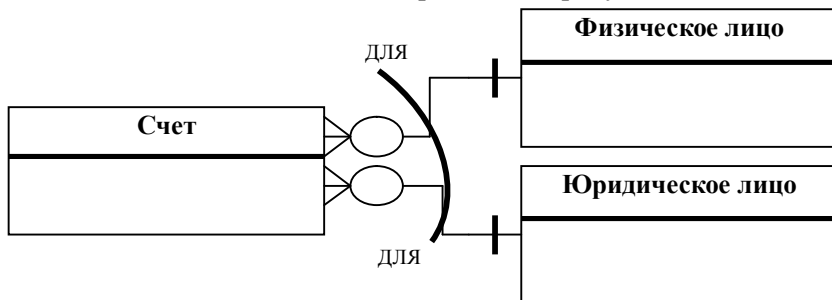


Рис. 6.4. Дуга

В этом случае атрибут ВЛАДЕЛЕЦ сущности СЧЕТ имеет особое значение для данной сущности — она делится на типы по категориям «для физического лица» и «для юридического лица».

Полученные в результате сущности называют подтипами, а исходная сущность становится супертипом. Чтобы понять, нужен супертип или нет, надо установить, сколько одинаковых свойств имеют различные подтипы. Следует отметить, что злоупотребление подтипами и супертипами является довольно распространенной ошибкой. Изображают их так, как показано на рисунке 6.5.

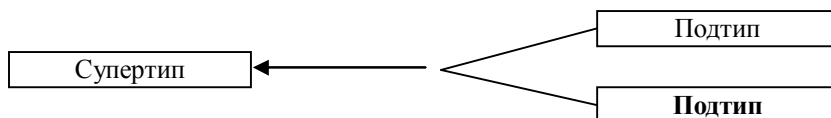


Рис. 6.5. Подтипы и супертип

Заданием E/R-модели характеризуется множество возможных состояний базы данных. В каждый момент своей жизни база данных обладает некоторым состоянием. С помощью определенных операций можно осуществить опрос состояния базы и, соответственно, изменить это состояние.

Как правило, запросы к базе данных и изменения ее состояния выражаются через мощные операции над сущностями и связями.

Типичный запрос к базе данных заключался бы в поиске: «найди фамилии и адреса всех сотрудников отделов x, y, z по специальностям a, b, c».

Такой запрос доставляет список, представляющий собой одноместное отношение, который затем может быть еще обработан с определенной точки зрения (например, просмотрен или распечатан). Языки запросов к базам данных предусматривают вполне определенный формат и логическую форму.

Аналогичным образом могут быть сделаны и изменения состояний и баз данных. Эти изменения могут включать следующие действия:

1. Внести новые сущности, воплощения и новые элементы связей.
2. Удалить определенные данные.
3. Изменить определенные компоненты.

Для формулирования запросов к базе данных и требований на изменение ее содержимого базы данных пользователю предоставляют **особый интерфейс**.

Отметим, что в самом общем случае **интерфейсом** называется описание **взаимодействия**.

Особый интерфейс состоит из таких команд, как «выбор», «исключение» или «добавление данных» и из (ограниченных) возможностей задавать логические формулы для образования подмножеств данных.

Для этого используются такие аспекты алгебры отношений (реляционной алгебры), как:

- 1) проекция (вычеркивание определенных компонент кортежей в отношении);
- 2) соединение (произведение отношений, причем делается соединение кортежей, совпадающих в общих атрибутах);
- 3) селекция (выбор кортежей, удовлетворяющих заданному условию относительно значений атрибутов).

6.4. Обобщенная архитектура СУБД

Основной интерес применения СУБД заключается в том, чтобы предложить пользователю (или прикладным процессам) абстрактное представление данных, скрыв особенности хранения и управления ими. СУБД должна представлять доступ к данным посредством прикладных процессов любым пользователям, в том числе не имеющим представления:

- о физическом размещении в памяти данных и их описаний;
- о механизме поиска запрашиваемых данных;
- о проблемах, возникающих при одновременном запросе одних и тех же данных большим числом пользователей (прикладных программ);
- о способах защиты данных от некорректных обновлений и (или) несанкционированного доступа;
- о поддержании базы данных в актуальном состоянии.

Обобщенная структура связей программ и данных при использовании СУБД показана на рисунке 6.6.

При выполнении основных из этих функций СУБД должна использовать различные описания данных, в которых должны быть учтены:

- сущности интересующей предметно области;
- атрибуты, характеризующие неотъемлемые свойства каждой сущности;

— связи, ассоциирующие выделенные сущности.

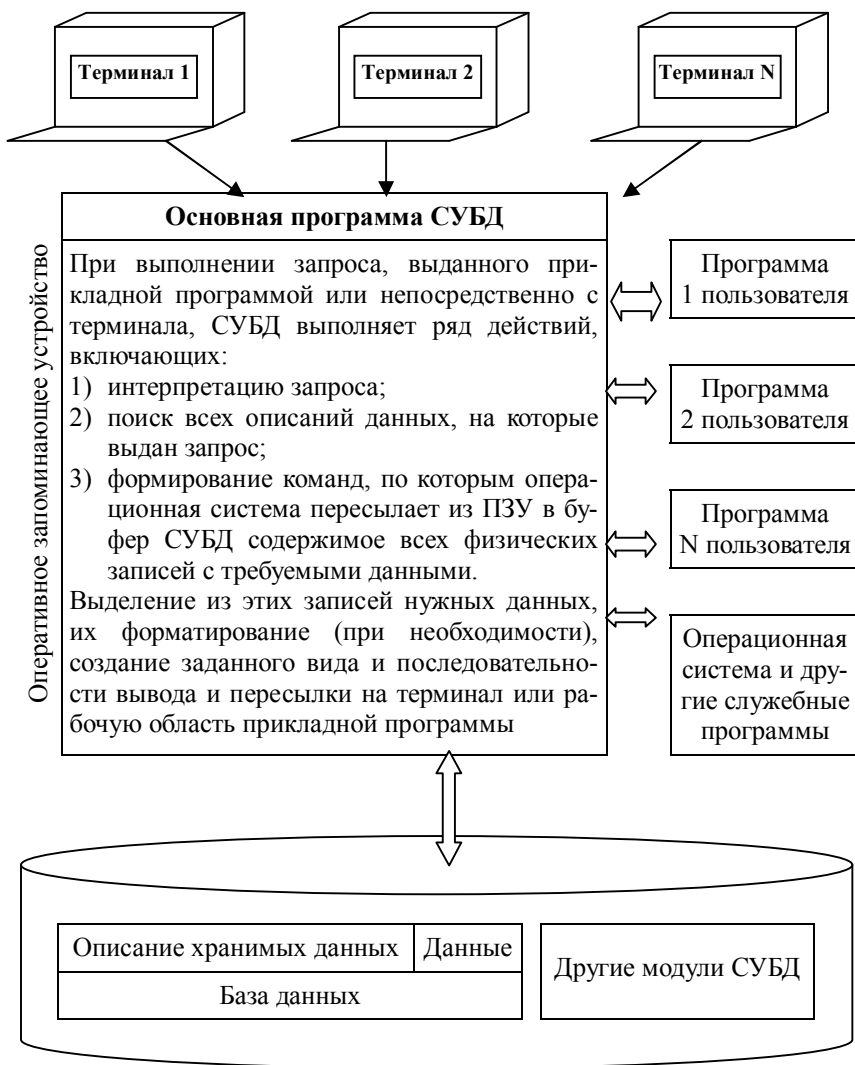


Рис. 6.6. Обобщенная структура связей программ и данных в СУБД

Подход к архитектуре и описанию данных предложен комитетом ANSISPARC (Комитет планирования Стандартов и норм

национального института стандартизации США) и имеет целью отделение пользовательского представления о БД от ее физической организации. В соответствии с этим предложением архитектура СУБД должна быть трехуровневой и должна содержать внешний, концептуальный и внутренний уровни (рис. 6.7).

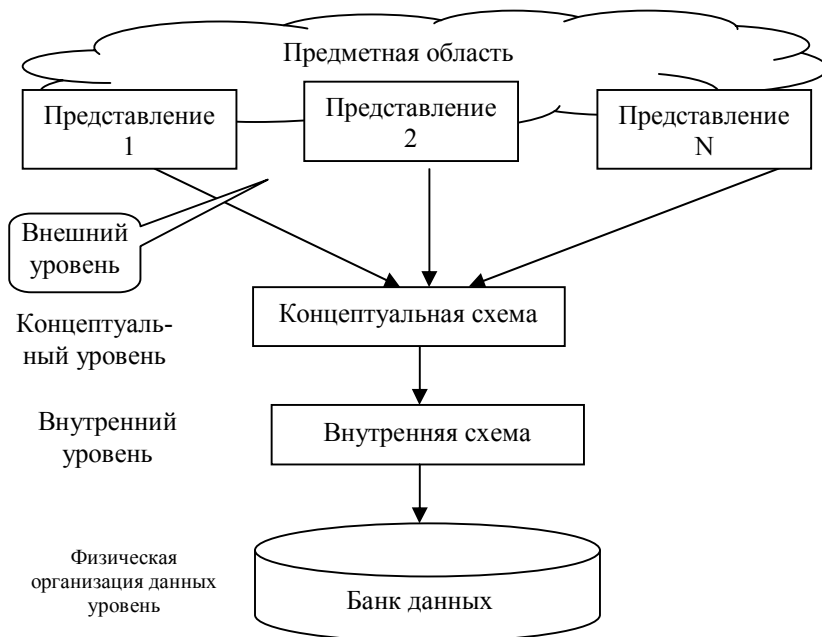


Рис. 6.7. Трехуровневая архитектура ANSI-SPARC

Такое отделение по уровням обеспечивает **независимость** хранимых данных, что необходимо по следующим причинам:

1. Каждый пользователь должен иметь возможность обращаться к данным, используя свое представление о них, изменять свое представление о данных без влияния на представление других пользователей.
2. Пользователи не должны иметь дело с деталями физической организации данных, т.е. не должны зависеть от особенностей физического хранения данных.

3. Администратор базы данных должен иметь возможность изменять структуру хранения данных так, чтобы эти изменения оставались прозрачными для остальных пользователей.

4. Структура БД не должна зависеть от физических аспектов хранения, например, при переключении на новое внешнее устройство памяти.

Внешний уровень — это представление БД с точки зрения конкретных пользователей. Указанный уровень может включать несколько различных представлений БД со стороны различных групп пользователей. При этом каждый пользователь имеет дело с представлением предметной области, выраженным в наиболее понятной и удобной для него форме. Такое представление содержит только те сущности, атрибуты и связи, которые интересны ему при решении профессиональных задач. Различные представления на внешнем уровне могут пересекаться, т.е. использовать общие описания абстракций предметной области. На внешнем уровне создается **инфологическая модель** БД, полностью независимая от платформы (вычислительной системы, на которой будет использоваться).

Инфологическая модель является человеко-ориентированной: средой ее хранения может быть память человека, а не ЭВМ. Инфологическая модель не изменяется до тех пор, пока какие-то изменения в реальном мире не потребуют изменения некоторых определений, чтобы эта модель продолжала отражать предметную область.

Концептуальный уровень — обобщающее представление БД, описывающее то, какие данные хранятся в БД, а также связи, существующие между ними. Концептуальный уровень является промежуточным в трехуровневой архитектуре. Содержит логическую структуру всей базы данных. Фактически это полное представление требований к данным со стороны организации, которая не зависит от соображений относительно способов их хранения. На концептуальном уровне необходимо выделить:

- 1) сущности, их атрибуты и связи;
- 2) ограничения, накладываемые на данные;
- 3) семантическую информацию о данных;
- 4) информацию о мерах обеспечения безопасности.

Концептуальный уровень поддерживает каждое внешнее представление. Поэтому на этом уровне содержатся любые доступные пользователю данные, за исключением сведений о методах хранения этих данных. На концептуальном уровне создается даталогическая модель.

Даталогическая модель представляет собой описание инфологической модели на языке определения данных конкретной СУБД. Эта модель является компьютерно-ориентированной (зависит от применяемой на компьютере СУБД и операционной системы).

Внутренний уровень — физическое представление БД, описывающее методы ее хранения в вычислительной системе. Данный уровень описывает физическую реализацию БД и предназначен для достижения оптимальной производительности и обеспечения экономного использования дискового пространства. Содержит описание структур данных и отдельных файлов, используемых для хранения на запоминающих устройствах.

На внутреннем уровне осуществляется взаимодействие СУБД с методами доступа операционной системы с целью эффективного размещения данных на носителях, создания индексов и т.д. На этом уровне используются:

- карта распределения дискового пространства для хранения данных и индексов;
- описание подробностей сохранения записей (с указанием реальных размеров сохраняемых элементов данных);
- сведения о размещении записей;
- сведения о возможности сжатия данных и выбранных методов шифрования.

Реализация перечисленной информации производится на **физическом уровне** вычислительной системы, который контролируется операционной системой. В настоящее время функции СУБД и операционной системы строго не разграничиваются. Имеются СУБД, использующие все методы доступа, предусмотренные в данной операционной системе, в других используются только некоторые из методов либо реализована своя файловая система.

На внутреннем уровне создается физическая модель.

Физическая модель дает возможность СУБД представлять программам и пользователям доступ к хранимым данным по

именам, не заботясь об их физическом расположении. Является компьютерно-ориентированной (зависит от СУБД и операционной системы). По этой модели СУБД отыскивает данные на внешних запоминающих устройствах.

В соответствии с предложениями ANSI-SPARC можно представить соответствующую трехуровневую модель данных (рис. 6.8):



Рис. 6.8. Уровни моделей данных

СУБД строится по модульному принципу, т.к. это достаточно сложный программный продукт. Состав и взаимосвязь модулей реальных СУБД сильно различаются, однако можно выделить ряд обобщенных компонентов (рис. 6.9):

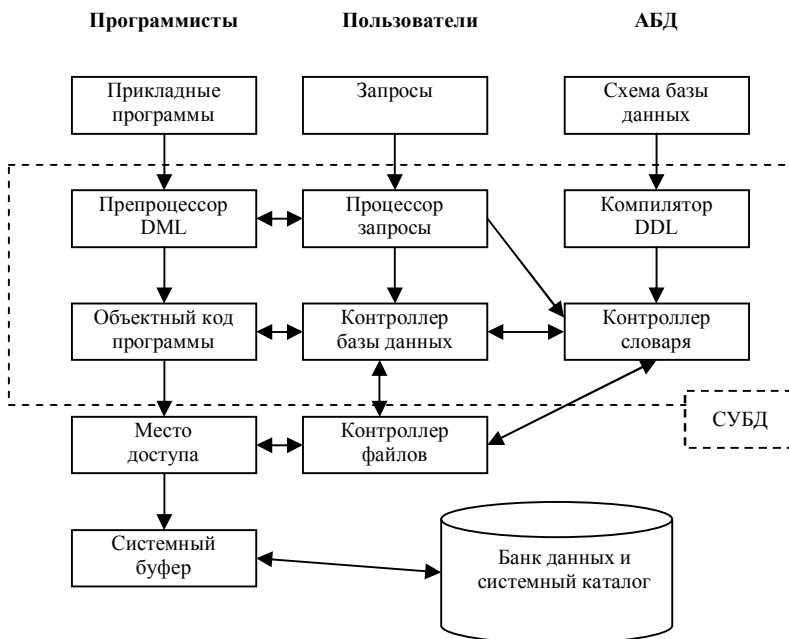


Рис. 6.9. Основные компоненты СУБД

Рассмотрим основные компоненты СУБД.

Процессор запросов — основной компонент СУБД, преобразующий запросы в последовательность низкоуровневых инструкций для контроллера БД.

Контроллер базы данных взаимодействует с запущенными пользовательскими программами и запросами. Контроллер БД принимает запросы и проверяет внешние и концептуальные схемы для определения тех концептуальных записей, которые необходимы для удовлетворения требований запроса. Контроллер БД вызывает контроллер файлов для выполнения поступившего запроса.

Контроллер файлов манипулирует предназначенными для хранения данных файлами и отвечает за распределение доступного дискового пространства. Он создает и поддерживает список структур и индексов, определенных во внутренней схеме. (Контроллер файлов не управляет физическим вводом — это функция операционной системы).

Препроцессор языка DML преобразует внедренные в прикладные программы DML операторы в вызовы стандартных функций базового языка. Для генерации соответствующего кода препроцессор языка DML должен взаимодействовать с процессором запросов. Язык манипулирования данными (Data Manipulation Language — DML) — совокупность языковых средств организации доступа к данным в некоторой модели данных и в соответствующих ей СУБД.

Компилятор языка DDL преобразует DDL команды в набор таблиц, содержащих метаданные. Затем эти таблицы сохраняются в системном каталоге, а управляющая информация — в заголовках файлов с данными. Язык определения данных (Data Definition Language — DDL) — формальный закон, используемый в некоторой модели данных для определения структуры данных.

Контроллер словаря управляет доступом к системному каталогу и обеспечивает работу с ним. Системный каталог доступен большинству компонентов СУБД.

Контроллер базы данных, как и другие компоненты СУБД, содержит большое количество программных единиц. Например, в контроллер базы данных входят следующие программные компоненты (рис. 6.10):

1. Контроль прав доступа (пользователь имеет или не имеет права на затребованную операцию).
2. Процессор команд (для выполнения запроса управление передается процессору команд).
3. Средства контроля целостности (при выполнении операций, связанных с изменением содержимого базы данных, проверяется целостность ключей).
4. Оптимизатор запросов определяет наилучшую стратегию выполнения запроса.
5. Контроллер транзакций осуществляет требуемую обработку операций, поступающих в процессе выполнения транзакций.
6. Планировщик отвечает за бесконфликтное выполнение параллельных с базой данных операций.
7. Контроллер восстановления обеспечивает восстановление данных до непротиворечивого состояния после сбоя.
8. Контроллер буферов отвечает за перенос данных между ОЗУ и внешними носителями.

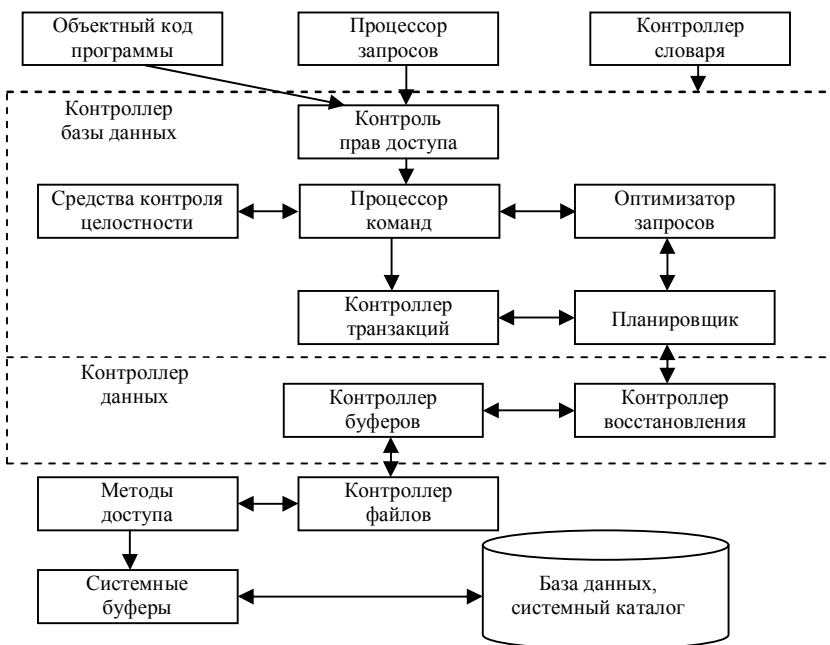


Рис. 6.10. Компоненты контроллера базы данных

6.5. Архитектура системы медицинской информационной системы на основе базы данных

Рассмотрим многоуровневую модель медицинской информационной системы (рис. 6.11 и 6.12).

Опишем назначение и особенности каждого уровня ИС.

1. **Уровень рабочей станции.** При проектировании системы на этом уровне определяются параметры компьютера рабочей станции — минимальные и рекомендуемые технические характеристики (процессор, объем оперативной и постоянной памяти и т.д.). На этом же уровне задаются эргономические требования к компьютеру, т.к. работа с ним прямо влияет на здоровье пользователя. Список поддерживаемых операционных систем зависит от степени мультиплатформенности ИС. Поэтому на решение об использовании той или иной операционной системы (далее — ОС) влияет и специальность пользователя. Перечислим основные факторы, определяющие выбор ОС для рабочих станций пользователей:

- список ОС, поддерживаемых ИС;
- надежность;
- доступность в использовании;
- легкость в освоении;
- цена;
- удобство в администрировании (значимость этого фактора растет с числом клиентов в сети).

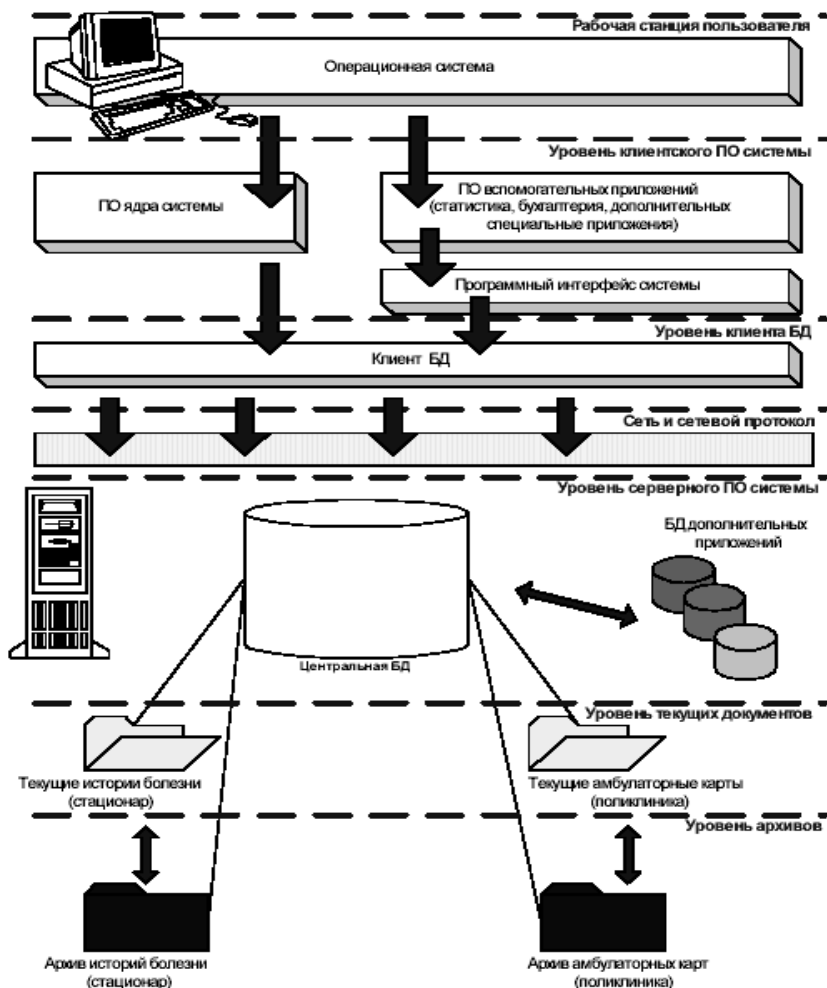


Рис. 6.11. Многоуровневая модель системы

2. Уровень клиентского ПО системы — приложения системы, с которыми осуществляет свою работу пользователь. Здесь мы разделяем все клиентское ПО на две группы: основное и вспомогательное. При этом для функционирования вспомогательного ПО добавляется новая прослойка — программный интерфейс системы, т.е. набор специального ПО, которое принимает вызовы от вспомогательного приложения, транслирует его в запросы к серверу БД и, используя средства клиента, отправляет их серверу.

Целями такого разделения стали следующие причины.

- Уменьшение количества функций, которые встраиваются в ядро клиентского ПО. Суть этого требования состоит в том, что основное ПО клиента должно уметь делать только основные или наиболее часто востребованные функции. При этом качество разработки и трудозатраты программиста самые высокие.

- Увеличение производительности работы основного ПО. Это требование совершенно логично. Самые востребованные функции клиент должен выполнять с самой высокой производительностью. За счет этого происходит экономия времени работы с компьютером, повышение эффективности использования системы и т.д. Только с этой целью основное клиентское ПО может обращаться к серверу БД напрямую.

- Увеличение гибкости системы. Не секрет, что адаптация ИС к нуждам конкретного лечебного учреждения требует зачастую участия разработчиков самой системы. ПО сервера и клиента заказчик закупает и устанавливает сам. При инсталляции системы устанавливается основное ПО (которое изменить может только разработчик системы), вспомогательное ПО и детальное описание интерфейса системы — вызовов библиотек, стандартных функций системы по поиску данных, доступу к историям болезни и т.д. Если заказчика не устраивает функциональность системы или если ему требуются дополнительные возможности, не предусмотренные разработчиком, он либо может заказать такое ПО у разработчика системы (оно будет выполнено в виде дополнительного приложения и не потребует переустановки или настройки основного ПО), либо может разработать его сам.

- Уменьшение трудозатрат программиста на создание нового ПО или на внесение изменений в существующие приложения.

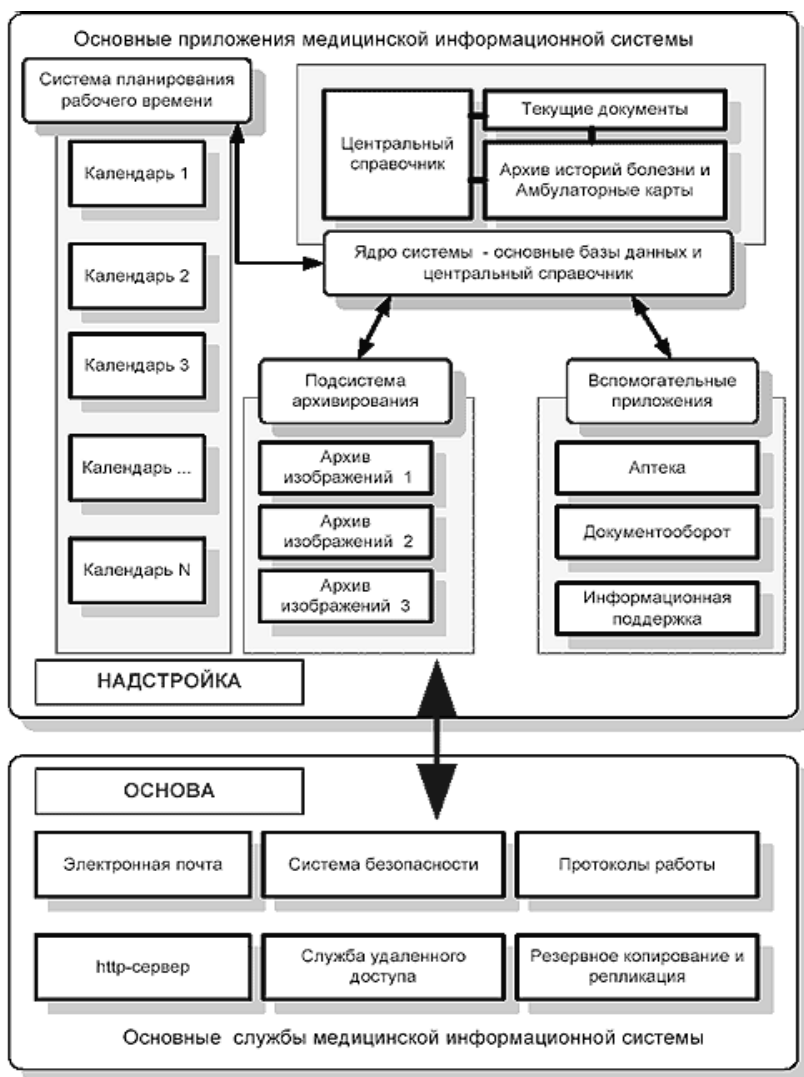


Рис. 6.12. Архитектура медицинской информационной системы

К основному ПО относим те приложения, которые врачи используют в своей работе для выполнения своих основных функциональных обязанностей (для врача — это работа с амбулаторными картами и историями болезни).

К дополнительному ПО отнесены вспомогательные приложения для всех медицинских работников и остальное ПО прочих сотрудников, таких как бухгалтеры, специалисты по медицинской технике и т.д. Сюда же мы относим программы, которые могут использоваться специалистами в повседневной работе (например, справка о движении больных), но которые в силу специфичности задачи или невысоких требований к производительности работы могут быть выполнены в виде дополнительных приложений.

К программному интерфейсу системы мы относим специальное ПО, инкапсулирующее в себе возможности клиента базы данных и взаимодействующее уже не с сервером базы данных, а с системой. Например, в случае распределенной БД поиск имеющихся в системе архивных историй болезни является нетривиальной задачей. Конечно, можно встроить этот алгоритм в саму программу. А что если ядро системы или архитектура ее будут изменены? Вносить изменения во все имеющиеся программы медицинской информационной системы? А если таких приложений десятки, а их инсталляций сотни? Именно поэтому целесообразнее переносить все прямые запросы к БД в промежуточное ПО. В случае надобности только в программный интерфейс вносятся изменения, а готовые и установленные приложения не потребуют переделки и переустановки. Кроме этого, при использовании такого ПО значительно увеличивается скорость разработки новых программ. Не секрет, что значительное число функций даже нового приложения содержит однотипные действия: подключиться к серверу, осуществить поиск, обработать ответ и т.д. Используя уже готовые и отлаженные модули из программного интерфейса системы, программист экономит время. Примером эффективного способа для создания программного интерфейса является использование в среде Windows динамических подключаемых библиотек (dynamic link libraries — DLL).

3. Уровень клиента БД. Клиент БД осуществляет диалог с основным и дополнительным ПО, выполняет проверку пароля пользователя и, самое главное, формирует запросы к серверу БД, осуществляя с ним диалог. Работа основного ПО у некоторых БД осуществляется прямо в клиенте (Lotus Notes), обеспечивая интерфейс пользователя, справку, дополнительные возможности типа электронной почты и т.д. Работа других клиентов (InterBase)

состоит только в обеспечении программной возможности доступа к серверу БД.

4. **Сеть и сетевой протокол.** Клиент БД, обращаясь с запросом к серверу, должен передать его по сети. При этом он должен поддерживать хотя бы один сетевой протокол. На этом уровне учитывается список поддерживаемых сетевых протоколов клиентов и сервера и решается вопрос об использовании конкретного или нескольких протоколов. Это решение влияет на производительность системы в целом, на возможности ее масштабирования, на простоту и надежность в работе и управлении. Наиболее распространенным на сегодня является протокол TCP/IP, используемый в сегментированных сетях и Интернете. Мы также считаем его предпочтительным при выборе. Кроме того, этот уровень является еще и разделительным — все последующие уровни относятся к серверу БД.

5. **Уровень серверного ПО системы.** Здесь мы не разделяем уровень на ПО сервера БД и прикладное ПО, расширяющее возможности сервера для обеспечения задач ИС. На этом уровне осуществляется принятие запросов от клиентов БД, выполнение и отправка ответов. В случае программно распределенной БД возможен вариант, когда не клиент определяет, к какой конкретно БД обратиться, а центральный сервер системы.

6. **Уровень текущих документов.** Небольшие по объему БД, которые содержат только текущие документы, позволяют осуществлять максимально производительную работу системы.

7. **Уровень архивов.** На этом уровне находятся архивные БД, производительность работы с которыми может быть низкой, однако потребность в документах этого уровня не такая высокая, как с текущими документами. Как правило, в архивах осуществляется только поиск документов или расчет статистики. Необходимое условие — отсутствие прав на запись или удаление документов с целью сохранности информации. Этот уровень может быть физически расположен на отдельном компьютере для разгрузки сервера текущих документов.

6.5.1. Безопасность информационной системы

Вся информация, собираемая в течение жизни пациента, разделяется с точки зрения доступа к ней сотрудников медицинского учреждения. Этот процесс регламентируется специальным разделом медицины — деонтологией. Медицинские информационные системы отличаются прежде всего тем, что в них хранится и обрабатывается информация, всесторонне определяющая социальный статус человека, а это обуславливает особую форму отношений между теми, кто ее формирует, и теми, кто использует. Значит, наряду с повышенными требованиями к достоверности информации, должны накладываться нравственные ограничения на доступ к ней, а также юридическая ответственность на предоставляющих ее лиц. Любой медицинский работник несет полную ответственность (моральную, административную и уголовную) за конфиденциальность информации, к которой он получает доступ в ходе своей профессиональной деятельности.

Наиболее серьезными причинами для использования мощной системы безопасности являются следующие причины:

1. Случайные или намеренные действия зарегистрированных пользователей информационных систем. Для защиты от этого фактора применяются электронные карточки, автоматическое блокирование рабочих станций, регулярные смены паролей, лимит на рабочее время, в течение которого пользователь может получить доступ к системе, и т.д.

2. Компьютерные вирусы. Основными способами борьбы здесь являются: полное запрещение использования непроверенных дисков, компакт-дисков или других носителей информации, запрет на использование Internet или непроверенной электронной почты на рабочих станциях системы, использование современных антивирусов и постоянное их обновление.

3. Использование сетей передачи данных может быть применено с целью перехвата или модификации данных, заполнению сети ложными пакетами информации и другими действиями, направленными на выход из строя сетевого оборудования или серверов. Охрана территорий, на которых расположены средства связи (информационный кабель, усилители сигнала — репитеры и т.д.), имеет высокую важность. Мы не будем затрагивать вопросы

использования аппаратных методов защиты компьютеров и их сетей, таких как брандмауэры (межсетевые экраны), проху-серверы, smart-карты и т.д. Детально остановимся на подсистеме безопасности ИС, которая должна включать в себя следующие функции:

- средства распределения прав доступа гарантируют возможность получения доступа только к той информации и программам, которые необходимы для выполнения функциональных обязанностей. При этом во всех отношениях «Пользователь — Система — Информация» роль координатора выполняет администратор системы, имеющий максимально возможные права доступа; главное из них право — возможность предоставлять или ограничивать доступ всех остальных пользователей и других служащих к системе;

- средства протоколирования использования БД позволяют четко контролировать доступ к ресурсам ИС, попытки несанкционированных действий или превышение пользователями своих прав;

- средства шифрования БД и сетевого трафика позволяют гарантировать, что во время передачи информации по каналам связи в случае ее перехвата она не сможет быть прочитана «злоумышленником»;

- средства сокрытия кода приложений ИС гарантируют, что попытка чтения кода приложений ИС невозможна или крайне затруднена. Эта необходимость существует из соображений, что в случае анализа исходного кода могут быть обнаружены слабые места в системе безопасности, которыми злоумышленник сможет воспользоваться.

6.5.2. Распределение прав доступа

Все современные БД позволяют в той или иной степени разделять права доступа пользователей к информации. Опишем принципиальную работу пользователя с точки зрения проверки прав доступа.

1. Началом работы является аутентификация. При этом используется учетная запись пользователя, которая образуется открытым именем пользователя и зашифрованным паролем. Учетная запись каждого пользователя хранится в определенном виде в ИС.

Например, Windows NT использует так называемую базу данных учетных записей, которые хранятся на контролерах домена NT. Lotus Notes хранит учетную запись пользователя в виде id-файла, который может помещаться на сервере, локально на рабочей станции пользователя или на дискете. При выполнении процедуры идентификации пользователь вводит свое имя и пароль, которые затем проверяются системой на правильность.

2. В случае выполнения первого шага для пользователя открывается сеанс связи с сервером. Все последующие действия выполняются от имени учетной записи, для которой он успешно ввел пароль.

3. При доступе к БД сервер проверяет, имеются ли у текущего пользователя права на ту информацию, которую он запрашивает. В случае положительного решения информация предоставляется пользователю. В случае отрицательного решения пользователю отказывается в доступе.

Итак, для каждого объекта медицинской информационной системы должен быть задан список, согласно которому сама ИС будет проверять, имеет ли данный пользователь право на доступ к этому объекту или нет. В литературе и на практике чаще всего такой список называется ACL (от англ. Access Control List — список управления доступом). Основными объектами ИС, для которых в обязательном порядке должен быть задан ACL, являются:

- информация, хранящаяся в БД;
- приложения, входящие в состав пакета программ системы;
- команды и функции в приложениях, которые могут использоваться с различным уровнем доступа.

Список управления доступом к информации в БД ИС должен указываться для:

- конкретного вида информации (поля);
- определенной области документа (например, статистической информации);
- отдельно взятого документа;
- отдельно взятой группы документов, относящихся к одному виду (например, к виду заключений лабораторной диагностики);
- отдельной базы данных, входящей в состав ИС;
- сервера.

При этом порядок проверки должен осуществляться от большого к малому, т.е. вначале необходимо пройти проверку на разрешение доступа к амбулаторной карте, затем к ее документам и т.д.

ACL объектов хранятся постоянно. Они генерируются одновременно с созданием документа несколькими способами:

- 1) вычислением на основании политики доступа или иного алгоритма;
- 2) наследованием от его предка (например, документ истории болезни может наследовать (копировать) ACL бланка истории болезни).

Изменение ACL возможно несколькими способами:

- 1) программным изменением, когда изменить ACL объекта может только приложение системы;
- 2) пользовательским изменением, когда пользователь вручную может корректировать ACL объекта.

Реакция системы на изменение ACL может быть различной. Изменение ACL сервера, каждой отдельно взятой БД или приложений, некоторых документов или отдельных полей должны протоколироваться в обязательном порядке. Администратор системы должен тщательно следить за протоколами этих изменений.

Реакция на проверку наличия текущего пользователя в ACL может быть в виде:

- скрытого отказа, когда пользователь просто «не видит» тех объектов, к которым он не имеет права доступа, — такая реакция может быть у приложения, которое динамически строит свое меню в зависимости от прав пользователя;
- непротоколируемого отказа, когда пользователю, пытающемуся получить доступ к объекту, выводится сообщение о том, что ему в доступе отказано, — такой отказ системой никак не протоколируется;
- протоколируемого отказа, когда этот факт система протоколирует в специальном журнале;
- отказ с извещением, когда пользователю в доступе отказывается и система извещает администратора или другое доверенное лицо о факте такого отказа, например, по электронной почте.

Контрольные вопросы

1. Поясните назначение банка данных.
2. Поясните назначение базы данных.
3. Поясните назначение базы знаний.
4. Перечислите функции семантического анализа.
5. Перечислите функции синтаксического анализа.
6. Перечислите функции прагматического анализа.
7. Какой режим обработки данных называется пакетным?
8. Какой режим обработки данных называется интерактивным?
9. Какой режим обработки данных называется запросным?
10. Какой режим обработки данных называется диалоговым?
11. Поясните назначение ER-диаграмм.
12. Поясните обобщенную структуру связей программ и данных при использовании СУБД.
13. Поясните трехуровневую архитектуру СУБД.
14. Что называется инфологической моделью?
15. Что называется даталогической моделью?
16. Что называется физической моделью?
17. Перечислите основные компоненты СУБД.
18. Перечислите компоненты контроллера базы данных.
19. Приведите пример многоуровневой модели медицинской информационной системы.

7. Распределенные информационные системы

7.1. Internet/Intranet-технологии

Осознание мировым сообществом роли информации как стратегического ресурса стимулировало разработки новых информационных технологий для получения, переработки, хранения и предоставления больших объемов информации пользователям. Первое место среди новых технологий занимают сетевые информационные технологии, открывающие принципиально иные возможности построения сбалансированного общества, с существенно большей реализацией индивидуальных возможностей его членов. «Информационное общество» несет в себе огромный потенциал для улучшения жизни граждан и повышения эффективности социального и экономического устройства государства.

Использование Internet/Intranet-технологий при создании информационных ресурсов и построении информационных систем различного назначения в ближайшее время станет доминирующим в мировом информационном пространстве по следующим причинам:

- 1) эти технологии позволяют организовать с достаточной простотой для пользователя системы поиска нужной информации;
- 2) предъявляют минимальные требования к рабочему месту клиента: клиент работает со стандартным программным обеспечением, а единственным требованием является поддержка работы WWW браузеров последних версий;
- 3) поддерживают распределенные системы хранения информации и множественные методы ее хранения;
- 4) поддерживают работу с практически неограниченным объемом разноплановых данных (текст, графика, изображение, звук, видео, векторные карты и др.);
- 5) предоставляют технологически простой способ администрирования информационных систем с одного рабочего места;
- 6) поддерживают удаленные методы редактирования и пополнения информации.

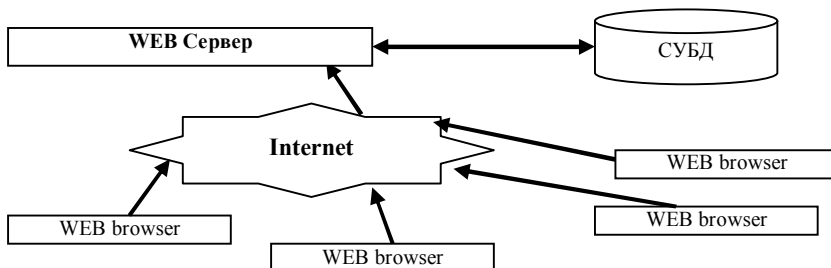


Рис. 7.1. Взаимодействие с БД через Интернет

Основой построения информационных систем с использованием Intranet-технологии является организация системы доступа к информации через WWW сервис Internet. Internet-технология позволяет оперативно управлять и актуализировать информацию, хранящуюся в базах данных через просмотрщик (браузер) WWW страниц (рис. 7.1).

Основной принцип, заложенный в Intranet-технологии создания информационных ресурсов и построения информационных систем, заключается в разделении вычислительных ресурсов как между многочисленными серверами, расположенными в различных концах сети, так и между серверами и клиентами.

Реализация этого принципа основана на использовании:

- **HTTP-SQL** (формирование SQL-запросов к БД с WWW сервера);
- **API** (организация динамических приложений на стороне сервера);
- **Java Script** (организация динамических приложений на стороне клиента) интерфейсов для формирования запросов пользователя к базам данных или к другим информационным источникам на получение и обработку данных.

Internet-технология позволяет удачно сочетать возможности гипертекстового оформления и возможности современных СУБД. Причем со стороны клиента полностью унифицируются запросы на поиск и представление информации, а также на получение аналитических справок и данных из информационных систем.

Вместе с тем рассматриваемые технологии позволяют использовать в сетевом режиме уже имеющиеся базы данных, не затрачивая при этом средства на их унификацию и приведение к единому

стандарту. Основные затраты здесь направлены только на соответствующие описания баз данных и запросов для HTTP-SQL интерфейса или для сервера обработки транзакций. Использование данной технологии позволяет решать весь спектр задач, присущий информационной системе, включая удаленный ввод и редактирование данных.

Математическое обеспечение для организации HTTP-SQL интерфейса является свободно распространяемым как для MS Windows NT систем, так и для некоммерческих UNIX платформ.

Можно использовать либо имеющиеся в наличии СУБД, либо приобретать сетевые (например, Informix, Oracle, MS SQL).

Любая информационная система, построенная на основе клиент-серверных Интернет-технологий, должна содержать следующие серверные компоненты:

- шлюз-сервер, управляющий правами доступа к информационной системе;
- WWW-сервер;
- сервер баз данных;
- сервер приложений и (или) сервер обработки транзакций.

Взаимодействие WWW сервера с базами данных может быть организовано двумя способами:

1) через сервер транзакций (рис. 7.2): использование коммерческих серверов транзакций подразумевает организацию более или менее стандартного интерфейса.

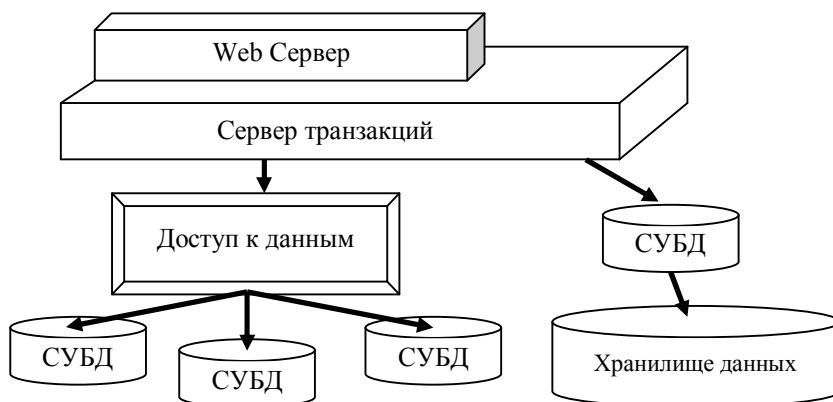


Рис. 7.2. Взаимодействие с БД с использованием сервера транзакций

2) через API интерфейс WWW сервера или сервера приложений: организация взаимодействия с базами данных при использовании API возможна по одной из приведенных структурных схем (рис. 7.3; 7.4); использование API приложений дает полную волю разработчикам.

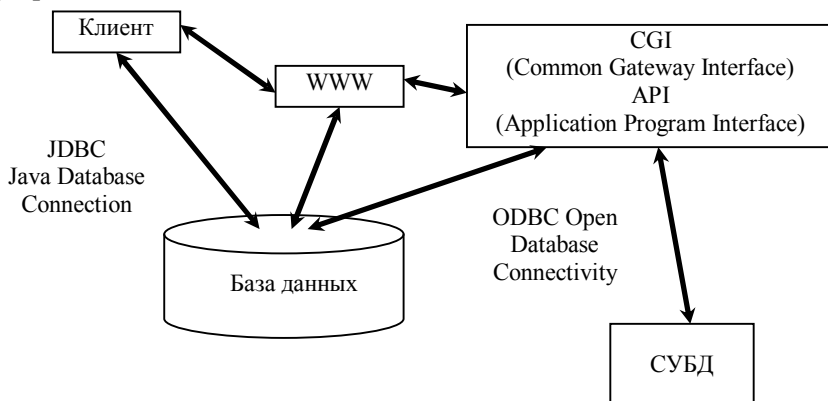


Рис. 7.3. Использование API интерфейса WWW сервера

На рисунке 7.3 представлена стандартная схема формирования информационной системы, основанная на использовании активных программ на сервере и стандартных средств доступа к БД, таких как Windows-NT ODBC интерфейс доступа к БД со стороны сервера и JDBC Java интерфейс доступа к БД со стороны клиента.

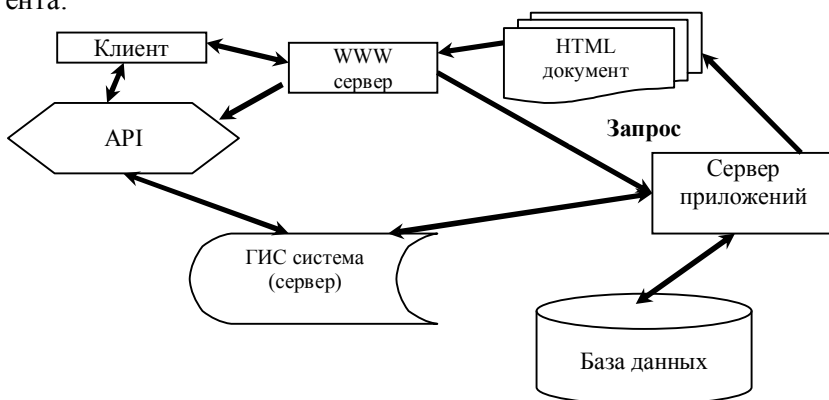


Рис. 7.4. Использование сервера приложений и API интерфейса WWW сервера

Схема, изображенная на рисунке 7.4, соответствует информационной системе, использующей сервер приложений. В случае размещения базы данных на машинах, находящихся в разных локальных сетях, необходимо строить доверительные базы с обязательным применением шлюзов для обеспечения прав доступа (рис. 7.5).

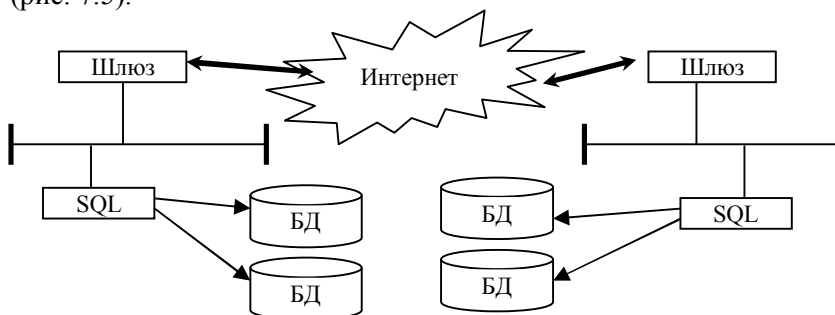


Рис. 7.5. Организация доверительных БД — работа через машину-посредника (шлюз)

7.2. ГИС-технологии и Internet

Использование технологий Internet позволяет объединять в единую информационную систему данные, расположенные в самых различных местах сети Internet. Для пользователя не имеет значения, где расположены эти источники информации. Осуществляя навигацию по карте, он может легко переходить от карты одного района к карте другого, не подозревая, что данные могут быть расположены на разных серверах сети. Геоинформационные системы объединяют данные, расположенные на различных серверах сети Internet.

Географическая информационная система (ГИС) — современная компьютерная технология для картографирования и анализа объектов реального мира, происходящих и прогнозируемых событий и явлений. Геоинформационные системы наиболее естественным образом отображают пространственные данные.

ГИС объединяет традиционные операции при работе с базами данных — запрос и статистический анализ — с преимуществами полноценной визуализации и географического (пространственного) анализа, которые предоставляет карта. Эта особенность дает

уникальные возможности для применения ГИС в решении широкого спектра задач, связанных с анализом явлений и событий, прогнозированием их вероятных последствий, планированием стратегических решений.

Данные в геоинформационных системах хранятся в виде набора тематических слоев, которые объединены на основе их географического положения. Этот гибкий подход и возможность геоинформационных систем работать как с векторными, так и с растровыми моделями данных эффективны при решении любых задач, касающихся пространственной информации.

ГИС тесно связаны с другими информационными системами и используют их данные для анализа объектов.

ГИС отличаются:

- развитые аналитические функции;
- возможность управлять большими объемами данных;
- инструменты для ввода, обработки и отображения пространственных данных.

ГИС широко используются в самых разных отраслях науки, управления и хозяйства, в том числе и для решения экологических задач. Эффективность использования геоинформационных технологий достигается картографической формой представления информации и оперативными возможностями ее анализа. Разноплановая информация представляется в виде отдельных слоев, а их совместный анализ в разных комбинациях позволяет получать дополнительную информацию в виде производных слоев, с их картографическим отображением (в виде изолинейных карт, совмещенных карт различных показателей и т.д.).

На рынке программных продуктов предлагаются различные ГИС, отличающиеся по функциональным возможностям, требованиям к аппаратным ресурсам и другим характеристикам.

Одна из широко распространенных в России ГИС — MapInfo Professional, разработанная фирмой MapInfo Corporation (США), а также 3D Kart Navigator.

3D Kart Navigator применяется в ОАО «Волготанкер» и используется для выполнения штурманских работ: прокладки курсов, расчета времени плавания, текущей навигации и т.д. На рисунке 7.6. представлен интерфейс ГИС.

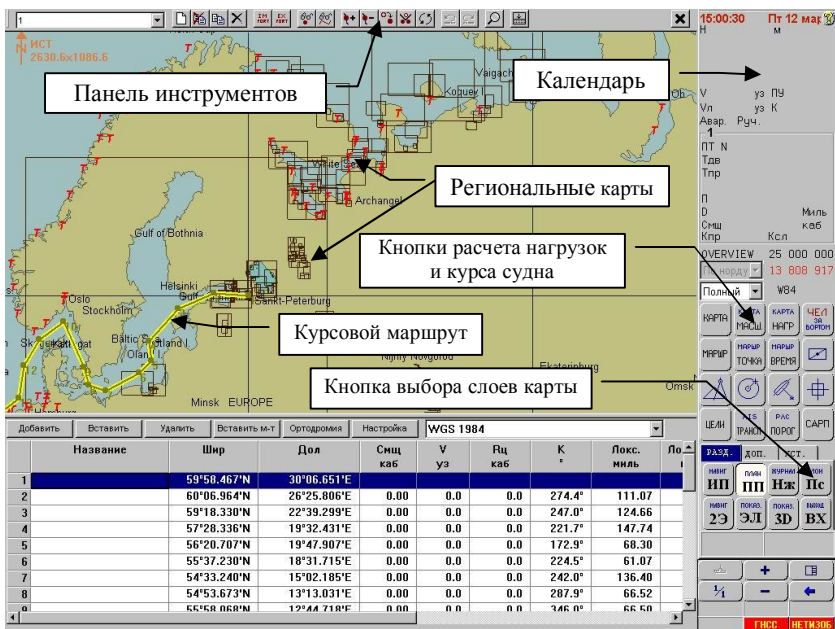


Рис. 7.6. Интерфейс D Kart Navigator

Internet-технологии предоставляют возможности создания распределенных ГИС, обеспечивая:

- 1) простоту администрирования сложных распределенных ГИС;
- 2) отсутствие необходимости тиражирования данных и программного обеспечения, его обновление выполняется на местах у держателей той или иной информации;
- 3) унифицированный интерфейс пользователя — для работы программы на клиентском компьютере используется стандартный www браузер иногда со встроенным картографическим компонентом (plugins + java приложения);
- 4) простоту установки программного обеспечения клиента, которое может устанавливаться (или обновляться на более новую версию) автоматически при входе на информационную страницу;
- 5) минимальную стоимость получения ГИС информации для конечного пользователя.

Принципы построения ГИС ориентированных систем в Internet можно классифицировать по трем основным способам хранения и передачи пространственных данных:

1) хранение и передача пространственных данных в виде растровых изображений (GIF или JPEG файлы);

2) хранение пространственных данных в векторном формате и передача их в растровом формате — этот подход используется в большинстве случаев (например, Internet Map Server фирмы ESRI), т.к. позволяет без дополнительного программного обеспечения у клиента реализовать систему для Internet; на основе этого подхода реализованы различные информационно-справочные системы;

3) создание интерактивной ГИС для Internet на основе архитектуры «клиент/сервер» с полностью векторным способом хранения и передачи пространственных данных и отображение последних с использованием активных клиентских приложений.

Последний подход имеет ряд преимуществ, поскольку обеспечивает все достоинства векторных карт и приемлемое время доступа к пространственным данным при интерактивной работе с электронной картой в условиях низкой пропускной способности каналов связи, позволяя осуществить избирательный принцип защиты информации на уровне отдельных картографических слоев.

При отображении электронной карты на клиентском месте можно использовать отдельные слои карты и связанные с картой атрибутивные данные, расположенные в разных базах и на разных серверах сети Internet. Это позволяет держателям картографической информации хранить собственные материалы на своих серверах и предоставлять к ним доступ определенным группам пользователей.

Преимущества векторного формата данных наиболее ощутимы при работе в сети со слабой пропускной способностью по следующим причинам:

1. Векторный формат более экономичен при передаче данных по сети.

2. Интерактивная работа с картой с возможностью формирования пространственных запросов по отдельным объектам.

3. Многослойное представление карты с возможностью управления слоями карты при помощи легенды.

4. Возможность отображения слоев карты только на заданных диапазонах масштабов визуализации.

5. Многократное использование данных, ранее полученных пользователем, без повторного обращения к серверу (кэширование данных).

6. Высокое качество изображения векторной карты, независимо от масштаба отображения.

Однако при работе в среде Intranet (в локальной сети) или в сети с хорошими каналами связи выгодно переложить работу с клиента на сервер. В этой ситуации более выгоден второй подход.

7.3. Безопасность

С вхождением России в мировое информационное пространство через информационные сети общего пользования, каковой является сеть Интернет, возникла проблема обеспечения безопасности различных информационных систем и создаваемых информационных ресурсов, а также организации управления информационными ресурсами и их содержанием.

Определяющим фактором интеграции в единое информационное пространство различных информационных систем и ресурсов является обеспечение должного уровня информационной безопасности, которая включает следующий комплекс мероприятий и технических решений по защите информации:

1) от нарушения функционирования сети путем исключения воздействия на информационные каналы, каналы сигнализации, управление и удаленную загрузку баз данных, коммуникационное оборудование, системное и прикладное программное обеспечение;

2) от несанкционированного доступа к информации путем обнаружения и ликвидации попыток использования ресурсов сети, приводящих к нарушению целостности сети и информации, изменению функционирования подсистем распределения информации;

3) разрушения встраиваемых средств защиты с возможностью доказательства неправомерности действий пользователей и обслуживающего персонала сети;

4) от внедрения программных «вирусов» и «закладок» в программные продукты и технические средства;

5) информационная безопасность Internet определяется особенностями базовых коммуникационной (TCP/IP) и операционной (UNIX) платформ.

Второй аспект защиты связан с принятием организационно-технологических мер построения сетей и информационных систем.

Реализация систем безопасности обеспечивается:

- конверсией технологий информационной безопасности и защиты информации и информационных систем, телекоммуникационной среды от несанкционированного использования и воздействий;

- обеспечением защиты ресурсов за счет параллельного доступа к управляющим базам данных и проверки полномочий при обращении к ресурсам сети;

- реконфигурацией сетей, узлов и каналов связи;

- организацией замкнутых подсетей и адресных групп;

- организацией специализированных защищенных компьютеров, локальных вычислительных сетей и корпоративных сетевых сегментов, что особенно важно для разработчиков информационных систем;

- обеспечением защиты технических средств и помещений от утечки по побочным каналам и от возможного внедрения в них электронных устройств съема информации;

- развитием и использованием технологий подтверждения подлинности объектов данных, пользователей и источников сообщений;

- использованием протоколов шифрования ip-пакетов, систем шифрования учетных данных и прав доступа, передача информации с использованием секретных ключей;

- применением технологий обнаружения целостности объектов данных.

Таким образом, реализация системы защиты информации и информационных ресурсов распадается на три независимые задачи, а именно:

- 1) обеспечение системы целостности информации и информационных систем;

- 2) организация авторизованного доступа к информации;

- 3) недопустимость появления в открытом доступе информации, составляющей государственную тайну или имеющей конфиденциальный характер.

При этом среди организационно-технологических мероприятий по защите информационных систем можно выделить три основных:

1) организация системы защиты на уровне IP-пакетов (технологический уровень модель OSI; рис. 7.7);

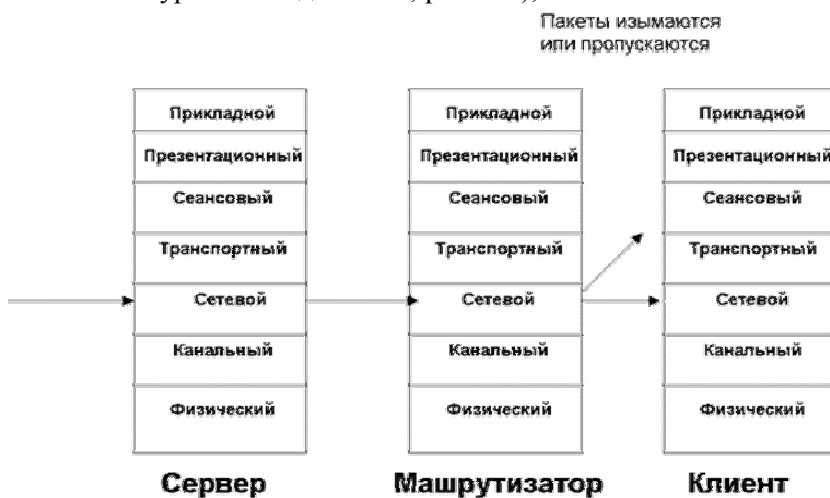


Рис. 7.7. Защита на уровне IP-пакетов

2) административный уровень защиты — контекстная проверка и просмотр пакетов с целью принятия решения (рис. 7.8);

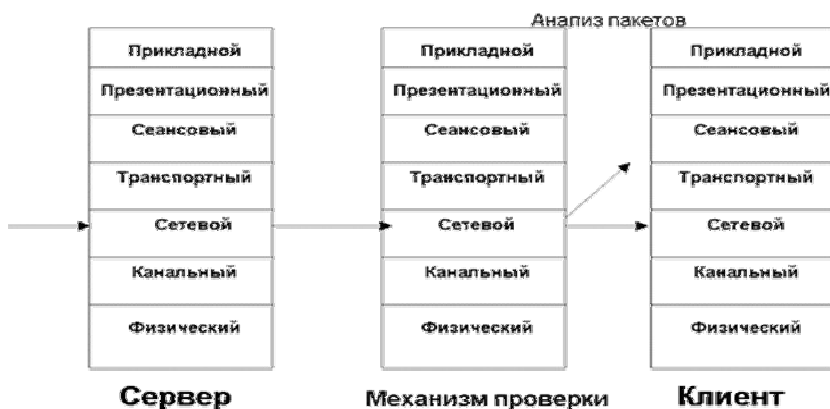


Рис. 7.8. Защита на уровне машины посредника (брандмауэр)

3) программный уровень защиты — шлюз выступает в качестве промежуточного звена между информационной системой и клиентом (рис. 7.9).

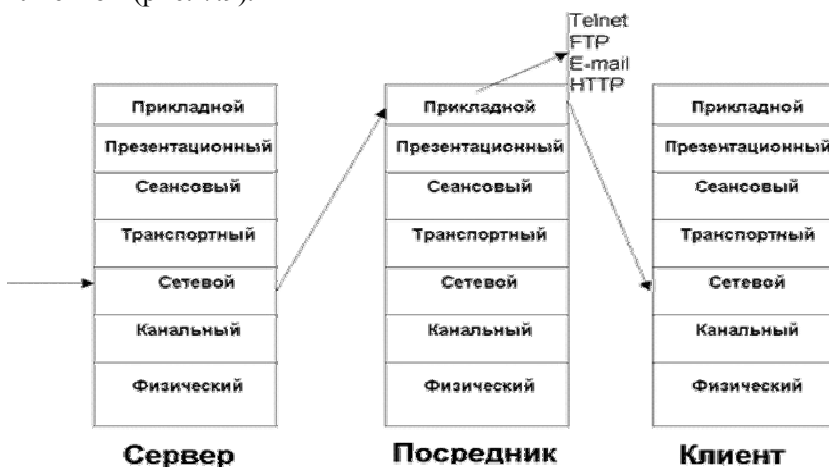


Рис. 7.9. Шлюз уровня приложений (проxy — посредник)

Для обеспечения конфиденциальности передаваемой по сети информации используются различные способы кодирования (шифрования) информации как на уровне приложений, так и на уровне пакетов.

Использование шифрования на уровне приложений определяется конкретным пользователем, в то же время Интернет-технологии позволяют организовать шифрование сообщений независимо от приложений на уровне IP-пакетов.

Протокол, управляющий шифрованием трафика IP-пакетов SKIP (Simple Key management for Internet Protocol — простой протокол управления криптоключами в Интернет), разработан компанией Sun Microsystems и предложен в качестве стандарта Internet.

В основе протокола SKIP лежит криптография открытых ключей, и он имеет по сравнению с существующими системами шифрования трафика ряд уникальных особенностей, а именно:

- универсальность (т.к. шифрует IP-пакеты, не зная ничего о приложениях, пользователях или процессах;
- сеансонеzависимость (т.к. за исключением однажды и навсегда запрошенного открытого ключа партнера по связи);

— независимость от системы шифрования (в том смысле, что различные системы шифрования могут подсоединяться к системе как внешние библиотечные модули).

Контрольные вопросы

1. Поясните взаимодействие с БД через Интернет.
2. Поясните взаимодействие с БД с использованием сервера транзакций.
3. Поясните использование API интерфейса WWW сервера.
4. Поясните использование сервера приложений и API интерфейса WWW сервера.
5. Поясните организацию доверительных БД.
6. Перечислите основные отличия ГИС.
7. Перечислите мероприятия и технические решения по защите информации.
8. Чем обеспечивается реализация систем безопасности?
9. Как организуются системы защиты на уровне IP-пакетов?
10. Чем обеспечивается административный уровень защиты?
11. Чем обеспечивается программный уровень защиты?
12. Что лежит в основе протокола SKIP?

8. Интеллектуальные информационные системы

8.1. Классификация интеллектуальных информационных систем

Интеллектуальные информационные системы (ИИС) ориентированы на генерацию алгоритмов решения задач и характеризуются следующими признаками:

- развитые коммуникативные способности (характеризуют способ взаимодействия интерфейса конечного пользователя с системой, в частности, возможность формулировать запросы на языке, максимально приближенном к естественному);

- умение решать сложные, плохо формализованные задачи (задачи, требующие построения оригинальных алгоритмов в зависимости от конкретной ситуации);

- способность к самообучению (возможность автоматического извлечения знаний для решения задач из накопленного опыта конкретных ситуаций);

- адаптивность (способность к развитию системы в соответствии с объективными изменениями модели проблемной области);

Каждому признаку интеллектуальности соответствует свой класс информационной системы (рис. 8.1). Это могут быть:

- 1) системы с интеллектуальным интерфейсом;
- 2) экспертные системы;
- 3) самообучающиеся системы;
- 4) адаптивные системы.

8.2. Системы с интеллектуальным интерфейсом

Интеллектуальные базы данных отличаются тем, что могут дать ответ на запрос, данные для которого в явном виде в базе отсутствуют. Такая информация генерируется на основе следующих зависимостей:

- вычислительных (например: «вывести список лекарств, цены на которые ниже среднеотраслевых» — необходимо провести статистический расчет цены по отрасли, а затем сделать вывод);

- структурных (например: «вывести список лекарств-заместителей» — необходимо вывести характерные значения признаков, а затем по ним осуществить поиск аналогичных объектов);

— логических (например: «вывести список потенциальных покупателей товара» — необходимо определить список посредников-продавцов, а затем, используя их базы потребителей, произвести поиск связанных с ними покупателей).

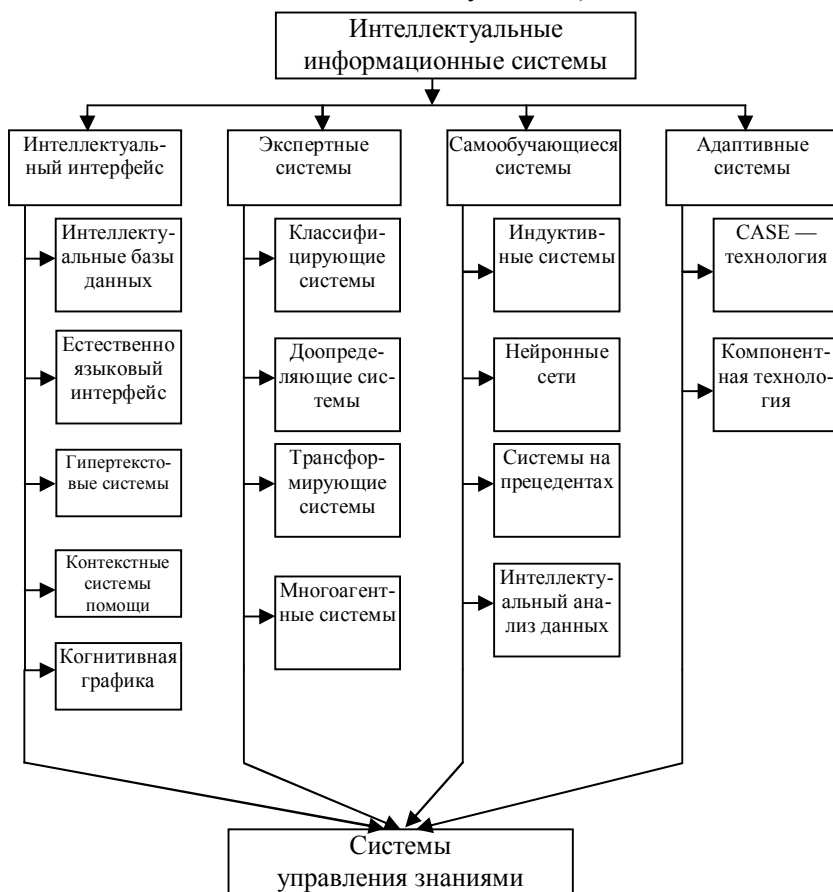


Рис. 8.1. Классификация интеллектуальных информационных систем

В интеллектуальных базах данных осуществляется поиск по условию, которое доопределяется в ходе решения задачи и информационная система без помощи пользователя по структуре базы данных сама строит путь доступа к файлам. Запрос пользователя осуществляется на основе естественно-языкового интерфейса в максимально удобной для него форме.

Естественно-языковый интерфейс предполагает трансляцию запросов и команд естественного языка на внутримашинный уровень представления знаний. Для этого необходимо решать задачи морфологического, синтаксического и семантического видов анализа и синтеза высказываний на естественном языке. Естественно-языковый интерфейс используется:

- для доступа к интеллектуальным базам данных;
- для контекстного поиска документальной текстовой информации;
- голосового ввода команд в системах управления;
- машинного перевода с иностранных языков.

Гипертекстовые системы предназначены для реализации поиска по ключевым словам в базах текстовой информации. Интеллектуальные гипертекстовые системы отличаются возможностью более сложной семантической организации ключевых слов, которая отражает различные смысловые отношения терминов. Механизм поиска работает с базой знаний ключевых слов, а затем непосредственно с текстом. Такие механизмы распространяются и на поиск мультимедийной информации, включающей текст, цифровую информацию, графические, аудио- и видеобразы.

Системы контекстной помощи можно рассматривать как частный случай интеллектуальных гипертекстовых и естественно-языковых систем. В таких системах пользователь описывает проблему, а система с помощью дополнительного диалога ее конкретизирует и сама выполняет поиск относящихся к ситуации рекомендаций. Их можно отнести к классу систем распространения знаний, а область применения связана преимущественно с приложениями к технической документации и к инструкциям по эксплуатации.

Системы когнитивной графики осуществляют интерфейс пользователя с помощью графических образов, которые генерируются в соответствии с происходящими событиями. Графические образы в наглядном и интегрированном виде описывают множество параметров изучаемой ситуации. Системы когнитивной графики широко используются в обучающих и тренажерных системах на основе виртуальной реальности, когда графические образы моделируют ситуацию.

8.3. Архитектура экспертных систем

Экспертные системы предназначены для решения задач на основе накапливаемой базы знаний, отражающей опыт работы экспертов в рассматриваемой проблемной области. Системы такого типа позволяют принимать решения в уникальных ситуациях, для которых алгоритм заранее неизвестен и формируется по исходным данным в виде цепочки рассуждений (правил принятия решений) из базы знаний. Экспертная система — инструмент усиления интеллектуальных способностей эксперта. Она выполняет функции:

- консультанта для неопытных или непрофессиональных пользователей;
- ассистента в связи с необходимостью анализа экспертом различных вариантов принятия решений;
- партнера эксперта по вопросам, относящимся к источникам знаний из смежных областей деятельности.

Архитектура экспертной системы (рис. 8.2) включает два основных компонента: базу знаний (хранилище единиц знаний) и программный инструмент доступа и обработки знаний, состоящий из механизма вывода заключений (решения), приобретения знаний, объяснения полученных результатов и интеллектуального интерфейса.

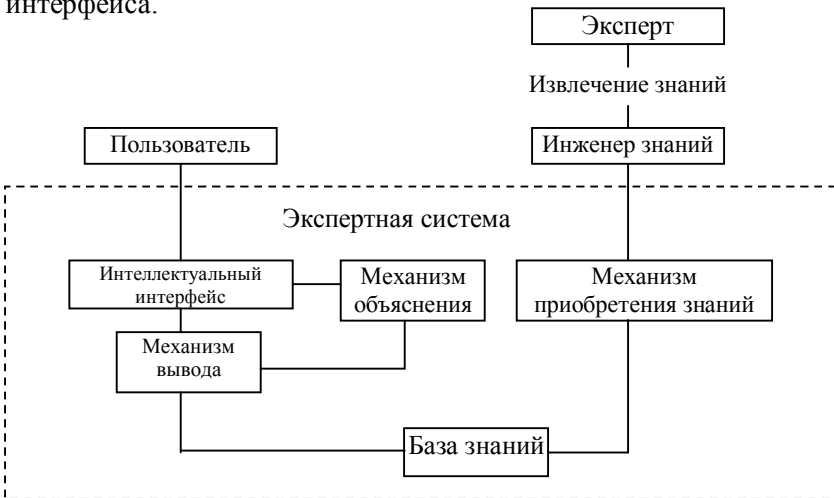


Рис. 8.2. Архитектура экспертной системы

Центральным компонентом экспертной системы является база знаний, выступающая по отношению к остальным компонентам как содержательная подсистема. Стоимость ее превышает остальные компоненты на порядок.

База знаний содержит совокупность единиц знаний, которые формализованы с помощью некоторого метода представления знаний об объектах предметной области, их взаимосвязи, действий над объектами и неопределенности, с которой эти действия осуществляются.

В качестве методов представления знаний используются либо правила, либо объекты (фреймы), либо их комбинации.

Правила представляют собой конструкции следующего типа:

Если <условие>

То <заключение> CF (Фактор определенности) <значение>

В качестве фактора определенности (CF) выступают условные вероятности байесовского подхода (от 0 до 1) либо коэффициент уверенности нечеткой логики (от 0 до 100). Фреймы представляют собой совокупность атрибутов, описывающих свойства и отношения с другими фреймами. В отличие от обычных записей, в базах данных каждый фрейм имеет уникальное имя. Для фреймового представления (объектно-ориентированного) применяется механизм наследования атрибутов, т.е. значения передаются от вышестоящего к нижестоящему.

Интеллектуальный интерфейс — это программа, осуществляющая обмен данными между пользователем и экспертной системой. Диалог должен вестись на языке, максимально приближенном к естественному. Последовательность решения задачи должна быть гибкой, должна соответствовать представлениям пользователя и вестись в профессиональных терминах.

Механизм вывода — это программа, формирующая из базы знаний конкретный алгоритм решения задачи. В основе лежит процесс связывания конкретной ситуации с единицами знаний и образования цепочки рассуждений (прямой или обратной).

Механизм объяснения поясняет ход решения по запросам пользователя. Объяснительные способности экспертной системы определяются способностью механизма вывода запоминать путь решения задачи и определенное количество «ходов». По запросу пользователя «Как» или «Почему» механизм вывода должен выдать

цепочку рассуждений до контрольной точки, на которой был осуществлен запрос. Полный ход решения зачастую может не интересовать пользователя, и механизм вывода должен уметь выбирать из цепочки рассуждений ключевые моменты с учетом их важности и знаний пользователя. При отсутствии понимания со стороны пользователя система должна переходить в диалоговый режим обучения на основе поддерживаемой модели проблемной области.

Механизм приобретения знаний. База знаний отражает знания экспертов в данной области знаний. Выявлением подобных знаний и представлением их в базе знаний занимаются **инженеры знаний**. Для ввода знаний и их последующего обновления экспертная система должна иметь в простейшем случае интеллектуальный редактор, проверяющий новые единицы знаний на непротиворечивость, проводящий синтаксический и семантический контроль. В более сложных случаях используются специальные сценарии опроса экспертов или извлекаются единицы знаний из опыта работы самой экспертной системы.

8.4. Классы экспертных систем

Экспертные системы по степени сложности делятся на:

1. В зависимости от способа формирования решения:
 - а) *Аналитические* (предполагают выбор решения из множества известных альтернатив);
 - б) *Синтетические* (генерируют неизвестные решения).
2. По способу учета временных признаков:
 - а) *Статические* (решают задачи при неизменных данных и знаниях, монотонно от ввода до конечного результата);
 - б) *Динамические* (допускают как изменение данных, так и возможность пересмотра в процессе решения полученных результатов и данных).
3. По виду используемых данных и знаний:
 - а) *Детерминированные* (данные и знания четко определены);
 - б) *Неопределенные* (данные и знания неполные, недостоверные, нечеткие или двусмысленные).
4. По числу используемых источников знаний:
 - а) *Один источник знаний*.
 - б) *Множественные источники знаний*. Могут быть альтернативными или дополняющими (кооперирующими).

В соответствии с перечисленными признаками выделяют следующие классы экспертных систем (рис. 8.3):

	Анализ	Синтез	
Достоверность знаний	Классифицирующие	Трансформирующие	Один источник - знаний
Неопределенность знаний	Доопределяющие	Многоагентные	Множество источников знаний
	Статика	Динамика	

Рис. 8.3. Классы экспертных систем

Классифицирующие системы решают задачи распознавания ситуаций, относя их к определенному классу. В качестве метода решений используется дедуктивный вывод (от общего к частному).

Доопределяющие системы решают задачи на основе неопределенных исходных данных и применяемых знаний. В качестве методов решения используется байесовский вероятностный подход, коэффициент уверенности, аппарат нечеткой логики.

Трансформирующие системы для решения задач используют повторяющиеся преобразования знаний в процессе решения задач, что связано с характером результата, который нельзя заранее предопределить, а также с динамичностью проблемной области. В качестве методов решения задач применяются такие разновидности гипотетического вывода, как:

- 1) генерация и тестирование (по исходным данным осуществляется генерация гипотез, а затем гипотезы проверяются поступающими фактами);
- 2) предположения и умолчания (по неполным данным подбираются знания об аналогичных классах объектов, которые в дальнейшем динамически адаптируются к конкретной ситуации в зависимости от ее развития);
- 3) использование общих закономерностей (в случае неизвестных ситуаций на основе общих закономерностей генерируются недостающие знания).

Многоагентные системы являются динамическими и используют распределенное решение задач. Каждый агент имеет собственную базу знаний и механизм вывода. Для многоагентных систем характерны следующие особенности:

— альтернативность рассуждений на основе использования различных источников знаний с механизмом устранения противоречий;

— распределенное решение проблем за счет разбиения на подпроблемы и их параллельного решения;

— множественность стратегий работы механизма вывода в зависимости от типа решаемых проблем;

— способность к самообучению с изменением правил поведения агентов;

— способность прерывания решения задач в связи с необходимостью получения дополнительных данных и знаний от пользователя, моделей, параллельно решаемых подпроблем.

Синтезирующие динамические системы применяются в следующих областях:

1) проектирование — определение конфигурации объектов с точки зрения достижения заданных критериев (бюджет предприятия);

2) прогнозирование — предсказание последствий развития текущей ситуации (тренды на биржевых торгах);

3) планирование — выбор последовательности действий по достижению поставленных целей (планирование цепочек поставок);

4) диспетчерирование — распределение работ во времени (планирование графиков работ, расписание занятий в вузе и т.д.);

5) мониторинг — слежение за текущей ситуацией (мониторинг сбыта готовой продукции);

6) управление — мониторинг, дополненный реализацией действий.

8.5. Экспертные медицинские системы

Врач во время работы (приема пациента, проведения операции, выписки рецепта и т.д.) должен сделать определенное заключение, ведущее к конкретным действиям, на основании фактов, полученных из наблюдений, ответов на вопросы, лабораторных анализов, измерения давления, пульса, температуры и других показателей. При этом он руководствуется знаниями, полученными при обучении, чтении специальной литературы, собственным

опытом, интуицией, советами коллег. Для облегчения принятия решения создаются и широко распространяются компьютерные экспертные медицинские системы, выполняющие роль интеллектуального справочника, предлагающего варианты вопросов к пациенту, основываясь на полученной до этого информации. Вводимая информация может быть не только текстовая, но и графическая, в виде кардиограмм, рентгенограмм и т.д. Система анализирует все параметры, учитывая историю болезни пациента, хранящуюся в базе данных, и предлагает врачу свой вариант диагноза и возможных действий. При этом она основывается на базе знаний квалифицированных специалистов, которая создавалась при разработке программы. Поэтому такую систему правильнее назвать «справочник-консультант». Конечно, окончательное решение принимает врач.

Поскольку эта сфера применения очень ответственна, существующие экспертные системы используют достаточно сложные алгоритмы, снижающие вероятность неверных действий, поэтому здесь трудно рассчитывать на универсальность и многопрофильность. Профессиональные медицинские пакеты очень специализированы. Например, программы экстренной диагностики и неотложной терапии, диагностики болей в области сердца, предупреждения осложнений беременности, анестезиологии и т.п.

В то же время компьютерные системы, и основанные на теории методов распознавания образов, и интеллектуальные (чаще говорят экспертные) медицинские системы должны опираться в своем создании на следующих критериях:

- достоверность и внутренняя непротиворечивость полученных о пациенте данных;
- устойчивость к большому числу дифференцируемых нозологических форм, в том числе гетерогенных по своей природе и/или клиническим проявлениям;
- интерпретация данных на основе медицинских знаний, характеризующихся отсутствием специфических проявлений идентифицируемых заболеваний;
- учет ассоциирующих симптомов в неявном виде, учитываемых врачом;
- принятие решения в условиях недостатка информации о проявлениях болезни у конкретного больного;

- объяснение и обоснование полученного системой решения;
- переоценка решений, высказанных в предыдущей консультации, в случае повторного обследования больного.

Медицинская направленность компьютерных диагностических систем определяется поставленными задачами и следующими уровнями применения:

- 1) предварительная диагностика на долабораторном этапе обследования с построением дифференциального ряда;
- 2) диагностика с выбором оптимального метода исследования для последующей окончательной нозологической идентификации патологии;
- 3) нозологическая диагностика с постановкой (обоснованием) конкретного диагноза на основе имеющейся информации (клинической и параклинической);
- 4) ориентировочная диагностика с оценкой тяжести состояния как основа для принятия решения (управления уровнем помощи) при угрожающих состояниях.

В отношении выбора методологии для построения автоматизированной системы наибольший интерес в настоящее время представляет класс экспертных систем (ЭС), опирающихся на теорию искусственного интеллекта. Это объясняется тем, что такие системы включают базу специальных знаний в конкретной проблемной области и предоставляют возможность ознакомления с протоколом результатов распознавания медицинской ситуации и логического объяснения предлагаемого системой решения.

Рассмотрим отдельные системы, наиболее интересные в медицинском плане (все они функционируют в тех или иных учреждениях России и СНГ). В дальнейшем изложении будем ориентироваться в основном на отечественные разработки в этой области, учитывая существенно важный аспект учета особенностей российских медицинских научных школ и реализации систем на естественном для нас русском языке.

В качестве примера начнем рассмотрение с созданной в Московском НИИ педиатрии и детской хирургии *автоматизированной системы для синдромной диагностики неотложных состояний у детей* «ДИН». Эта ЭС содержит информацию о 42 синдромах, которые представляют собой список диагностических предположений-гипотез. Поскольку выбор лечения во многом определяется

прогнозом возможных осложнений, в системе описаны взаимосвязи синдромов, определяемые причинно-следственными, временными и ассоциативными отношениями. Причинно-следственные связи предполагают информацию о синдромах, которые могут быть причиной данного синдрома или, наоборот, являться его следствием. В последнем случае речь идет о прогнозировании возможных осложнений, обусловленных наблюдающимся у ребенка в данный момент синдромом. Временные связи позволяют восстанавливать информацию о предшествующих синдромах, которые могли послужить причиной того состояния, которое имеется в данный момент, что особенно важно для тех случаев, когда ребенок поступает в отделение реанимации без анамнеза. Ассоциативные связи позволяют учитывать:

- на фоне каких состояний мог развиваться данный синдром;
- фоном для каких синдромов он может служить;
- с какими синдромами он может сочетаться, т.е. какие синдромы могут встречаться у ребенка одновременно.

Врач получает информацию о ведущем синдроме или синдромах (с указанием степени или стадии заболевания), имеющихся на данный момент у ребенка, а если желает, то и весь список гипотетических состояний, рассмотренных системой в процессе постановки диагноза.

Системы, ориентированные на вопросы неотложной диагностики и интенсивной терапии, создают новые возможности для повышения эффективности лечения при критических состояниях, находятся в прямой связи со снижением инвалидизации и смертности детей.

Несомненный интерес для диагностики, классификации и коррекции терапии острых расстройств кровообращения у детей представляет программный комплекс «Айболит», созданный в Центре сердечно-сосудистой хирургии имени А.Н.Бакулева и активно применяемый при оперативных вмешательствах и выборе послеоперационного лечения в условиях реанимационного отделения. В нем используются математические модели, методы имитации. Начальные оценки свойств делаются на основании обработки данных больных, получаемых в процессе мониторингового контроля нарушений кровообращения. Система работает в режиме диалога «врач — ЭВМ», в реальном режиме времени, обрабатывая

данные, получаемые с медицинской аппаратуры, контролирующей состояние жизненно важных систем организма.

Информационная и экспертно-диагностическая система «ДИНАР» для реанимационно-консультативного центра (РКЦ), разработанная Санкт-Петербургской педиатрической медицинской академией и Свердловской ОДКБ при участии сотрудников Института биофизики УроРАН, позволяет обеспечить дистанционное наблюдение за больными с угрожаемыми состояниями, определение ведущего патологического синдрома и степени тяжести, помощь при выборе лечения, принятие тактического решения с учетом распределения централизованных ресурсов медицинского обеспечения.

Экспертная система «ДИНАР» используется как автоматизированное рабочее место врача-диспетчера РКЦ. Диагностический блок обеспечивает автоматизированный опрос (анамнез жизни и заболевания, динамика клинических проявлений) и определяет оптимальное тактическое решение. Справочный блок предоставляет врачу РКЦ необходимую для принятия тактического решения информацию (о лечебных учреждениях, по фармакологии и диагностике угрожающих состояний, схемы лечения основных неотложных состояний). В банке данных накапливаются сведения о консультированных системой больных. Опрос врача-абонента осуществляется на первом этапе по жестко заданному дереву вопросов, на втором — по алгоритму автоматического выбора списков вопросов, а на последующих этапах реализован более свободный принцип задания вопросов.

Посиндромный подход к диагностике угрожающих жизни состояний новорожденного, при котором по выраженности отдельных синдромов оценивается состояние основных систем организма, реализован в Российском государственном медицинском университете.

8.6. Самообучающиеся системы

Самообучающиеся системы основаны на методах автоматической классификации примеров реальной практики (обучение на примерах). Примеры накапливаются за определенный исторический период и составляют обучающую выборку, которая может быть:

1. С «учителем» (для каждого примера задается в явном виде значение признака его принадлежности к некоторому классу);
2. Без «учителя» (система сама выделяет классы ситуаций по степени близости значения признаков классификации).

В результате обучения системы автоматически строятся обобщенные правила или функции, определяющие принадлежность задачи к определенному классу.

К общим недостаткам самообучающихся систем относятся:

- возможная неполнота или зашумленность (избыточность) обучающей выборки (проблемы с адекватностью базы знаний);
- проблемы со смысловой ясностью зависимых признаков и соответственно неспособность объяснить пользователю полученных результатов.
- ограничения в размерности признакового пространства и соответственно неглубокое описание проблемной области.

В самообучающихся системах выделяются **индуктивные системы**, которые характеризуются обобщением примеров по принципу от частного к общему. Процесс классификации примеров осуществляется следующим образом:

1. Выявляется подмножество примеров, относящихся к одним и тем же подклассам, и определяются их значимые признаки.
2. По значению выбранного признака множество признаков разбивается на подмножества.
3. Выполняется проверка принадлежности образовавшегося подмножества примеров одному классу.
4. Если подмножество примеров принадлежит одному подклассу, то классификация заканчивается.
5. Для несовпадающих примеров классификация продолжается.

8.7. Нейронные сети

Можно выделить два способа решения задачи — логический и интуитивный. Логический метод оперирует набором формальных правил, интуитивный — накопленным опытом. При решении задачи первым способом она обычно разбивается на подзадачи, каждая из которых, в свою очередь, разбивается на несколько элементарных функций с известным алгоритмом вычислений. Зная

алгоритм каждой элементарной функции, можно решать сложнейшие задачи, соединяя элементарные функции в нужной последовательности. Традиционное программирование как раз и обеспечивает такую возможность. При попытке применить такой метод к решению интуитивных задач программист может столкнуться с невозможностью выделить алгоритм элементарной функции или может вообще разбить задачу на такие функции.

Современный компьютер, обрабатывая информацию с большой скоростью, делает это на основе алгоритмов, написанных программистом. Разумеется, с усложнением задач резко возрастает время, затрачиваемое на написание алгоритмов их решения. Поэтому трудности разработки экспертных систем определяются уже не техническими, а человеческими факторами.

Растущие объемы информации, появление новейших диагностических и лечебных технологий стали причинами принципиально иных подходов к обработке и интерпретации медицинских данных, возможности накопления, хранения и использования опыта квалифицированных специалистов. Для научных исследований в медицине также уже совершенно недостаточно использовать стандартные наборы статистических методов для обработки материала. Требуется универсальные инструменты с возможностью гибкой постановки задач, применимые ко всем областям биологии и медицины, обладающие большой информационной емкостью и помехоустойчивостью, не нуждающиеся в длительном времени для разработки. Всем вышеперечисленным требованиям могут отвечать системы, способные самонастраиваться на решение задач. Созданные только один раз исходно одинаковыми, они должны обучаться решать совершенно разные задачи.

Возможности настройки нового класса экспертных систем должны распространяться вплоть до индивидуального пользователя, например, до конкретного врача, имеющего возможность обучать экспертную систему на своем собственном опыте и на основе тех данных, которые доступны этому врачу. Это означает, что необходимо коренное изменение технологии производства таких систем. Существующие традиционные системы принятия решений, основанные на четких правилах вывода, создаются, как правило, группой специалистов, в числе которых математики, программисты и предметные специалисты, ставящие задачи.

Возможности настройки таких систем на конечного потребителя почти всегда сведены к минимуму, и он, приобретая такую систему, часто сталкивается с ее неприменимостью к конкретным условиям его работы (например, другой спектр лабораторных анализов или методов обследования, принятых в данной клинике). Выход — дать специалисту возможность самому конструировать экспертную систему исходя из конкретных условий, собственного опыта и опыта коллег. Такое конструирование должно производиться без знания предметным специалистом математического аппарата, с требованием только обычных навыков работы на ПК. Кроме того, в этой ситуации снимается проблема доверия к заключениям экспертной системы, которая работает, опираясь на опыт и знания того специалиста, который ее сконструировал, его коллег, которым он доверяет, а также основываясь на реальных данных, которые он сам получил в результате наблюдений.

Основные требования, которым должны удовлетворять самообучающиеся медицинские экспертные системы принятия решений, диагностики, назначения лечебно-профилактических мероприятий и прогнозирования, таковы:

1) индивидуализация (настройка на традиции клинических школ, геосоциальные особенности региона применения, наборы медико-биологических данных, особенности лечебно-диагностических технологий, индивидуальный опыт и знания специалиста);

2) динамическое развитие (накопление опыта системы в процессе функционирования, следуя изменениям в пунктах, перечисленным в предыдущем требовании);

3) возможность перенастройки при резком изменении условий (например, при перенесении в другой регион);

4) способность к экстраполяции результата (требование, обратное индивидуальности) — система не должна резко терять качество работы при изменении условий;

5) возможность конструирования «с нуля» конечным пользователем (специалист должен иметь возможность придумать совершенно новую экспертную систему и иметь возможность про-сто и быстро создать ее);

6) «нечеткий» характер результата — важное требование, подразумевающее, что решение выдаваемое системой, не должно быть окончательным. Оно может быть, например, вероятностным

либо может предлагать сразу несколько вариантов на выбор. Это дает возможность специалисту критически оценивать решение системы и не лишает его инициативы в принятии окончательного решения.

7) экспертная система является только советчиком специалиста, не претендуя на абсолютную точность решения, она должна накапливать опыт и знания и значительно ускорять доступ к ним, моделировать результат при изменении условий задачи; при этом ответственность за решение всегда лежит на специалисте.

Этим требованиям достаточно хорошо удовлетворяют нейроинформационные технологии, базирующиеся на искусственных нейронных сетях. В основе функционирования нейронных сетей лежат алгоритмы, моделирующие распространение сигналов по нейронам и синапсам нервной системы. Существует достаточно большой набор архитектур и метаалгоритмов функционирования нейронных сетей, при этом задачи, решаемые современной нейроинформатикой, в большинстве случаев требуют подгонки архитектуры и алгоритмов обучения нейросетей под определенный класс задач или даже под конкретную задачу. Поэтому разработка универсальной технологии создания медицинских экспертных систем, включающей оптимизацию архитектур и метаалгоритмов функционирования нейронных сетей, теоретические и методологические основы функционирования нейросетей при работе с медико-биологической информацией, является актуальной задачей.

Все задачи, решаемые человеком, с позиций нейроинформационных технологий можно условно разделить на две группы:

1) задачи, имеющие известный и определенный набор условий, на основании которого необходимо получить четкий, точный, недвусмысленный ответ по известному и определенному алгоритму;

2) задачи, в которых не представляется возможным учесть все реально имеющиеся условия, от которых зависит ответ, а можно лишь выделить приблизительный набор наиболее важных условий (т.к. часть условий при этом не учитывается, ответ носит неточный, приблизительный характер, а алгоритм нахождения ответа не может быть выписан точно).

Для решения задач первой группы с большим успехом можно использовать традиционные компьютерные программы. Как бы

ни был сложен алгоритм, ограниченность набора условий (входных параметров) дает возможность составления алгоритма решения и написания конкретной программы, решающей данную задачу. Нет никакого смысла в использовании нейроинформационных технологий для решения таких задач, т.к. в этом случае нейросетевые методы будут априорно хуже решать такие задачи. Единственным исключением является случай, когда алгоритм вычисления ответа слишком большой и громоздкий, а время на решение конкретной задачи по этому алгоритму не удовлетворяет практическим требованиям; кроме того, при получении ответа не требуется абсолютная точность.

При решении задач второй группы применение нейротехнологии оправдывает себя по всем параметрам, при выполнении, однако, двух условий: во-первых, наличия универсального типа архитектуры и единого универсального алгоритма обучения (отсутствие необходимости в их разработке для каждого типа задач), во-вторых, наличия примеров (предыстории, фиксированного опыта), на основании которых производится обучение нейронных сетей. При выполнении этих условий скорость создания экспертных систем возрастает в десятки раз, и соответственно снижается их стоимость.

Практически вся медицинская и биологическая науки состоят именно из задач, относящихся ко второй группе, и в большинстве этих задач достаточно легко набрать необходимое количество примеров для выполнения второго условия. Это задачи диагностики, дифференциальной диагностики, прогнозирования, выбора стратегии и тактики лечения и др. Медицинские задачи практически всегда имеют несколько способов решения и «нечеткий» характер ответа, совпадающий со способом выдачи результата нейронными сетями.

Все неалгоритмируемые или трудноалгоритмируемые задачи, решаемые нейронными сетями, можно классифицировать на два принципиально различающихся типа в зависимости от характера ответа — задачи классификации и задачи предикции.

Задачи классификации — это основная и очень обширная группа медико-биологических задач. Ответом в них является класс — выбор одного варианта из заранее известного набора вариантов. Классификация может быть бинарной (элементарная

классификация) — в этом случае набор возможных ответов состоит из двух вариантов (классов) — и n -арной, где число классов более двух. Примерами бинарной классификации могут служить как объективные категории (пол человека — мужской или женский; характер опухоли — доброкачественный или злокачественный), так и субъективные (здоров человек или болен; наличие или отсутствие склонности к простудным заболеваниям). В некоторых случаях не представляется возможным отнесение ответа задачи к объективной или субъективной категории, и это не имеет принципиального значения для обучения и работы нейросетевой экспертной системы.

Важной чертой задачи классификации по определению является возможность выбора одного и только одного варианта решения (класса), поэтому постановка диагноза не может считаться одной классификационной задачей, т.к. у одного человека может одновременно присутствовать несколько патологий. В случае невозможности выбора одного варианта ответа (множественности выбора) задача подразделяется на подзадачи, каждая из которых представляет собой классификационную задачу.

Другой вид задач для нейросетей — задачи предикции, или предсказания. Они подразделяются на предсказание числа (одномерная предикция) и вектора (векторная предикция, более общий случай). Отличие от классификационных задач заключается в том, что ответ в задачах предикции может быть дробным и может принимать любые значения на каком-либо интервале.

Векторная предикция предполагает, что ответ может быть представлен в виде нескольких независимых друг от друга чисел, образующих точку (или вектор) в многомерном пространстве, размерность которого равна количеству предсказываемых чисел. Число координат вектора называется при этом размерностью вектора ответа.

При решении реальных задач возможны различные комбинации предикции и классификации, и задача должна быть поставлена самим предметным специалистом.

8.7.1. Архитектура нейронной сети

Основой работы самообучающихся нейропрограмм является нейронная сеть, представляющая собой совокупность нейронов — простых элементов, связанных между собой определенным образом. Нейроны и межнейронные связи задаются программно на обычном компьютере или могут иметь «материальную» основу — особую микросхему (нейрочип), которые применяются в специально созданных нейрокомпьютерах. Структура взаимосвязей между нейронами в нейрокомпьютере или нейропрограмме аналогична таковой в биологических объектах. Искусственный нейрон имеет коммуникации с другими нейронами через синапсы, передающие сигналы от других нейронов к данному (дендриты) или от данного нейрона к другим (аксон). Кроме того, нейрон может быть связан сам с собой. Несколько нейронов, связанных между собой определенным образом, образуют нейронную сеть.

Нейросеть, так же как и биологический аналог, должна иметь каналы для связи с внешним миром. Одни каналы обеспечивают поступление информации из внешнего мира на нейросеть, другие выводят информацию из нейросети во внешний мир. Поэтому одни нейроны сети рассматриваются как входные, другие же — как выходные. Часть нейронов может не сообщаться с внешним миром, а взаимодействовать с входными, выходными и такими же нейронами («скрытые» нейроны).

Существует огромное количество способов соединения нейронов, растущее с увеличением числа нейронов в сети. Наиболее употребительной является слоистая архитектура, в которой нейроны располагаются «слоями». В наиболее общем случае аксоны каждого нейрона одного слоя направлены к нейронам следующего слоя. Таким образом, нейроны первого слоя являются входными (принимающими информацию из внешнего мира), нейроны последнего слоя — выходными (выдающими информацию во внешний мир). Схема трехслойной сети изображена на рис. 8.4.

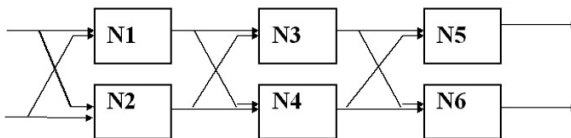


Рис. 8.4. Трехслойная сеть с шестью нейронами

Другой вид архитектуры — полносвязная, когда каждый нейрон соединен с каждым, в том числе сам с собой. Пример простейшей нейросети из трех нейронов показан на рисунке 8.5. Для удобства изображения из каждого нейрона выходит не один, а несколько аксонов, направленных на другие нейроны или во внешний мир, что аналогично присоединенным к одному аксону через синапсы нескольким дендритам.

Именно полносвязные нейросети для построения экспертных систем выбираются исходя из следующих соображений: во-первых, при одинаковом числе нейронов полносвязные сети имеют большее количество межнейронных связей, что увеличивает информационную емкость сети; во-вторых, полносвязная архитектура является намного более универсальной, что не требует экспериментов с вариациями схемы соединений для каждой задачи; в-третьих, в случае эмуляции сети на обычной ЭВМ полносвязные сети обладают серьезными преимуществами — прежде всего скоростью функционирования и простотой программной реализации без ущерба для качества обучаемости.

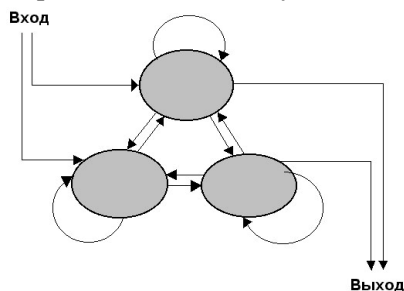


Рис. 8.5. Схема простейшей нейронной сети из трех нейронов
(сеть имеет тринадцать синапсов, четыре из которых служат для связи с внешним миром, а остальные соединяют нейроны между собой)

8.7.2. Функционирование нейрона

Рассмотрим устройство и функционирование отдельного нейрона. Каждое соединение от нейрона к нейрону называется синапсом. На рисунке 8.6 представлен нейрон с группой синапсов, соединяющих нейрон либо с другими нейронами, либо с внешним миром. Для рассмотрения работы нейрона неважно, приходит ли сигнал к нейрону из внешнего мира или с другого нейрона,

и неважно, куда отправляется сигнал с нейрона. В полносвязных сетях выходной сигнал направляется всем остальным нейронам.

Нейрон состоит из двух функциональных блоков: входного сумматора и собственно нейрона, или преобразователя.

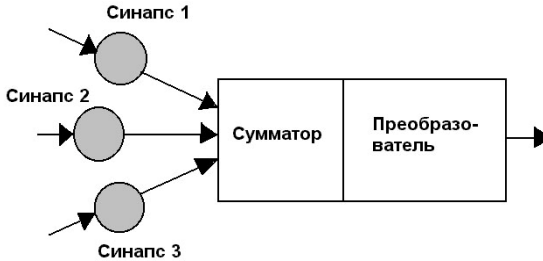


Рис. 8.6. Схема нейрона

Функционирование нейрона происходит следующим образом.

В текущий момент времени через входные синапсы (на рисунке их три) на нейрон направляются сигналы от других нейронов и/или из внешнего мира. Каждый синапс имеет параметр, называемый весом синапса и представляющий какое-либо число. Сигнал, проходящий через синапс, умножается на вес этого синапса. В зависимости от веса сигнал может быть усилен (модуль веса > 1) или ослаблен (модуль веса < 1) по амплитуде. Сигналы от всех синапсов, ведущих к данному нейрону, принимает сумматор.

Сумматор производит суммирование всех пришедших сигналов и подает на собственно нейрон (преобразователь) одно число — полученную сумму. Величина этого числа будет зависеть как от величин исходных сигналов, так и от весов синапсов. Нейрон, получивший это число, преобразует его согласно своей функции, в результате которой получается другое число, и отправляет его по «аксону» всем остальным нейронам через соответствующие синапсы. Последующие нейроны производят с полученными сигналами такие же операции, с тем лишь различием, что, во-первых, веса их синапсов могут быть уже другими, во-вторых, другие нейроны могут иметь другой вид функции преобразования. В конструируемых нами нейронных сетях все нейроны имеют одну и ту же функцию. Эта функция, называемая характеристической, имеет вид:

$$f(X) = X/(C + |X|) \quad (1),$$

где X — сигнал, поступающий с сумматора, C — константа, называемая характеристикой нейрона. Экспериментальным путем мы получили, что оптимальный диапазон характеристики для решения подавляющего большинства задач составляет от 0,1 до 0,8. Графики характеристической функции для обоих случаев представлены на рисунке 8.7. Выбор функции такого вида обусловлен тем, что она гладкая, непрерывная на всем диапазоне переменных X , диапазон значений всегда ограничен. Подбор характеристики и ее влияние на работу нейронной сети будут рассмотрены ниже.

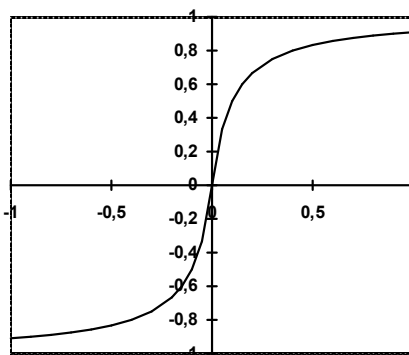


Рис. 8.7. График характеристической функции

8.7.3. Функционирование нейросети

В случае эмуляции нейросети на обычном компьютере все математические операции осуществляет программа. Нейронная сеть при этом представляет собой массив синаптических весов. Этот массив может находиться либо на диске компьютера в виде файла определенного формата (при хранении нейросети), либо в оперативной памяти компьютера (когда нейросеть функционирует).

При создании новой нейросети в памяти компьютера отводится место под массив синаптических весов, называемый картой. Этот массив заполняется совершенно случайными числами из определенного диапазона. Поэтому каждая созданная сеть даже при одних и тех же параметрах (число нейронов, их характеристика) является уникальной. Уникальность сетей проявляется в том, что

сети с одинаковыми параметрами, обучающиеся на одинаковых задачах, ведут себя неодинаково. Это касается времени обучения, качества обучения, уверенности в выдаваемых ответах при тестировании. В этом еще одно сходство сетей с биообъектами.

Рассмотрим работу сети безотносительно к процессу обучения или выдачи решения. После инициализации сети веса всех синапсов случайны. В начальный момент времени на входные нейроны через входные синапсы (один или несколько) подается из внешнего мира вектор входных сигналов, представляющий набор чисел (или одно число). Далее этот сигнал начинает распространяться по всем связям между нейронами, изменяясь при прохождении через каждый нейрон согласно схеме функционирования нейрона. В конечном итоге, после одного прохода выходные нейроны выдадут во внешний мир какие-либо сигналы. Вся процедура однократного прохождения сигналов по нейронной сети является тактом функционирования сети. Можно не считывать сигналы с выходных нейронов после одного такта функционирования, а продолжить обмен сигналами еще несколько раз, не подавая сигналов на вход. Обычно количество тактов функционирования между подачей сигналов на вход и снятием сигналов с выхода фиксировано для данной сети. Как правило, этот параметр задается при инициализации сети и хранится в файле сети вместе с матрицей коэффициентов (как и некоторые другие параметры: число нейронов, характеристика нейронов и т.д.).

Таким образом, нейронная сеть, получающая на входе некоторый сигнал, способна после прохода его по нейронам выдавать на выходе определенный ответ, который зависит от весовых коэффициентов всех нейронов и от самого сигнала. Очевидно, что при осуществлении таких процедур на только что инициализированной сети мы будем получать на выходе сигналы, лишенные всякого смысла (весовые коэффициенты случайны). Чтобы добиться выдачи сетью требуемого результата, необходимо обучить ее.

8.7.4. Общая схема обучения нейронной сети

Для обучения нейронной сети необходима обучающая выборка (задачник), состоящая из примеров. Каждый пример представляет собой задачу одного и того же типа с индивидуальным набором

условий (входных параметров) и заранее известным ответом. Например, в качестве входных параметров в одном примере могут использоваться данные обследования одного больного, тогда заранее известным ответом в этом примере может быть диагноз. Несколько примеров с разными ответами образуют задачник. Задачник располагается в базе данных, каждая запись которой является примером.

Рассмотрим общую схему обучения нейросети.

1. Из обучающей выборки берется текущий пример (изначально первый), и его входные параметры (представляющие в совокупности вектор входных сигналов) подаются на входные синапсы обучаемой нейросети. Обычно каждый входной параметр примера подается на один соответствующий входной синапс.

2. Нейросеть производит заданное количество тактов функционирования, при этом вектор входных сигналов распространяется по связям между нейронами (прямое функционирование).

3. Измеряются сигналы, выданные теми нейронами, которые считаются выходными.

4. Производится интерпретация выданных сигналов, и вычисляется оценка, характеризующая различие между выданным сетью ответом и требуемым ответом, имеющимся в примере. Оценка вычисляется с помощью соответствующей функции оценки. Чем меньше оценка, тем лучше распознан пример, тем ближе выданный сетью ответ к требуемому. Оценка, равная нулю, означает, что требуемое соответствие вычисленного и известного ответов достигнуто. Заметим, что только что инициализированная (необученная) нейросеть может выдать правильный ответ совершенно случайно.

5. Если оценка примера равна нулю, ничего не предпринимается. В противном случае на основании оценки вычисляются поправочные коэффициенты для каждого синаптического веса матрицы связей, после чего производится подстройка синаптических весов (обратное функционирование). В коррекции весов синапсов и заключается обучение.

6. Осуществляется переход к следующему примеру задачника, и вышеперечисленные операции повторяются. Проход по всем примерам обучающей выборки от первого до последнего считается одним циклом обучения.

При прохождении цикла каждый пример имеет свою оценку. Вычисляется, кроме того, суммарная оценка множества всех примеров обучающей выборки. Если после прохождения нескольких циклов она равна нулю, обучение считается законченным, в противном случае циклы повторяются.

Количество циклов обучения, так же как и время, требующееся для полного обучения, находятся в зависимости от многих факторов — величины обучающей выборки, количества входных параметров, вида задачи, типа и параметров нейросети и даже от случайного расклада весов синапсов при инициализации сети.

С учетом подразделения задач на классификационные и предикционные рассмотрим принцип вычисления оценки у нейросетей, решающих такие задачи.

8.7.5. Обучение нейросетей-классификаторов

Обучаемая нейросеть может иметь любое количество нейронов, которое, однако, не должно быть меньше числа классов в задаче, решению которой обучается нейросеть. Обычно количество нейронов устанавливается равным числу входных сигналов в задаче или меньшим. Как правило, в полносвязных нейросетях, применяемых нами, все нейроны являются входными (каждый из них принимает один, часть или все входные сигналы). Выходными считаются последние нейроны, причем их число в сети-классификаторе равно числу классов в задаче. После работы сети с одним примером выходные сигналы снимаются с каждого выходного нейрона отдельно.

При решении классификационной задачи нужно интерпретировать ответы выходных нейронов, т.к. окончательным ответом является класс, к которому относится пример. Интерпретация производится следующим образом: т.к. число выходных нейронов совпадает с количеством классов в решаемой задаче, каждый выходной нейрон отвечает за класс с соответствующим номером. Номер выходного нейрона, выдавшего наибольший выходной сигнал, и есть номер класса, к которому, по мнению нейросети, относится данный пример. Однако при обучении необходимо не только установить факт совпадения или несовпадения ответов, но и вычислить оценку, показывающую, насколько вычисленный ответ отличается от требуемого. Это достигается вычислением

разности между двумя максимальными выходными сигналами, выданными выходными нейронами. При этом выявляется не только выходной нейрон, выдавший максимальный сигнал, но и выходной нейрон, выдавший второй по величине сигнал. Если число выходных нейронов больше двух (n -арная классификация), ответы остальных нейронов не имеют значения. Если разность не превышает заранее установленного значения (уровня надежности), оценка приравнивается к нулю, пример считается распознанным абсолютно правильно, а коррекции синаптических весов на данном примере не требуется.

Ситуацию распознавания примера в задачах классификации можно представить в виде некоторых действий, производящихся в пространстве, размерность которого равна числу классов в задаче. В случае бинарной классификации рабочее пространство представляет собой квадратный участок плоскости (двухмерного пространства), минимальные и максимальные координаты которого ограничены значениями -1 и 1 .

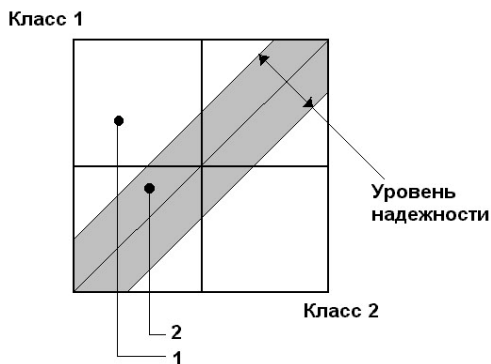


Рис. 8.8. Схема проекции обучающих примеров в двухмерном пространстве при бинарной классификации: 1 — пример первого класса, попавший в зону «надежности»; 2 — пример первого класса, распознанный правильно, но с недостаточным уровнем надежности

Рассмотрим рисунок 8.8. Значения вертикальной координатной оси соответствуют выходным сигналам первого выходного нейрона (отвечающего за первый класс), значения горизонтальной координатной оси — выходным сигналам второго выходного нейрона (второй класс). Пространство разделяется на две части диагональной линией. В текущий момент времени параметры

очередного примера подаются на входные нейроны нейросети. После всех преобразований ответ примера проецируется в определенную точку пространства и в зависимости от значений, снятых с выходных нейронов, попадает в одну из областей, разделенных диагональю. Область каждого класса в пространстве соответствует как можно большим значениям (1) «своей» оси координат и как можно меньшим значением (–1) другой оси координат.

Таким образом, чем больше будет значение, снятое с какого-либо выходного нейрона, тем больше будет значение координаты точки по оси, соответствующей этому нейрону, и наоборот. Ситуация, когда значения, снятые и с того, и с другого нейронов, равны, приведет к попаданию точки на диагональ (независимо от значений чисел). Пунктирной линией отсечены зоны «надежности», находящиеся у углов квадрата, соответствующих классам задачи. Чем больше расстояния между линиями, тем больше заданный уровень надежности и, соответственно, выше требования к сети. Пример, ответ которого «попал» в свою область, и к тому же в зону надежности, считается распознанным абсолютно верно.

В процессе обучения ответы примеров, подаваемых сети многократно, постепенно «разводятся» как можно дальше от диагонали и как можно ближе к углам, каждый из которых соответствует определенному классу. После того, как все примеры попали в зоны «надежности», соответствующие их классам, сеть считается обученной полностью. Постепенно повышая уровень надежности, можно добиваться большей точности ответов сети и лучшего распознавания примеров обучающей выборки.

При каком минимальном превышении сигнала одного выходного нейрона над сигналом другого следует считать пример правильно распознанным? Универсального решения не существует, поэтому специалист, обучающий нейросети, сам задает уровень надежности исходя из реальной задачи. В некоторых случаях достаточно только правильно определить класс (с минимальным превышением одного сигнала над другими). Тогда задается небольшой уровень надежности. В других случаях уровень надежности должен быть увеличен. Максимальное значение уровня надежности равно 2. Это идеальное число, которое в принципе недостижимо при обучении. Очевидно, что при больших значениях уровня надежности сеть будет обучаться дольше. Кроме того, для

каждой реальной задачи существует свой предельный уровень надежности, с которым может обучиться нейросеть.

В основе обучения нейросети (подстройки синаптических весов) лежит усовершенствованный метод двойственного функционирования. В приведенной схеме число входных нейронов точно соответствовало количеству обучающих параметров (размерности вектора входных сигналов). Однако в наших экспертных системах мы использовали более гибкую и одновременно более мощную подачу входных сигналов на вход нейросети. Суть ее заключается в том, что вектор входных сигналов перед подачей на сеть преобразуется умножением на адаптивную матрицу приема сигналов, содержащую подстраиваемые, как и веса синапсов, значения. Кроме повышения скорости и качества обучения, это дает возможность задавать число нейронов в сети произвольным, независимым от числа обучающих параметров. Сети с небольшим числом нейронов, естественно, занимают меньше места на диске и в памяти компьютера, быстрее загружаются с диска для выдачи ответа (иногда для тестирования примеров приходится каждый раз загружать сети с диска, особенно если сетей много, а все они одновременно не помещаются в оперативной памяти).

8.7.6. Обучение нейросетей-предикторов

Обучение сетей-предикторов принципиально не отличается от обучения классификаторов. Различия заключаются в способе оценки ответа сети и смысле уровня надежности.

Как уже говорилось, ответ в задачах предикции выражается любым числом (или вектором чисел). При этом оценивать правильность ответа достаточно просто. В случае одномерной предикции оценка равна квадрату разности между известным и вычисленным ответами. Некоторые изменения претерпевает структура нейросети.

В качестве исходной берется полносвязная сеть с адаптивной матрицей приема сигналов. Все нейроны сети на последнем такте функционирования считаются выходными. Над сетью достраивается адаптивный выходной сумматор, принимающий сигналы с нейронов с последнего такта функционирования и выдающий результат. Вводится также вектор сдвига, который складывается

покоординатно с сигналами, приходящими на нейроны в момент подачи входных сигналов. Вектор сдвига подается на нейроны на каждом такте, а не только в момент подачи входных сигналов, и для удобства содержится в матрице связей нейронов.

Таким образом, при решении задач предикции смысл уровня надежности кардинально меняется. Теперь его снижение приведет к тому, что от сети будет требоваться большая точность при выдаче ответа на пример. Поэтому уровень надежности в предикторах уместнее называть уровнем отклонения. Естественно, с него снимаются ограничения по величине, т.к. точность решения задачи (величина уровня отклонения) зависит от размерности параметра, являющегося ответом, и может изменяться в широких пределах.

В векторных предикторах, где ответ представлен набором чисел, оценка вычисляется как сумма квадратов отклонений по каждой составляющей вектора ответа. Такой же смысл имеет и параметр, задаваемый как уровень отклонения.

8.7.7. Методологические аспекты обучения нейросетей

Иногда (крайне редко при решении медико-биологических задач) встречаются ситуации, когда сеть не может обучаться. Это происходит в том случае, когда на определенном этапе обучения исчерпываются дальнейшие возможности поиска закономерностей между обучающими параметрами и результатами. Простейшая ситуация - когда два примера с совершенно одинаковыми наборами параметров подаются сети как принадлежащие разным классам (в классификаторах) или имеющие различное значение ответа (в предикторах). Очевидно, оба этих примера всегда будут попадать в одну и ту же точку в пространстве, их невозможно будет отделить друг от друга, и процесс обучения остановится. Программа, управляющая нейросетями, сигнализирует об окончании процесса обучения, причем указывает, что дальнейшее обучение невозможно. Задача специалиста, обучающего нейросети, — избежать таких ситуаций, а для этого требуются четкая постановка задачи и тщательный контроль обучающей выборки.

Обученная нейросеть автоматически записывается на диск компьютера как обыкновенный файл и может храниться там,

сколько необходимо. В любой момент времени можно считать сеть с диска и продолжить обучение решению данной задачи со старой или новой обучающей выборкой. Одна нейросеть обучается решать только одну задачу классификации или предикции, однако может использовать для обучения различные обучающие выборки. Они могут различаться по количеству примеров, но должны соответствовать друг другу по числу обучающих параметров, числу классов (в классификационной задаче), а главное, по смыслу.

Говоря об обучении нейросетей, следует рассмотреть еще один важный аспект этой темы. Успех обучения во многом зависит от числа нейронов в сети, точнее, от числа синапсов. Именно весовые коэффициенты синапсов хранят «опыт» сети. Теоретически, бесконечно увеличивая число нейронов и синапсов, всегда можно добиться полного обучения сети на данном задатнике, однако это ли является целью создателя экспертной системы? Очевидно, нет. Главное, чтобы обученная сеть хорошо распознавала примеры, как раз не входящие в задатник.

Проблема заключается в том, что сеть с заведомо большим (избыточным) числом синапсов (относительно данного задатника) может хорошо обучиться, просто «механически запомнив» имеющиеся примеры. Такая сеть обучится быстро (нет необходимости как можно более точной подстройки весов) за счет количества, а не качества.

Хорошим практическим выходом из данной затруднительной ситуации были бы сети, способные автоматически наращивать число нейронов при невозможности дальнейшего обучения, не теряя при этом уже имеющегося опыта. Последнее условие вызывает значительные трудности. Нейросеть представляет собой единое целое, и добавление нового нейрона к сети, работающей в рамках имеющейся сейчас концепции, приведет к необходимости полностью переучивать сеть. Это требует обращения к первоначальному задатнику, что во многих случаях неприемлемо.

Поэтому создателю самообучающихся систем приходится идти на компромисс: либо делать сеть с некоторым избытком нейронов, имеющую резерв для накопления опыта, но обладающую относительно низкой способностью к экстраполяции, либо обучить сеть с небольшим числом нейронов, которая вряд ли сможет

набрать потом дополнительный опыт. Все это, конечно, зависит еще и от задачника: от того, насколько тесные взаимосвязи имеются между обучающими параметрами и известными ответами примеров. Чем больше таких взаимосвязей, тем меньше необходимости в «механическом запоминании» примеров. Практика показывает, что большинство биологических и медицинских задач имеют достаточно хорошие взаимосвязи (конечно, при грамотной постановке задачи и выборе обучающих параметров). Однако в рамках предлагаемой методологии с учетом высокой скорости обучения нейросетей разработаны стратегия и тактика обучения, позволяющие обойти вышеуказанный компромисс, правда, за счет большего времени, необходимого для обучения.

8.8. Системы управления знаниями

Система управления знаниями (далее — СУЗ) — это совокупность организационных процедур, организационных подразделений (служб управления знаниями) и компьютерных технологий, которые обеспечивают интеграцию разнородных источников знаний и их коллективное использование в деловых процессах. Отличительной особенностью системы управления знаниями является интеграция множества разнородных, часто территориально распределенных источников знаний для решения общих задач. СУЗ интегрирует знания как из внутренних, так и из внешних источников. Источники знаний могут иметь недокументированную форму (неявные знания экспертов), документированную текстовую, табличную, графическую и структурируемую форму в виде баз знаний экспертных систем.

Система управления знаниями обычно используется в двух аспектах:

- обеспечение качественными знаниями процессов решения различных задач;

- создание интерактивной среды взаимодействия специалистов в процессе решения задач.

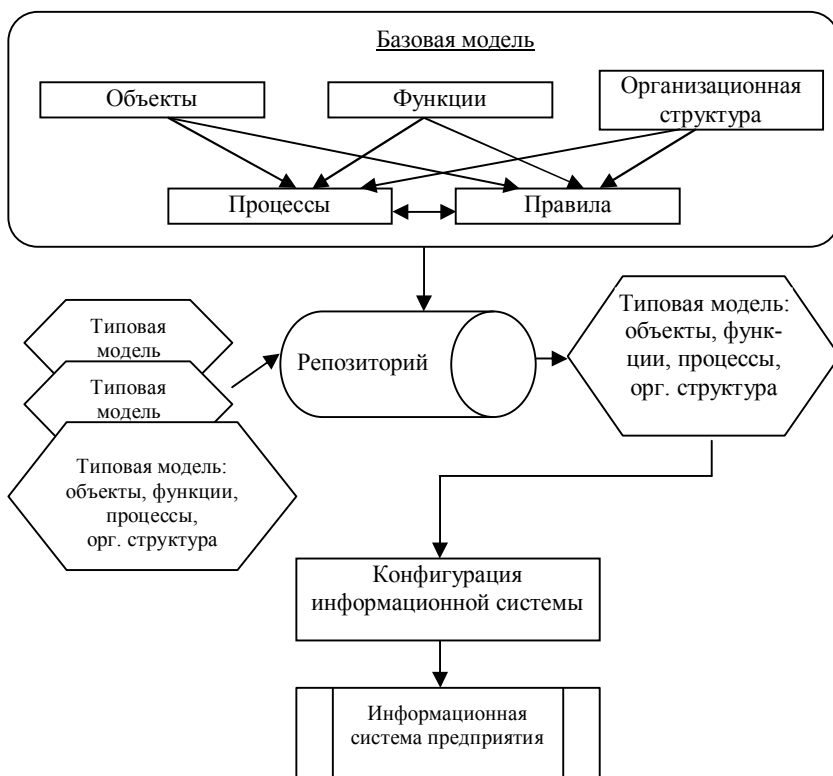


Рис. 8.9. Конфигурация адаптивной информационной системы на основе компонентной технологии

Для систем управления знаниями характерны следующие особенности:

1. **Интеллектуальное ассистирование.** Система управления знаниями в отличие от экспертных систем не заменяет эксперта в процессе решения задач, а помогает ему, обеспечивая релевантной информацией и правилами принятия решений в конкретной ситуации. При этом в процессе решения задачи пользователь рассматривает различные варианты решений, представляемые системой управления знаниями, модифицирует постановку задачи или моделирует ситуацию, выбирая, таким образом, наиболее приемлемые решения. В другом режиме решения задачи пользователь самостоятельно решает задачу, а результат решения оценивает с

помощью системы управления знаниями на предмет правильности, эффективности, например, обращаясь с помощью системы управления знаниями к коллегам-экспертам или экспертным системам.

2. Сбор и систематическая организация знаний из различных источников. Интеграция множества разнородных источников знаний осуществляется на основе единой системы концептуализации знаний. Основное требование к источникам знаний — предотвратить потерю и увеличить доступность всех видов корпоративных знаний путем обеспечения централизованного, хорошо структурированного информационного хранилища.

3. Минимизация проектирования базы знаний как наполнения концептуальной схемы. Не все источники знаний могут быть известны, они могут быть или противоречивыми, или могут часто изменяться. Система управления знаниями, с одной стороны, должна обеспечивать методы обработки таких ситуаций, а с другой стороны, должна обеспечивать легкое подключение новых источников знаний по мере появления. При этом должна модифицироваться и сама концептуальная схема системы управления знаниями.

4. Быстрая адаптация системы управления знаниями к изменяющимся информационным потребностям. Адаптация системы управления знаниями осуществляется на основе обратной связи пользователей к системе в результате формулирования ими новых требований и разрезов анализа, а также информирование о пробелах, неверности и запаздывании знаний. Кроме того, происходит наполнение базы прецедентов с указанием успешного и неудачного решения проблем пользователями. При этом выполняется преобразование индивидуального знания в групповое знание, доступное для пользователей системы управления знаниями.

5. Интеграция с существующей программной средой. К системе управления знаниями подключаются разнообразные аналитические инструменты, которые позволяют проводить извлечение неявной информации, содержащейся в источниках знаний, с помощью таких методов, как статистический анализ и нейронные сети, экспертные системы, математическое и имитационное моделирование. Эти инструменты позволяют обнаруживать закономерности в отражаемой действительности и выявлять

наиболее рациональное поведение в существующих условиях. Пользователь становится исследователем-экспериментатором, который не просто запрашивает интересующую его информацию, но и выдвигает и проверяет различные гипотезы. В этом смысле полезно подключение специализированных инструментальных программных средств.

6. Активная презентация релевантной информации. Система управления знаниями становится компетентным партнером в кооперативном решении проблем, учитывающем круг информационных потребностей пользователя и формирующем для него знания исходя из его предполагаемых интересов. Особенно эффективно это свойство реализуется в системах управления отношениями с клиентами (CRM — customer relationship management).

Контрольные вопросы

1. Дайте определение понятия «интеллектуальные информационные системы».
2. Перечислите классы информационных систем.
3. Дайте определение понятия «интеллектуальные базы данных».
4. Поясните назначение гипертекстовых систем.
5. Поясните назначение систем контекстной помощи.
6. Поясните назначение систем когнитивной графики.
7. Поясните архитектуру экспертной системы.
8. Приведите классификацию экспертных систем.
9. Поясните особенности экспертных медицинских систем.
10. Приведите пример компьютерной диагностической системы.
11. Поясните принцип функционирования самообучающейся системы.
12. Перечислите основные требования, которым должны отвечать самообучающиеся медицинские экспертные системы принятия решений.
13. Перечислите основные требования, которым должны отвечать самообучающиеся медицинские экспертные системы диагностики.

14. Перечислите основные требования, которым должны удовлетворять самообучающиеся медицинские экспертные системы назначения лечебно-профилактических мероприятий.

15. Перечислите основные требования, которым должны удовлетворять самообучающиеся медицинские экспертные системы прогнозирования.

16. Поясните архитектуру нейронной сети.

17. Поясните устройство и функционирование отдельного нейрона.

18. Приведите общую схему обучения нейронной сети.

19. Поясните особенности обучения нейросетей-классификаторов.

20. Поясните особенности обучения нейросетей-предикторов.

21. Дайте определение понятия «система управления знаниями».

22. Поясните конфигурацию адаптивной информационной системы на основе компонентной технологии.

СОДЕРЖАНИЕ

Введение	3
1. Информационные системы	6
1.1. Понятие информации.....	6
1.2. Информационный обмен и информационный процесс	10
1.3. Сети информационного обмена.....	12
<i>Контрольные вопросы</i>	17
2. Классификация информационных систем	19
2.1. Архитектура «клиент/сервер»	26
2.1.1. RDA-модель	26
2.1.2. DBS-модель	28
2.1.3. AS-модель	30
2.2. Области применения и примеры реализации информационных систем	33
<i>Контрольные вопросы</i>	34
3. Жизненный цикл информационных систем	36
3.1. Классификация проектов информационных систем.....	36
3.2. Разработка, эксплуатация	38
3.3. Основные процессы жизненного цикла	40
3.4. Структура жизненного цикла	42
3.5. Каскадная и спиральная модели жизненного цикла.....	43
<i>Контрольные вопросы</i>	46
4. Информационные системы документального поиска	48
4.1. Телемедицина	48
4.2. Интернет и информационно-поисковые системы.....	50
4.3. Содержание документального поиска. Релевантность и пертинентность.....	51
4.4. Функциональные операционные схемы информационно-поисковых систем	52
4.5. Документы, информационно-поисковый язык, поисковые массивы.....	54
4.6. Типы информационно-поисковых систем	55
4.7. Лингвистическое обеспечение	56

4.8. Информационно-поисковые языки с иерархической классификацией	58
4.9. Вербальные информационно-поисковые языки	59
4.10. Семантические и синтагматические языки	60
4.11. Индексирование	61
4.12. Языки запросов, структура запросов	62
4.13. Язык описания данных	62
4.14. Информационный поиск в сети Интернет	66
4.15. Поисковые системы, механизмы поиска	67
4.16. Агенты и кроулеры, роботы	70
4.17. Сравнительный обзор поисковых систем	71
<i>Контрольные вопросы</i>	80
5. Корпоративные системы документооборота	82
5.1. Корпоративные информационные системы	82
5.2. Корпоративная система электронного документооборота Lotus Notes	84
5.3. Корпоративные сети и соединения	87
5.4. Автоматизированная система ведения медицинской документации	90
<i>Контрольные вопросы</i>	93
6. Информационные системы на базах данных	94
6.1. Банки данных, базы данных, базы знаний	94
6.2. Синтаксический, семантический, прагматический аспекты информационного процесса	96
6.3. Моделирование отношений <i>сущность/связь</i> . E/R-модели	98
6.4. Обобщенная архитектура СУБД	102
6.5. Архитектура системы медицинской информационной системы на основе базы данных	110
6.5.1. Безопасность информационной системы	116
6.5.2. Распределение прав доступа	117
<i>Контрольные вопросы</i>	120
7. Распределенные информационные системы	121
7.1. Internet/Intranet-технологии	121
7.2. ГИС-технологии и Internet	125

7.3. Безопасность	129
<i>Контрольные вопросы</i>	133
8. Интеллектуальные информационные системы	134
8.1. Классификация интеллектуальных информационных систем	134
8.2. Системы с интеллектуальным интерфейсом	134
8.3. Архитектура экспертных систем	137
8.4. Классы экспертных систем	139
8.5. Экспертные медицинские системы	141
8.6. Самообучающиеся системы	145
8.7. Нейронные сети	146
8.7.1. Архитектура нейронной сети	152
8.7.2. Функционирование нейрона	153
8.7.3. Функционирование нейросети	155
8.7.4. Общая схема обучения нейронной сети	156
8.7.5. Обучение нейросетей-классификаторов	158
8.7.6. Обучение нейросетей-предикторов	161
8.7.7. Методологические аспекты обучения нейросетей	162
8.8. Системы управления знаниями	164
<i>Контрольные вопросы</i>	167

Учебное издание

***Абрамов Николай Васильевич, Мотовилов Николай Васильевич,
Наумов Николай Дмитриевич, Черкасов Сергей Николаевич***

ИНФОРМАЦИОННЫЕ СИСТЕМЫ В МЕДИЦИНЕ

Учебное пособие

Художник обложки *Л.П.Павлова*
Компьютерная верстка *Е.С.Борзова*

Изд. лиц. ЛР № 020742. Подписано в печать 01.04.2008
Формат 60×84/16. Бумага для множительных аппаратов
Гарнитура Times. Усл. печ. листов 10,75
Тираж 500 экз. Заказ 704

*Отпечатано в Издательстве
Нижевартовского государственного гуманитарного университета
628615, Тюменская область, г.Нижевартовск, ул.Дзержинского, 11
Тел./факс: (3466) 43-75-73, E-mail: ngpirc@wsmail.ru*