

Информационные системы

**Учебное пособие по дисциплине Информационные системы
для специальностей ПИЭ и ПИЮ**

Составители Орел А. А., Ромакина О.М.

Саратов 2001

Глава 1	2
Основные понятия	2
Глава 2.....	6
Применение информационных систем	6
Автоматизированные информационные системы (АИС)	9
Жизненный цикл ИС	12
Эффективность ИС	13
Пользователи ИС. Трехуровневое представление данных	14
История и основные направления развития ИС.....	15
Классификация ИС	17
Структура программного обеспечения ИС.....	20
Структура приложений в ИС	21
Структура хранения информации в ИС	21
Классификация программных решений.....	22
Этапы разработки автоматизированных информационных систем	24
Разработка и анализ бизнес-модели	25
Основные понятия электронного документооборота	26
Электронный документооборот	26
Модели информационного пространства предприятия.....	27
Эволюция модели документооборота	28
Глава 3.....	30
Теоретические основы информационных систем	30
Знаковые системы	30
Предметная область	37
Модель "сущность-связь"	45
Глава 4.....	61
Информационные системы как средство автоматизации бизнес – процессов	61
Сис-технологии	61
Системы класса MRP - CSRP	66

Глава 1

Основные понятия

Информация - это содержание сообщения, сигнала, памяти, а также сведения, содержащиеся в сообщении, сигнале или памяти.

Информационные процессы, т.е. процессы передачи, хранения и переработки информации, играют важную роль в жизни общества.

Информация всегда связана с материальным носителем, а ее передача - с затратами энергии. Однако одну и ту же информацию можно хранить в различном материальном виде и передавать с различными энергетическими затратами, причем последствия - в том числе и материальные - переданной информации совершенно не зависят от физических затрат на ее передачу. Поэтому информационные процессы не сводимы к физическим, и информация, наряду с материей и энергией, является одной из фундаментальных сущностей окружающего мира.

Достижения техники в 18 - 19 вв. практически целиком были связаны с успехами физики и химии. Благодаря им были созданы и широко распространились различные преобразователи материи и энергии: двигатели, металлургические и химические производства, электрогенераторы и т. д. Эффективность их работы описывается с помощью физических понятий: мощности, к. п. д., грузоподъемности, количества вырабатываемой энергии и др. В 20 в. с развитием техники появились устройства другого рода: средства связи, устройства автоматики, а с 40-х гг. - *вычислительной техники*. Выяснилось, что эффективность их работы с помощью физических понятий описать невозможно и что существенные характеристики таких устройств нужно описывать совсем другими способами. В результате возникли точное понятие информации и математическая *теория информации*.

Теория информации как самостоятельная научная дисциплина, связанная с восприятием, передачей и переработкой, хранением и использованием информации, была основана американским ученым *К. Шенноном* в конце 40-х гг. 20-го в. Предложенная Шенноном теория основывалась на фундаментальном понятии количественной меры неопределенности - *энтропии* - и связанного с нею понятия - *количества информации*. Другим фактором в становлении теории информации стало осознание того, что носитель информации - *сигнал* - имеет случайную природу.

На основе понятий энтропии и количества информации в теории информации введены важные характеристики сигналов и *информационных систем*.

Приведем основные из них.

1. Скорость создания информации ***H*** - энтропия источника, отнесенная к единице времени.

2. Скорость передачи информации ***R*** - количество информации, передаваемое по каналу связи в единицу времени (например, для речи в обычном темпе - около 20 бит/с).

3. Избыточность - свойство сигналов, состоящее в том, что каждый элемент сигнала (например, символ в тексте) несет информации меньше, чем может нести потенциально. При отсутствии помех избыточность вредна, так как снижает эффективность использования системы (снижает скорость передачи по каналу связи, увеличивает требуемый объем памяти при запоминании, увеличивает число операций при обработке и пр.). Вместе с тем избыточность - единственное средство борьбы с помехами, так как именно она позволяет установить, какие символы были испорчены шумами, и восстановить переданный сигнал. Избыточность измеряется по формуле:

$$r = \frac{n - n_0}{n}$$

где *n* - текущая длина (число символов) сигнала, а *n₀* - длина сигнала при максимальной информационной загрузке, т.е. минимальная из длин сигналов, несущих ту же информацию.

4. Пропускная способность канала связи ***C*** - максимальная скорость передачи информации: $C = \max R$, где максимум отыскивается среди всех мыслимых приемно-передающих систем, связанных данным каналом (К. Шеннон рассматривал разные способы кодирования при фиксированных мощностях сигналов и шумов). Экспериментально установлено, что пропускная

способность зрения и слуха человека около 5 бит/с.

Средства связи, какие бы физические процессы они ни использовали, - это средства передачи информации. Объединение понятий "информация" и "управление" привело *Н. Винера* в 40-х гг. к созданию *кибернетики*, которая, в частности, впервые указала на общность информационных процессов в технике, обществе и живых организмах.

Математическое понятие информации связано с ее измерением. В теории информации принят энтропийный подход, который учитывает ценность информации, содержащейся в сообщении для его получателя, и исходит из следующей модели. Получатель сообщения имеет определенные представления о возможных наступлениях некоторых событий. Эти представления в общем случае недостоверны и выражаются вероятностями, с которыми он ожидает то или иное событие. Общая мера неопределенности (*энтропия*) характеризуется некоторой математической зависимостью от совокупности этих вероятностей. *Количество информации* в сообщении определяется тем, насколько уменьшается эта мера после получения сообщения. Например, тривиальное сообщение, т.е. сообщение о том, что получателю и без того известно, не изменяет ожидаемых вероятностей и не несет для него никакой информации. Сообщение несет полную информацию о данном множестве событий, если оно целиком снимает всю неопределенность. В этом случае количество информации в нем равно исходной энтропии.

Количество информации это числовая характеристика *сигнала*, которая не зависит от его формы и содержания и характеризует степень неопределенности, которая исчезает после выбора (получения) сообщения в виде данного сигнала.

Рассмотрим пример. Пусть имеется колода, содержащая 32 различные карты. Чтобы выбрать одну из них существует 32 возможности, характеризующие исходную неопределенность ситуации. Число 32 в данном примере можно считать количеством информации, заложенном в одном выборе из 32 возможностей. Американский инженер Р. Хартли в 1928 году предложил в качестве меры неопределенности логарифм от числа возможностей:

$$H = k \cdot \log_a m \quad (1)$$

Здесь H – количество информации, k – коэффициент пропорциональности, m – число возможных выборов, a – основание логарифма. Чаще всего принимают $k = 1$ и $a = 2$. В этом случае единицей количества информации будет выбор из двух возможностей. Такая единица носит наименование бита и представляется одним символом двоичного алфавита.

Величина H в соотношении (1) определяет число вопросов, ответы на которые позволяют выбрать одну из двух альтернатив. В примере с колодой из 32 карт необходимо и достаточно получить ответы “да” или “нет” на пять вопросов, каждый из которых вдвое сокращает область дальнейшего выбора. Такими вопросами могут быть:

1. Карта красной масти?
2. Трефы?
3. Одна из четырех старших?
4. Одна из двух старших?
5. Дама?

Для выбора дамы пик ответами будут “нет”.”нет”.”да”.”нет”.”да” или последовательность из пяти двоичных символов 00101.

В случае, когда выборы не равновероятны вместо формулы (1) используется формула К. Шеннона. В этом случае H зависит не только от m , но и от вероятностей выбора символов и вероятностей связей между ними.

Для количества собственной или индивидуальной информации К.Шеннон в 1948 году предложил соотношение:

$$h_i = \log_2 \frac{1}{P_i} = -\log_2 P_i \quad (2)$$

где P вероятность выбора i – го символа алфавита.

В качестве меры количества информации удобнее пользоваться не h_i , а средним значением количества информации, приходящимся на один символ алфавита:

$$H = \sum_1^m P_i h_i = - \sum_1^m P_i \log_2 P_i \quad (3)$$

Значение H достигает максимума при равенстве всех P , т.е. при $P = 1/m$. В этом случае соотношение (3) превращается в формулу Р. Хартли (1).

$$H_{\max} = -\log_2 P = \log_2 m$$

В технике часто используют более простой и грубый способ измерения информации, который можно назвать объемным. Он основан на подсчете числа символов в сообщении, т.е. связан с его длиной и не учитывает содержания. Длина сообщения зависит от числа различных символов, употребляемых для записи сообщения, т.е. от мощности алфавита. Например, одно и то же число "девятнадцать" в десятичном алфавите записывается двумя символами - 19, а в двоичном алфавите - пятью символами - 10111. В вычислительной технике применяются две стандартные единицы измерения: бит и байт. Бит - это один символ двоичного алфавита. Байт - это один символ, который можно представить восьмиразрядным двоичным кодом; мощность алфавита этого представления равна числу различных восьмиразрядных двоичных кодов, т.е. $2^{**}8 = 256$, и может включать, например, все символы клавиатуры компьютера.

Эти два способа измерения информации, как правило, не совпадают, причем энтропийное количество информации не может быть больше числа двоичных символов (битов) в сообщении. Если же оно меньше этого числа, то говорят, что сообщение избыточно. В примере с колодой из 32 карт оба способа измерения информации дают одно и то же число 5. Это значит, что если кодировать карты колоды последовательностями из пяти двоичных символов, то такой код будет неизбыточным. Тривиальные сообщения всегда избыточны, так как имеют нулевую информацию с точки зрения энтропии, но содержат ненулевое число символов.

При любых видах работы с информацией всегда идет речь о ее представлении в виде определенных символических структур. Наиболее распространены одномерные представления информации, при которых сообщения имеют вид последовательностей символов. Так информация представляется в письменных текстах, при передаче по каналам связи, в памяти компьютера. Однако широко используются и многомерные представления информации, причем под многомерностью понимают не только расположение элементов информации на плоскости или в пространстве (в виде рисунков, схем, *графов*, объемных макетов и т. д.), но и множественность признаков используемых символов. Например, информацию могут нести не только значения букв и цифр, но и их цвет, размер, вид шрифта.

Формирование представления информации называется ее кодированием. Часто термин "кодирование" употребляется в более узком смысле, как переход от исходного представления к представлению, удобному для хранения, передачи или обработки. В этом случае обратный переход к исходному представлению называется декодированием.

При кодировании могут ставиться разные цели и соответственно применяться разные методы. Наиболее распространенные цели кодирования - это экономность, т.е. уменьшение избыточности сообщения; повышение скорости передачи или обработки; надежность, т.е. защита от случайных искажений; сохранность, т.е. защита от нежелательного доступа к информации; удобство физической реализации (например, двоичное кодирование информации в компьютере); удобство восприятия. Эти цели часто противоречат друг другу. Экономные сообщения могут оказаться ненадежными, так как они не содержат лишних символов, и искажение любого символа может изменить смысл сообщения.

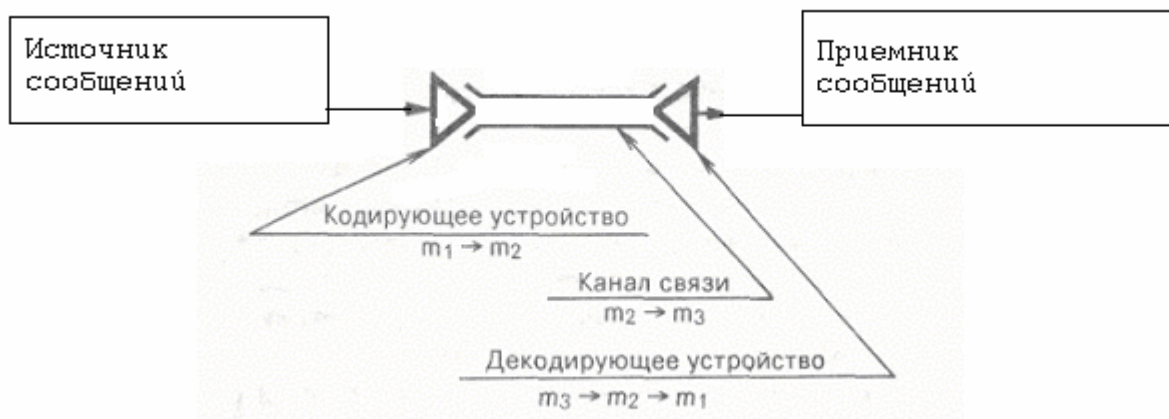
Например, обычная запись чисел цифрами гораздо экономнее и удобнее для вычислений, чем запись словами. Однако искажение или удаление любой цифры изменяет величину числа. Поэтому в финансовых документах, где надежность крайне важна, цифровые представления чисел в некоторых местах заменяются или дублируются словесными представлениями. (Сумму иногда пишут прописью.) *Теория кодирования* подробно исследует проблемы разумного сочетания экономности и надежности при передаче информации. Экономное сообщение может повысить скорость обработки (более короткое сообщение будет быстрее передано или прочтено), но может и уменьшить ее. Например, очень экономно использовать вместо фамилий и имен коды. Но тогда надо либо помнить все соответствия между фамилиями и кодами, либо каждый раз обращаться к справочнику. Защита информации от нежелательного доступа требует специальных способов

обеспечения секретности (паролей, ключей и т.д.). Это увеличивает объем хранимой информации и замедляет работу с ней.

На разных этапах сложного информационного процесса достигаются разные цели, и поэтому информация неоднократно перекодируется, т.е. изменяет свое представление.

Задачи кодирования и декодирования сообщений, поступающих к потребителям и посылаемых из источников информации теория кодирования решает с учетом наилучшего согласования посылаемой информации с каналами связи (*каналами передачи данных*). Под этим понимается стремление максимально использовать пропускную способность канала, а также обеспечить передаваемым сообщениям нужную степень защищенности от помех.

Рассмотрим обобщенную схему передачи данных.



Множество M_1 описывает множество потенциально возможных сообщений, которые может передать источник информации. Из этого множества выбирается некоторое сообщение m_1 . Но прежде чем его направить в канал, оно кодируется и преобразуется в некоторое сообщение m_2 из множества M_2 , характеризующего множество закодированных сообщений.

Поскольку способов кодирования может быть много, то M_2 отличается от M_1 и установление соответствия между элементами этих множеств требует специальной договоренности.

Например, хорошо известная азбука Морзе, применяемая для кодирования букв алфавита для передачи сообщений по каналу связи, кодирует сообщения совершенно не так, как их кодирует матрос, передающий такое же сообщение с борта корабля на берег с помощью пары сигнальных флажков.

Закодированное сообщение поступает в канал связи и под воздействием помех этого канала превращается в искаженное сообщение m_3 из множества M_3 . На приемном конце канала связи необходимо провести декодирование искаженного сообщения. Сначала надо m_3 превратить в сообщение из множества M_2 . Затем, используя информацию о способах кодирования, получить то сообщение m_1 , которое было послано. Но переход от m_3 к сообщению из M_2 при сильных искажениях неоднозначен. Поэтому искаженные сообщения декодировать нелегко.

Одна из важных задач, решаемых в теории кодирования, - это разработка специальных корректирующих кодов, которые позволяют находить и исправлять ошибки, возникающие при передаче сообщений. Один из простейших кодов такого типа мог бы быть таким. Вместо четырех букв, входящих в слово "рука", источник сообщения мог бы передавать их в удвоенном виде. Тогда сообщение, поступившее в канал связи, выглядело бы как "ррууккаа". Если искажению подверглась одна буква передаваемого слова, то на приемном конце канала появилось бы сообщение "рщууккаа". Восстановление правильного сообщения значительно облегчается благодаря принятому виду кодирования. Ясно, что ошибки в трех последних позициях нет. Ошибка имеется только в первой позиции, так как буквы в ней различаются. Для исправления обнаруженной ошибки достаточно проверить две гипотезы: m_2 есть либо "щука", либо "рука". Первая гипотеза отклоняется, так как в M_2 нет слова "щука".

Данный пример показывает, что поиск и исправление ошибок требуют избыточности в

передаваемом сообщении. Желательно, чтобы эта избыточность была минимальной. В противном случае передача сообщений будет идти с низкой скоростью.

Проблема кодирования с минимальной избыточностью - одна из задач, исследуемых и решаемых в теории кодирования. Созданы многие десятки корректирующих и обнаруживающих ошибки кодов для каналов связи с различными характеристиками шумов и искажений, вносимых при передаче сообщений.

В теории кодирования фундаментальное значение имеют две теоремы, доказанные *К. Шенноном*. Первая теорема Шеннона говорит о том, что при канале, не вносящем своих помех, можно закодировать сообщения таким образом, чтобы среднее число элементов кода, приходящихся на один элемент кодируемого алфавита, было бы минимальным, (этот минимум определяется некоторым параметром H - *энтропией* источника информации, характеризующего статистические свойства источника). Такое кодирование называется эффективным статистическим кодированием.

Вторая теорема Шеннона относится к каналам с искажениями. Согласно этой теореме, для таких каналов всегда существует способ кодирования, при котором сообщения будут передаваться с любой высокой достоверностью, если только скорость передачи не превышает пропускной способности канала связи.

Специальным разделом в теории кодирования является теория арифметических кодов, используемых для повышения надежности работы вычислительных устройств. Арифметические коды позволяют обнаруживать и исправлять ошибки, возникающие в *арифметических устройствах* компьютеров. Однако, за такой сервис приходится расплачиваться лишним временем и оборудованием. Поэтому арифметические коды используются только в специальных компьютерах, предназначенных для решения ответственных задач, когда ошибки в работе машины могут привести к весьма нежелательным последствиям.

Глава 2

Применение информационных систем.

Начало этому направлению положили исследования в области документалистики и анализа научно-технической информации, которые проводились до появления компьютеров. Своего развития информационные системы достигли тогда, когда в их состав вошли компьютеры. Информационные системы решают несколько основных задач.

А. Анализ и прогнозирование потоков разнообразной информации, перемещающихся в обществе. Изучаются потоки документов с целью их минимизации, стандартизации и приспособления для эффективной обработки на вычислительных машинах, а также особенности потоков информации, протекающей через журналы, газеты, радиоканалы, телевизионные каналы и другие каналы распространения информации. Оценивается влияние распространяемой информации на научно-технический прогресс и состояние общества.

Б. Исследование способов представления и хранения информации, создание специальных языков для формального описания информации различной природы, разработка специальных приемов сжатия и кодирования информации, аннотирования объемных документов и реферирования их. В рамках этого направления развиваются работы по созданию банков данных большого объема, хранящих информацию из различных областей знаний в форме, доступной для вычислительных машин.

В. Построение различных процедур и технических средств для их реализации, с помощью которых можно автоматизировать процесс извлечения информации из документов, не предназначенных для вычислительных машин, а ориентированных на восприятие их человеком: Эти исследования тесно связаны с проблемой извлечения смысла (содержания) тех или иных документов при вводе их в банки данных и другие информационные хранилища, ориентированные на компьютеры.

Г. Создание информационно-поисковых систем, способных воспринимать запросы к информационным хранилищам, сформулированные на естественном языке, а также специальных языках запросов для систем такого типа.

Д. Создание сетей хранения, обработки и передачи информации, в состав которых входят информационные банки данных, *терминалы*, обрабатывающие центры и средства связи.

Работы в области информационных систем опираются, с одной стороны, на исследования в прикладной лингвистике, которая создает языки для записи информации и поиска ответов в информационных массивах по поступающим запросам, а с другой стороны, на теорию информации, поставляющую модели и методы, которые используются при организации циркуляции информации в каналах передачи данных.

Информационная система - это хранилище информации, снабженное процедурами ввода, поиска, размещения и выдачи информации. Наличие таких процедур - главная особенность информационных систем..

Работа информационных систем заключается в обслуживании двух встречных потоков новой информации: ввода новой информации и выдачи текущей информации по запросам. Поскольку главная задача информационной системы - обслуживание клиентов, система должна быть устроена так, чтобы ответ на любой запрос выдавался быстро и был достаточно полным. Эти требования обеспечиваются наличием стандартных процедур поиска информации и тем, что данные системы расположены в определенном порядке.

Рассмотрим, например, отдел кадров небольшого предприятия, в котором хранятся стандартные сведения о сотрудниках: год рождения, семейное положение, домашний адрес, должность, оклад. Время от времени в отдел кадров поступает информация об их изменении. Порядок размещения данных образует трехуровневую *иерархическую структуру (дерево)*: первый уровень дерева - это предприятие, второй уровень - фамилии его сотрудников, третий уровень - документы о каждом из них.

Для каждого сотрудника заводят стандартную карточку, в которую внесены нужные данные в одинаковом для всех порядке и указания на то, где хранятся соответствующие документы (приказы, справки и т.д.). Чтобы этот порядок соблюдался, при получении каждого нового документа следует проделать определенные процедуры: найти нужную карточку, занести в нее данные, найти папку и положить в нее документ.

Время от времени на каждом из уровней может произойти переполнение. Поэтому производится регулярная чистка. В информационных системах эту процедуру называют удалением мусора.

Сама идея информационных систем и некоторые принципы их организации возникли задолго до появления компьютеров. Однако компьютеризация на несколько порядков повысила эффективность информационных систем и расширила сферы их применения.

Во-первых, резко возросли скорости всех видов обработки информации: поиска и размещения (внутри компьютера), выдачи (на экран или печать), передачи и ввода (по средствам электронной и космической связи в информационные системы любой точки земного шара). Для некоторых видов информационных систем именно скорости передачи и ввода играют решающую роль. Таковы, например, автоматизированные системы продажи авиационных и железнодорожных билетов или многотерминальные системы электронной торговли ценными бумагами, где только высокая скорость ввода информации может исключить продажу мест (или акций), которые минуту назад были проданы с другого терминала.

Во-вторых, во много раз увеличились возможности хранения больших объемов информации: как за счет того, что машинные носители информации (диски и ленты) в сотни и тысячи раз компактнее бумажных носителей (бумаг и книг), так и за счет того, что только при высоких скоростях ЭВМ можно проводить поиск в таких объемах за приемлемое время.

В-третьих, благодаря использованию электронной связи и сетей ЭВМ потеряло значение расстояние между информационной системой, источниками информации и ее клиентами. Достаточно иметь *терминал, персональную ЭВМ* или другое устройство, позволяющее запрашивать и получать нужную информацию и соединенное с системой каналами связи.

Не следует думать, что высокая эффективность современных информационных систем автоматически достигается только применением современных технических средств. Для того чтобы максимально использовать их огромные возможности, нужно хорошо проработать структурные, алгоритмические и языковые вопросы, т.е. разработать структуры данных, алгоритмы обработки информации и языки общения с системой.

На примере отдела кадров продемонстрирована иерархическая структура данных. Однако в иерархической структуре не на все вопросы можно получить одинаково быстрые ответы. В рассмотренном примере по фамилии сотрудника легко найти нужные сведения о нем (его признаки). Гораздо труднее по конкретному значению признака найти сотрудников, которые им обладают, - придется перебрать все карточки. Это происходит потому, что в иерархии (дереве) данных фамилии расположены выше (на втором уровне), чем признаки. В отделе кадров гораздо чаще бывают вопросы первого типа (по фамилии найти признак); поэтому неудобства, связанные с вопросами второго типа, не так велики.

Если чаще бывают вопросы "по признаку (скажем, по году рождения) найти фамилии всех, кто им обладает", то иерархию надо строить по-другому: сгруппировать сотрудников по годам, т.е. в дереве данных расположить признак "год рождения" выше фамилии. Если же фамилии ищут по разным признакам и вопросы разных видов задаются одинаково часто, то иерархическая структура вообще оказывается неудобной. Вопросы построения различных структур данных и эффективных алгоритмов обработки информации в них рассматриваются в теории *баз данных* - теоретической основе построения информационных систем.

Языковые проблемы связаны с обеспечением удобного обращения с системой. Обращение пользователя к системе за информацией называется запросом. В простейшем случае фиксируется небольшое число возможных запросов, на которые заранее готовы ответы. Так устроены, например, железнодорожные справочные, где на пульте расположены кнопки с названиями станций. Здесь число возможных запросов равно числу кнопок. При нажатии кнопки выдается вся имеющаяся информация о поездах по данной станции.

Несколько более сложный случай - запрос, в котором зафиксирован набор признаков. Значения некоторых признаков указываются пользователем (по ним надо искать), значения остальных признаков должна указать система. Здесь естественная форма запроса - бланк (на бумаге или на экране дисплея), графы которого - стандартные, но число вариантов заполнения может быть велико.

Гораздо сложнее ситуация, когда пользователю нужен стандартный набор различных данных, удовлетворяющих определенным условиям, например, логическая комбинация признаков ("выдать фамилии всех сотрудников, родившихся в интервале с 1940 по 1950 гг., которые либо неженаты, либо живут в Московской области"), или транзитный железнодорожный маршрут с наименьшим временем ожидания поездов, или сложная цепочка обменов при обмене жилплощади. Система, способная отвечать на такие запросы, должна располагать специальным языком запросов, на котором пользователь описывает то, что ему нужно. Такой язык может содержать различные операции над признаками: логические, арифметические, операции сравнения и т. д. *Транслятор* превращает такой запрос в алгоритм поиска нужной информации.

Наряду с языком запросов (т.е. входным языком), пользователю часто требуется задать определенную форму для выходной информации. Это нужно, в частности, при машинной выдаче финансовых и отчетных документов. Для этого создаются языки выходных документов, с помощью которых пользователь указывает форму и расположение выходной информации на бумаге или экране.

Еще одна важная проблема, которую приходится решать при создании информационных систем, - это защита информации. Имеется несколько сторон этой проблемы. Во-первых, защита от помех (особенно при передаче по линиям связи) и сбоев аппаратуры. Для ее организации используются методы *теории кодирования*. Во-вторых, защита от неправильных действий некомпетентного пользователя: никакое неправильное нажатие кнопок терминала или нарушение инструкций при общении с системой не должны портить информацию в системе. И, наконец, защита от несанкционированного доступа, т.е. от пользователей, желающих получить информацию, к которой у них нет права доступа, либо исказить имеющуюся в системе информацию. Для такой защиты используются программные пароли, средства шифрования и т. д.

Программное обеспечение первых информационных систем создавалось каждый раз "с нуля": для новой системы заново строились структура данных и программы обработки информации в ней, разрабатывался язык запросов, транслятор с него и т. д. В настоящее время существуют разнообразные средства программирования информационных систем - системы управления базами данных (СУБД), в состав которых входят средства организации структуры данных, языки запросов

и выходных документов, программы ввода информации, удаления мусора и др. СУБД существенно ускоряют процесс разработки информационных систем.

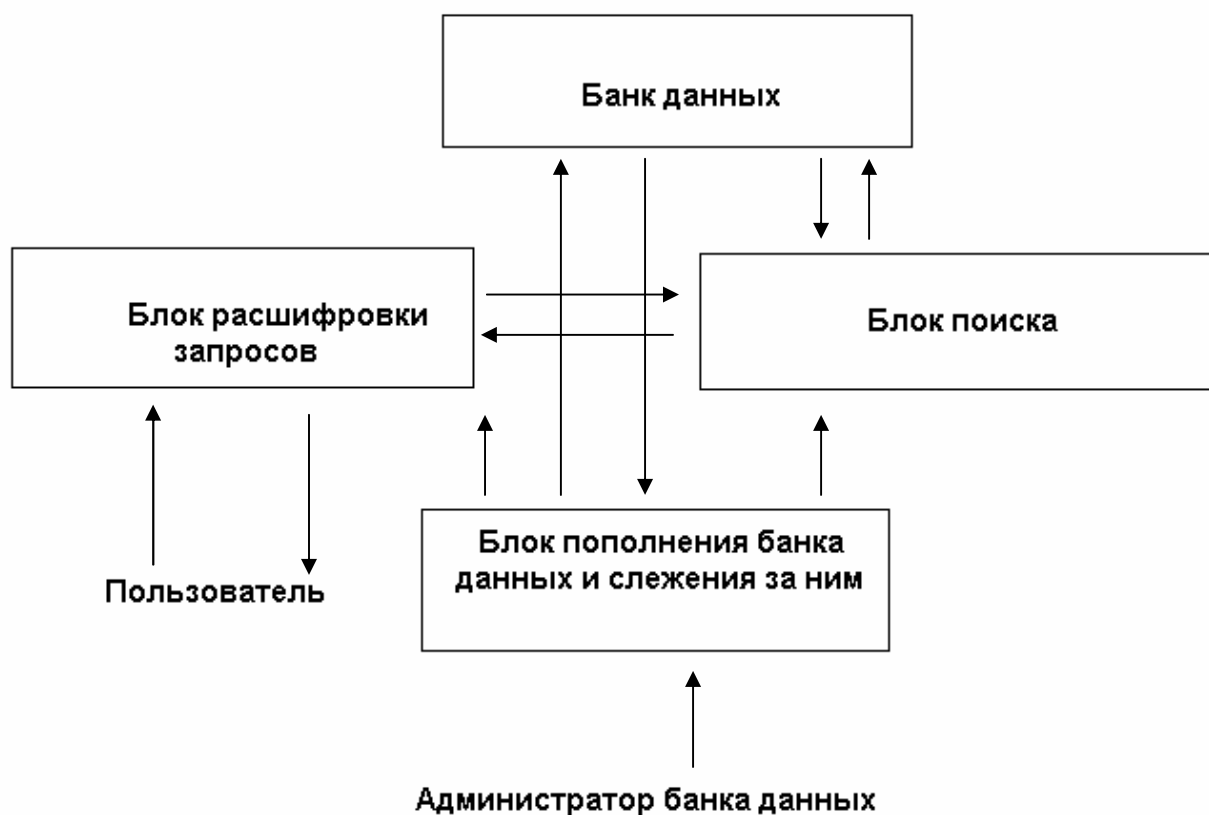
Массовое распространение ЭВМ, электронных средств связи и совершенствование СУБД ведут к постоянному расширению областей использования информационных систем. Справочные службы, библиотечные каталоги, различные картотеки - это традиционные сферы их применения. Автоматизированные информационные системы применяются также на производстве: в них вводится информация о ходе выполнения производственных заказов, о наличии инструмента, деталей, сырья на складах предприятия. В деловой сфере, где постоянно обновляются сведения о товарах и фирмах, биржевая и банковская информация, курсы валют и ценных бумаг, и в сфере обслуживания (системы продажи билетов, бронирования мест в гостиницах) они также необходимы.

Имея персональный компьютер и несложную СУБД, человек, не знающий программирования, может создать собственную информационную систему. Современные средства связи позволяют подключить компьютер к соответствующим каналам связи и стать абонентом различных информационных систем.

В *информационном обществе* роль информационных систем как в трудовой деятельности, так и в быту возрастет еще более. Людей, разрабатывающих, обслуживающих и использующих такие системы, станет гораздо больше, чем людей, изготавливающих и применяющих материальные орудия труда.

Автоматизированные информационные системы (АИС)

В этот класс систем прикладной информатики входят информационные системы, общая структура которых показана на рисунке.



Основу такой системы составляет *банк данных*, в котором хранится большая по объему информация о какой-либо области человеческих знаний. Территориально этот банк может быть распределенным. Важно, что для пользователя этот банк представляется как единое хранилище информации, куда он может обратиться с запросом

В неавтоматизированных *информационных системах* запрос обрабатывается человеком. В АИС запросы обрабатывают с помощью специальных программ, реализуемых на компьютере. Для того чтобы это стало возможным, необходим специальный язык запросов, понимаемый АИС. Для

пользователя желательно, чтобы язык запросов был как можно ближе к естественному языку. В этом случае АИС должна обладать системой понимания текстов на естественном языке или языке, близком к нему. В таких АИС используется *диалоговая система*.

Устройство блока расшифровки запроса зависит от выбранного языка запроса. Оно достаточно просто, если запросы жестко фиксированы. Если же в качестве языка запроса используется подмножество естественного языка, то этот блок может быть достаточно сложным.

После расшифровки запроса по информации, содержащейся в нем, формируется поисковое предписание (или поисковый образ), представляющее задание для процедуры поиска в банке данных. Поиск в банке данных осуществляется блоком поиска. Найденная информация выдается потребителю в удобной для него форме.

Банк данных требует постоянного обновления, пополнения и чистки. Для этого используется специальный входной канал, которым пользуется администратор банка данных.

Частным случаем АИС являются *информационно-поисковые системы*, предназначенные как для коллективного, так и для индивидуального пользования.

В таких системах хранится информационный массив, из которого по требованиям пользователей выдается нужная информация. Поиск информации по требованию пользователя осуществляется либо автоматически, либо вручную (как в библиотеках, когда с запросом к работнику справочного фонда обращается читатель, а работник пользуется системой каталогов).

Во втором случае используются ЭВМ, снабженные специальными программными средствами, анализирующими процессы запросов, поиска и выдачи нужных документов.

Информационно-поисковые системы делятся на два типа.

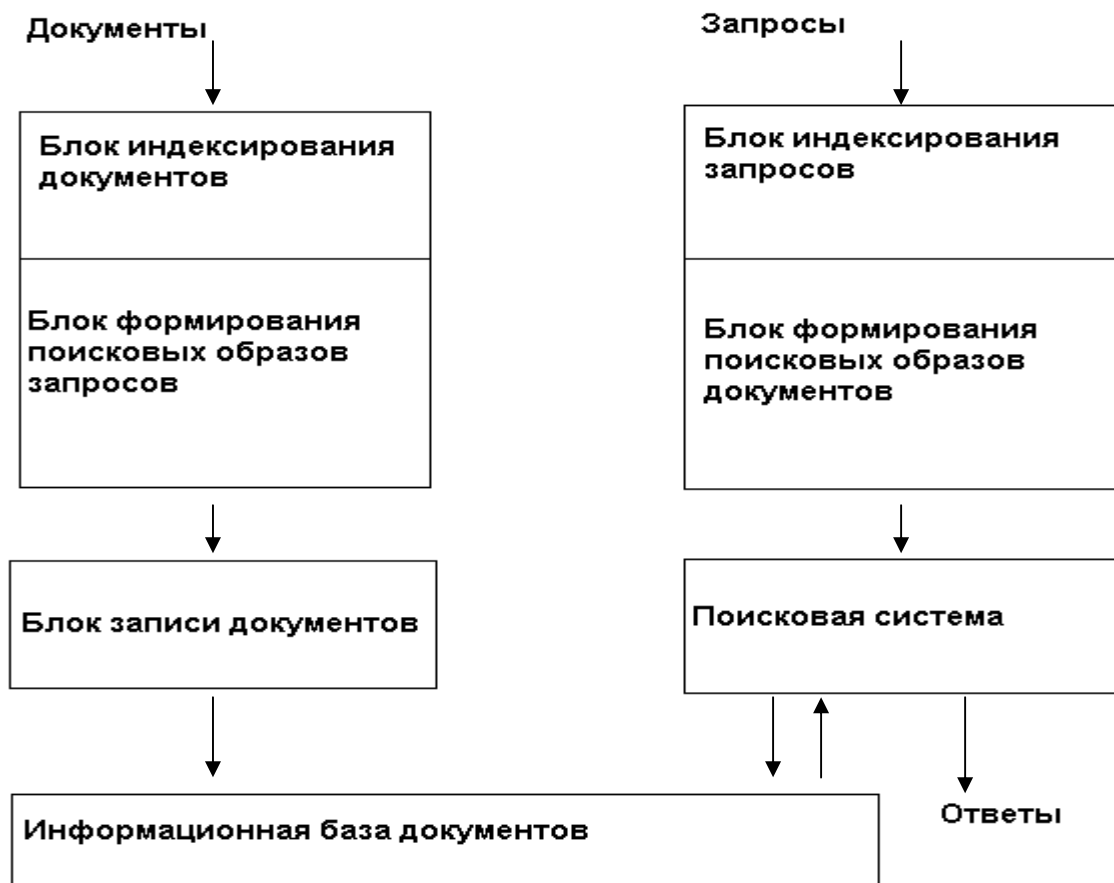
1. Документографическая ИПС. В такой ИПС все хранимые документы индексируются некоторым специальным образом. Каждому документу (статье, отчету, протоколу и т.п.) присваивается индивидуальный код, составляющий поисковый образ документа. Поиск идет не по самим документам, а по их поисковым образам, которые содержат информацию (адрес) о местонахождении документа. Именно так ищут книги по заказам читателя в больших библиотеках (в маленьких библиотеках библиотекарь обычно ищет книги сам). По требованию читателя сначала находят карточку в каталоге, а потом по шифру, указанному на ней, отыскивается и сама книга.

Различия документографических ИПС определяются тем, как устроен поисковый образ документа. В простейшем случае это просто его индивидуальное название (например, название, автор, год издания книги). В более сложных случаях нет однозначного соответствия между поисковым образом документа и самим документом. Вполне возможен случай, когда поисковый образ документа соответствует нескольким различным документам и, наоборот, один и тот же документ соответствует не одному, а нескольким поисковым образам.

Такой неоднозначностью обладают, например, поисковые образы документов в дескрипторных системах. Дескриптор - слово или словосочетание, которое тесно связано с содержанием документа. Совокупность дескрипторов определяет группу документов со сходным содержанием. В последнее время журналы, публикующие научные статьи, требуют от своих авторов, чтобы для каждой статьи они указывали список ключевых слов, которые и играют роль дескрипторов.

По набору этих ключевых слов (набору дескрипторов) можно найти статью среди всех статей, введенных в ИПС дескрипторного типа.

Общая блок-схема ИПС дескрипторного типа показана на рисунке.



Эта схема имеет два входа. По одному происходит пополнение хранящегося в системе информационного массива документов, а по второму поступают запросы пользователей.

2. Фактографическая ИПС. В отличие от документографических ИПС в ИПС такого типа хранятся не документы, а факты, относящиеся к какой-либо *предметной области*. Хранимые факты могут быть извлечены из различных документов. В базе фактов они связываются между собой системой разнообразных отношений. Такая сеть в ИПС носит название тезауруса предметной области. Запросы, поступающие в фактографические ИПС, используют тезаурус для поиска ответов на запросы. Поиск осуществляется методом *поиска по образцу*, широко применяющемуся в *базах знаний* систем *искусственного интеллекта*.

ИПС фактографического типа постепенно приближаются по своей организации и функционированию к развитым *базам данных* и знаний.

Любая ИС включает в себя четыре компонента: информационные средства, программные средства (обеспечение), технические средства, персонал. Информационное обеспечение реализуется в виде файловой системы или в виде базы данных. **База данных (БД)** – это совокупность описаний объектов предметной области и связей между ними, актуальных для конкретной предметной области.

Структура данных в ИС обычно сложна (сложность определяется не столько объемом, сколько количеством взаимосвязей), задачи по обработке данных однотипны для разных предметных областей (создание, поиск, ввод и вывод, группировка, сортировка). Поэтому все типовые функции по работе с данными выделены в специальную систему. **Система управления базой данных** (СУБД – это комплекс программных и языковых средств создания, ведения и манипулирования данными).

Программные средства делят на две части: системное программное обеспечение (СПО) и прикладное программное обеспечение (ППО). В состав СПО входит операционная система ЭВМ (ОС) и СУБД. ОС настолько тесно связана с техническими средствами, что их часто объединяют и называют **программно-аппаратной платформой**, например, ПЭВМ IBM PC используется платформа WINTEL (Windows + Intel).

Персонал – это специалисты, которые обслуживают и сопровождают ИС, их часто включают в состав системы, поскольку без персонала невозможна работы сложной системы.

Жизненный цикл ИС

В основе деятельности по созданию и использованию ИС лежит понятие жизненного цикла.

Жизненный цикл – это модель создания и использования ИС, отражающая ее различные состояния, начиная с момента возникновения необходимости в данном комплексе средств и заканчивая моментом его полного выхода из употребления у пользователей.

Опыт создания и использования ИС позволяет условно выделить следующие основные этапы их жизненного цикла:

- анализ – определение того, что должна делать система;
- проектирование – определение того, как система будет делать то, что она должна делать. Проектирование это, прежде всего, спецификация подсистем, функциональных компонентов и способов их взаимодействия в системе;
- разработка – создание функциональных компонентов и подсистем по отдельности, соединение подсистем в единое целое;
- тестирование – проверка функционального и параметрического соответствия системы показателям, определенным на этапе анализа ;
- внедрение – установка и ввод системы в действие;
- сопровождение – обеспечение штатного процесса эксплуатации системы на предприятии заказчика.

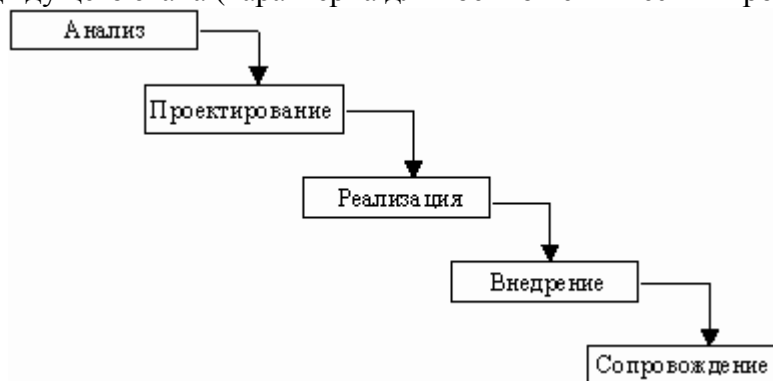
Этапы разработки, тестирования и внедрения ИС обозначаются единым термином – реализация.

Жизненный цикл образуется в соответствии с принципом нисходящего проектирования и, как правило, носит итерационный характер: реализованные этапы, начиная с самых ранних, циклически повторяются в соответствии с изменениями требований и внешних условий, введением дополнительных ограничений и т.п.

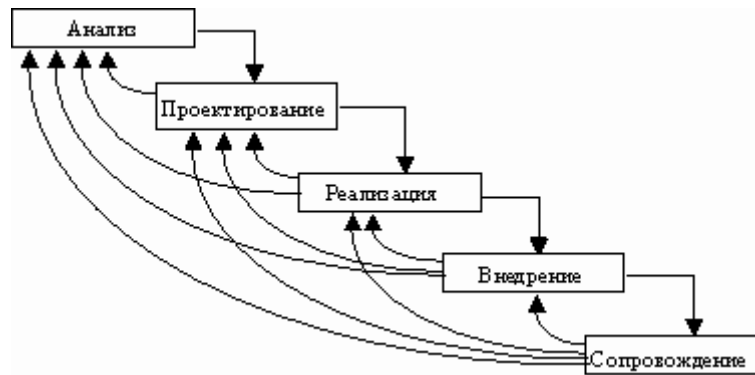
На каждом этапе жизненного цикла порождается определенный набор технических решений и отражающих их документов, при этом для каждого этапа исходными являются документы и решения, принятые на предыдущем этапе.

Существующие модели жизненного цикла, определяют порядок исполнения этапов в процессе создания ИС, а также критерии перехода от этапа к этапу. В соответствии с этим наибольшее распространение получили три следующие модели.

Каскадная модель – предполагает переход на следующий этап после полного завершения работ предыдущего этапа (характерна для военно-технических проектов).



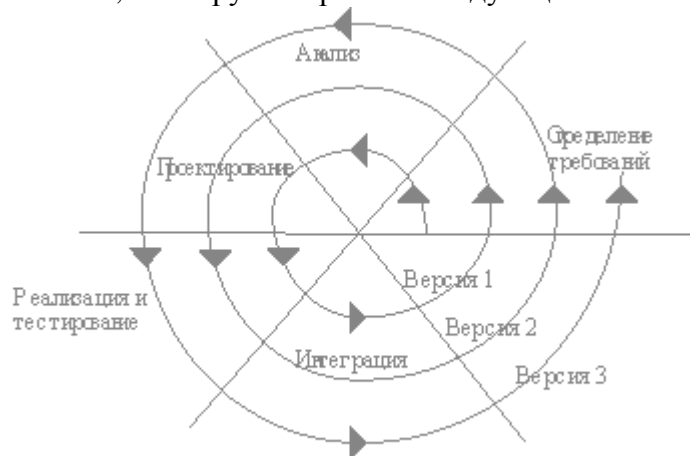
Каскадный подход хорошо зарекомендовал себя при построении ИС, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования, с тем чтобы предоставить разработчикам свободу реализовать их как можно лучше с технической точки зрения. В эту категорию попадают сложные расчетные системы, системы реального времени и другие подобные задачи. Однако, в процессе использования этого подхода обнаруживается ряд его недостатков, вызванных прежде всего тем, что реальный процесс создания ПО никогда полностью не укладывается в жесткую схему. В процессе создания ПО возникает потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений.



Поэтапная итерационная модель. Эта модель создания ИС предполагает наличие циклов обратной связи между этапами.

Преимущество такой модели заключается в том, что межэтапные корректировки обеспечивают большую гибкость и меньшую трудоемкость по сравнению с каскадной моделью. Однако время жизни каждого из этапов может растянуться на весь период создания системы.

Спиральная модель – опирается на начальные этапы жизненного цикла: анализ, предварительное и детальное проектирование. Каждый виток спирали соответствует поэтапной модели создания фрагмента или версии системы, на нем уточняются цели и характеристики проекта, определяется его качество, планируются работы следующего витка спирали.



Основная проблема спирального цикла - определение момента перехода на следующий этап. Для ее решения необходимо ввести временные ограничения на каждый из этапов жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. План составляется на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков.

Нерешенные вопросы и ошибки, допущенные на этапах анализа и проектирования ИС, порождают на последующих этапах трудные, часто неразрешимые проблемы и, в конечном счете, приводят к неуспеху всего проекта.

Главная особенность разработки современных ИС состоит в концентрации усилий на двух начальных этапах ее жизненного цикла - анализе и проектировании, при относительно невысокой сложности и трудозатратах на последующих этапах.

Эффективность ИС

Для оценки эффективности ИС служит набор критериев, которые количественно определяют степень соответствия системы целям ее создания. Критерий эффективности должен быть наглядным, напрямую зависеть от работы системы, допускать приближенную оценку по результатам экспериментов. Оценивают как ИС в целом, так и ее компоненты. Одновременное достижение всех целей невозможно, поэтому на практике выбирают компромиссное решение: один из критериев оптимизируется, а остальные служат в качестве ограничений. Приведем типичные цели создания ИС и критерии для их оценки:

№	Цели	Критерии
1	Максимальная полнота отображения информации	Отношение объема информации в системе к объему информации на объекте
2	Максимальная скорость предоставления информации	Время обработки данных Время ответа на запрос
3	Максимальное удобство пользователя	Время на формирование запроса и понимание ответа
4	Минимальные расходы	Капитальные вложения + Текущие затраты
5	Максимальное извлечение полезной информации	Отношение объема входной информации к объему выходной информации
6	Минимальная избыточность базы данных	Отношение объема избыточной информации к объему хранимой информации

Пользователи ИС. Трехуровневое представление данных

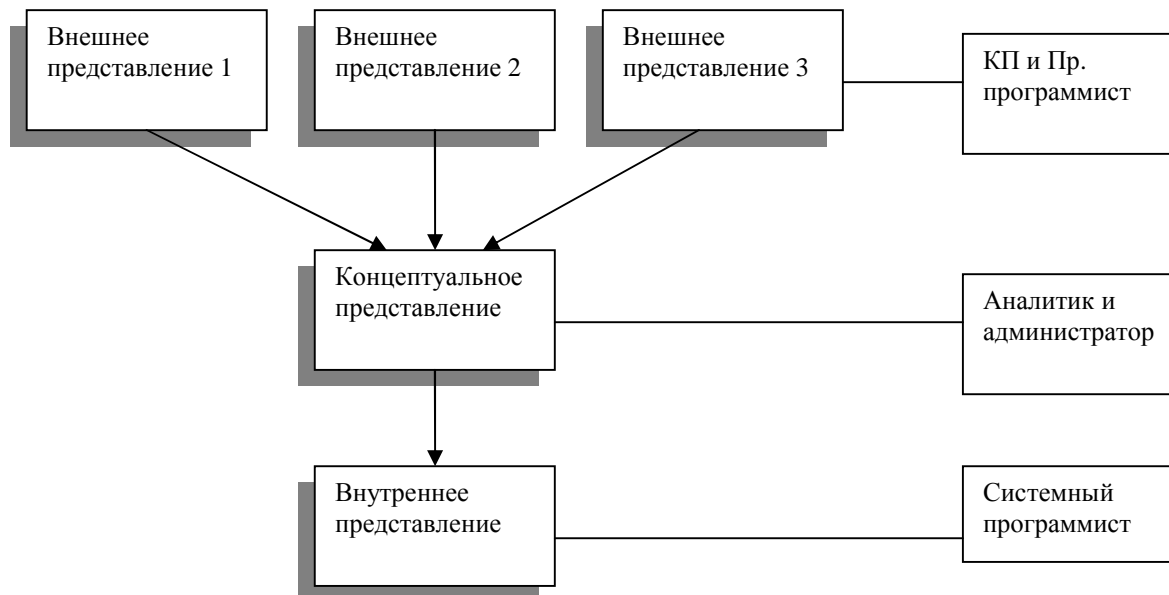
Пользователей ИС можно разделить на следующие группы:

- случайный пользователь, взаимодействие которого с ИС не обусловлено служебными обязанностями;
- конечный пользователь (потребитель информации) - лицо или коллектив, в интересах которых работает ИС. Он работает с ИС повседневно, связан с жестко ограниченной областью деятельности и, как правило, не является программистом, например, это может быть бухгалтер, экономист, руководитель подразделения;
- коллектив специалистов (персонал ИС), включающий администратора банка данных, системного аналитика, системных и прикладных программистов.

Рассмотрим более подробно состав и функции персонала ИС.

Администратор - это специалист (или группа специалистов), который понимает потребности конечных пользователей, работает с ними в тесном контакте и отвечает за определение, загрузку, защиту и эффективность работы банка данных. Он должен координировать процесс сбора информации, проектирования и эксплуатации БД, учитывать текущие и перспективные потребности пользователей. **Системные программисты** занимаются разработкой и сопровождением базового математического обеспечения ЭВМ (ОС, СУБД, трансляторов, сервисных программ общего назначения). **Прикладные программисты** разрабатывают программы для реализации запросов к БД. **Аналитик** строит математическую модель предметной области, исходя из информационных потребностей конечных пользователей; ставит задачи для прикладных программистов. На практике персонал небольших ИС часто состоит из одного - двух специалистов, которые выполняют все перечисленные функции.

Для разных классов пользователей можно выделить несколько уровней представлений об информации в ИС.



Внешнее представление данных - это описание информационных потребностей конечного пользователя и прикладного программиста. Связь между этими двумя видами внешнего представления осуществляет аналитик.

Концептуальное представление данных - отображение знаний обо всей предметной области ИС. Это наиболее полное представление, отражающее смысл информации, оно может быть только одно и не должно содержать противоречий и двусмысленностей. Концептуальное представление - это сумма всех внешних представлений, которое учитывает перспективы развития ИС, знания о методах обработки информации, знания о структуре самой ИС и др.

Принято различать две формы концептуального представления информации: инфологическую (информационно - логическую) модель которая не привязана к конкретной реализации и ориентирована на пользователя; и даталогическую модель, которая учитывает требования конкретной СУБД.

Внутреннее (физическое) представление - это организация данных на физическом носителе информации. Этот уровень характеризует представления системных программистов и практически используется только тогда, когда СУБД не обеспечивает требуемого быстродействия или специфического режима обработки данных.

Таким образом, многоуровневое представление об информации в ИС обусловлено потребностями различных групп пользователей и уровнем развития инструментальных средств создания ИС. Оно позволяет разделить работу по созданию и обслуживанию ИС на относительно независимые части.

История и основные направления развития ИС

Понятие информационной системы (ИС) на протяжении своего существования претерпело значительные изменения. Условно можно выделить три поколения ИС. Рассмотрим основные характеристики компонентов этих ИС.

Первое поколение (до 70-х годов) предназначалось для решения установившихся задач, которые четко определялись на этапе создания системы и затем практически не изменялись.

Основные черты 1-го поколения ИС:

- Техническое обеспечение систем составляли маломощные ЭВМ 2-3 поколения.
- Информационное обеспечение (ИО) представляло собой массивы (файлы) данных, структура которых определялась той программой, в которой они использовались.
- Программное обеспечение специализированные прикладные программы, например, программа начисления заработной платы.
- Архитектура ИС - централизованная. Как правило, применялась пакетная обработка задач. Конечный пользователь не имел непосредственного контакта с

ИС, вся предварительная обработка информации и ввод производились персоналом ИС.

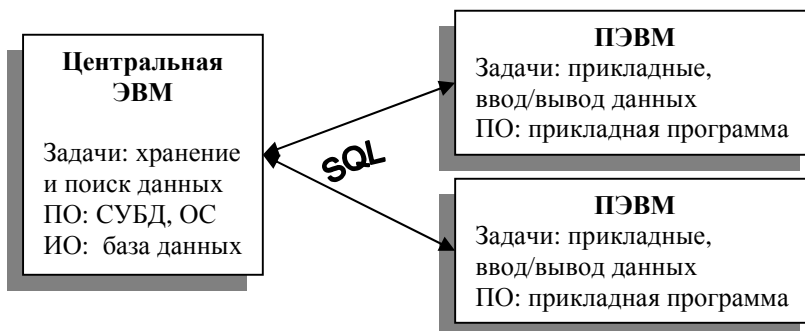
Недостатки ИС 1-го поколения:

- Сильная взаимосвязь между программами и данными, то есть изменения в предметной области приводили к изменению структуры данных, а это заставляло переделывать программы.
- Трудоемкость разработки и модификации систем.
- Сложность согласования частей системы, разработанных разными людьми в разное время.

Второе поколение. Стремление преодолеть недостатки 1-го поколения ИС породило в 70-х годах технологию баз данных. База данных создается для группы взаимосвязанных задач, для многих пользователей и это позволяет частично решить проблемы 1-го поколения ИС. Вначале СУБД разрабатывались для больших ЭВМ, и их количество не превышало десятка. Благодаря появлению ПЭВМ технология БД стала массовой, создано большое количество инструментальных средств и СУБД для разработки ИС, что в свою очередь вызвало появление большого количества прикладных ИС в прикладных областях.

Основные черты 2-го поколения ИС:

- Основу ИО составляет база данных,
- Программное обеспечение состоит из прикладных программ и СУБД.
- Технические средства: ЭВМ 3-4 поколения и ПЭВМ.
- Средства разработки ИС: процедурные языки программирования 3-4 поколения, расширенные языком работы с БД (SQL, QBE).
- Архитектура ИС: наиболее популярны две разновидности: персональная локальная ИС, централизованная БД с сетевым доступом.



Большим шагом вперед явилось развитие принципа "дружественного интерфейса" по отношению к пользователю (как к конечному, так и к разработчику ИС). Например, повсеместно применяется графический интерфейс, развитые системы помощи и подсказки пользователю, разнообразные инструменты для упрощения разработки ИС: системы быстрой разработки приложений (RAD-системы), средства автоматизированного проектирования ИС (CASE- средства).

К концу 80-х годов выявились и недостатки систем 2-го поколения:

- большие капиталовложения в компьютеризацию предприятий не дали ожидаемого эффекта, соответствующего затратам (увеличились накладные расходы, но не произошло резкого повышения производительности);
- внедрение ИС столкнулось с инертностью людей, нежеланием конечных пользователей менять привычный стиль работы, осваивать новые технологии;
- к квалификации пользователей стали предъявляться более высокие требования (знание персонального компьютера, конкретных прикладных программ и СУБД, способность постоянно повышать свою квалификацию).

В связи с этим постепенно стало формироваться 3-е поколение ИС. Рассмотрим основные **черты современного поколения ИС**.

Техническая платформа - мощные ЭВМ 4-5 поколения, использование разных платформ в одной ИС (большие ЭВМ, мощные стационарные ПК, мобильные ПК). Наиболее характерно

широкое применение вычислительных сетей - от локальных до глобальных.

Информационное обеспечение: ведутся интенсивные разработки с целью повышения интеллектуальности банка данных в следующих направлениях:

- новые модели знаний, учитывающие не только структуру информации, но и активный характер знаний,
- средства оперативного анализа информации (OLAP) и средства поддержки принятия решений (DSS),
- новые формы представления информации, более естественные для человека (мультимедиа, полнотекстовые БД, гипертекстовые БД, средства восприятия и синтеза речи).

Программное обеспечение: новым является появление и развитие открытой компонентной архитектуры ИС. Компонент - это программа, выполняющая какой-либо осмысленный с точки зрения конечного пользователя набор функций и имеющая открытый интерфейс. ПО ИС собирается из готовых компонентов. Компонент может функционировать на разных типах ЭВМ и связь между компонентами устанавливается не на этапе компиляции, а в реальном масштабе времени. Такой принцип построения позволяет использовать накопленный опыт программистов, ускорять разработку ИС, создавать распределенные ИС.

Архитектура ИС - разнообразна в связи с многоплатформенностью. Развивается трехступенчатая модель ИС.



Благодаря такому построению снижаются требования к клиентским машинам и общая стоимость системы, повышается общая эффективность и производительность. Узким местом является пропускная способность и надежность вычислительных сетей.

Методы разработки ИС: при традиционном подходе сначала выявлялись информационные потоки на предприятии, а затем к этой структуре привязывалась ИС, повторяя и закрепляя тем самым недостатки организации бизнеса. В 90-93 г.г. бурно обсуждалась идея бизнес - реинжиниринга, предложенная М. Хаммером. Она состоит в том, что для получения существенного эффекта от ИС необходимо одновременно с разработкой ИС пересмотреть и бизнес-процессы, удалив и упростив некоторые из них.

Таким образом, современная корпоративная ИС должна создаваться как часть предприятия, включающая бизнес-архитектуру, персонал и информационные технологии.

Классификация ИС

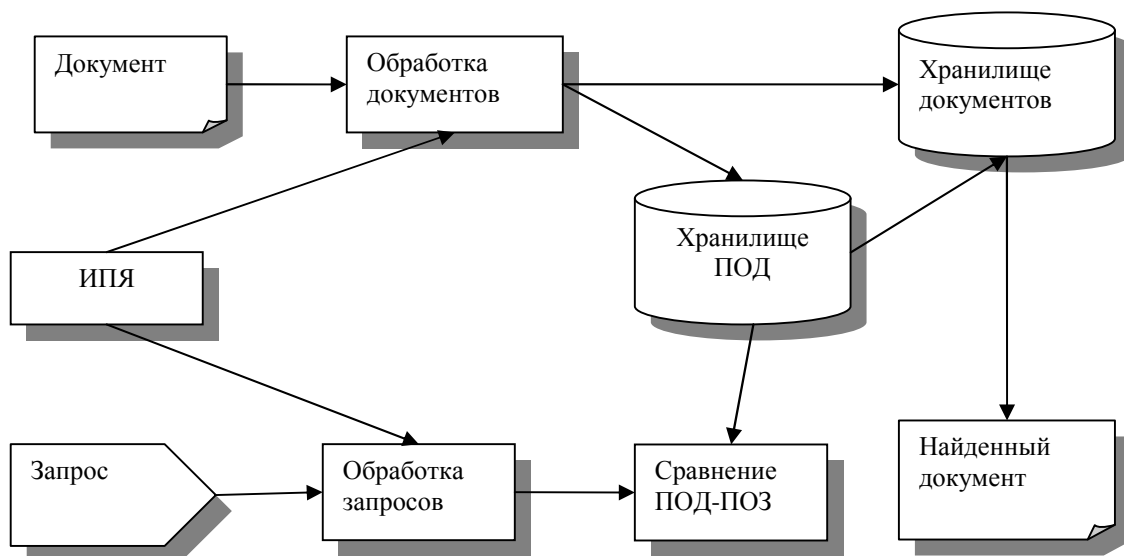
По режиму работы ИС делятся на пакетные, диалоговые и смешанные. Пакетные ИС работают в пакетном режиме: вначале данные накапливаются и формируется пакет данных, а затем пакет последовательно обрабатывается рядом программ. Недостаток этого режима - низкая оперативность принятия решений и обособленность пользователя от системы. Диалоговые ИС работают в режиме обмена сообщениями между пользователями и системой (например, система продажи авиабилетов). Этот режим особенно удобен, когда пользователь может выбирать перспективные варианты из числа предлагаемых системой.

По способу распределения вычислительных ресурсов ИС делятся на локальные и распределенные. Локальные ИС используют одну ЭВМ, а в распределенных ИС взаимодействуют несколько ЭВМ, связанных сетью. Отдельные узлы сети обычно территориально удалены друг от друга, решают разные задачи, но используют общую информационную базу.

По функциям различают три вида ИС: информационно - поисковые системы (ИПС), системы обработки данных (СОД) и автоматизированные системы управления (АСУ).

Системы обработки данных (СОД) предназначены для решения задач типа расчета заработной платы, статистической отчетности и т.п. Такие системы наряду с функциями ввода, выборки, коррекции информации выполняют математические расчеты без применения методов оптимизации. АСУ отличается от СОД тем, что сама выполняет управленческие функции по отношению к объекту. В АСУ включаются прикладные программы для принятия и оптимизации управленческих решений. Примером АСУ является система для оптимального управления запасами материалов на складе.

Информационно-поисковые системы (ИПС) предназначены для поиска требуемого документа или факта в множестве документов.



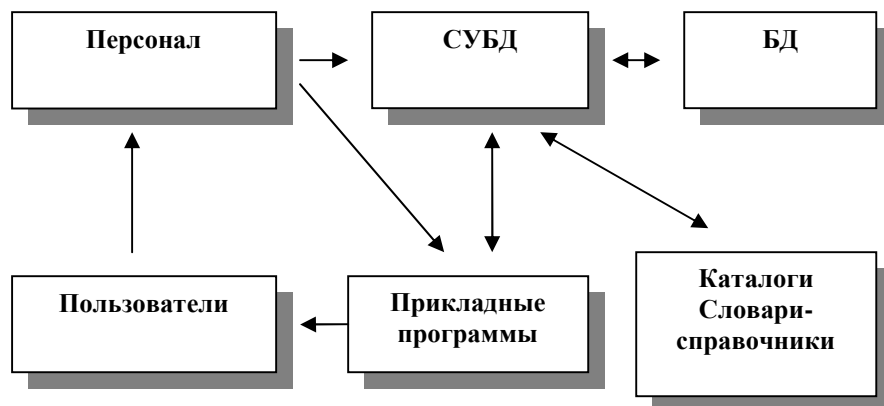
Поисковый образ документа (ПОД) получается в результате процесса индексирования, который состоит из двух этапов: выявление смысла документа и описание смысла на специальном информационно-поисковом языке (ИПЯ).

Запрос к ИПС описывается также на этом языке. Поиск документа состоит в сравнении множества хранящихся в системе ПОД и текущего поискового образа запроса (ПОЗ), в результате чего пользователю выдается требуемый документ или отказ. Различают два режима работы ИПС: текущее информирование пользователей о новых поступлениях и ретроспективный поиск по разовым запросам [6].

По концепции построения ИС делятся на файловые системы, автоматизированные банки данных (АБД), интеллектуальные банки данных (банки знаний) и хранилища данных.

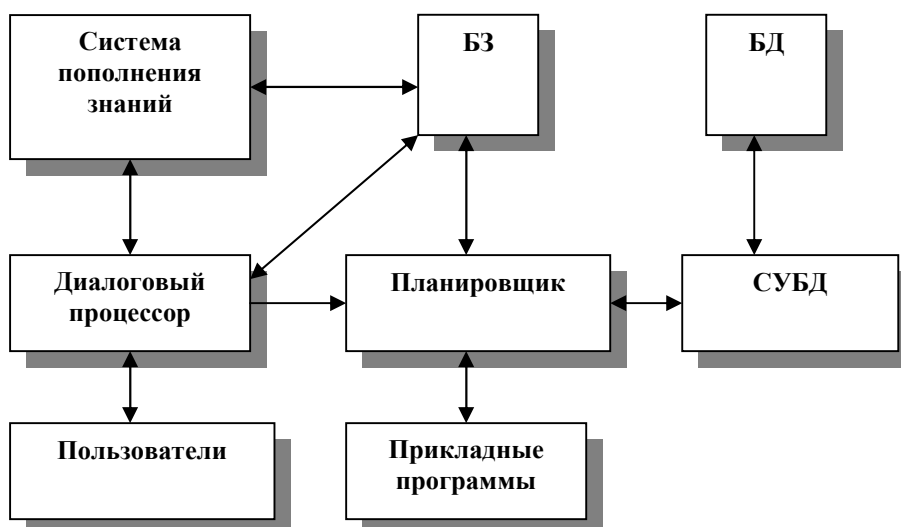
Информационное обеспечение ИС первого типа построено в виде файловых систем. В современных ЭВМ операционная система берет на себя распределение внешней памяти, отображение имен файлов в соответствующие адреса во внешней памяти и обеспечение доступа к данным. Программное обеспечение ИС напрямую использует функции ОС для работы с файлами. Файловые системы обычно обеспечивают хранение слабо структурированной информации, оставляя дальнейшую структуризацию прикладным программам. В таких системах сложно решить проблемы согласования данных в разных файлах, коллективного доступа к данным, модификации структуры данных.

Банком данных называют систему специальным образом организованных баз данных, программных, технических, языковых и организационно - методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.



В отличие от файловых систем, структура базы данных меньше зависит от прикладных программ, а все функции по работе с БД сосредоточены в специальном компоненте - системе управления базами данных (СУБД), которая играет центральную роль в функционировании банка данных, так как обеспечивает связь прикладных программ и пользователей, данными. Сведения о структуре БД сосредоточены в словаре-справочнике (репозитории). Этот вид информации называется метainформацией. В состав метainформации входит семантическая информация, физические характеристики данных и информация об их использовании. С помощью словарей данных автоматизируется процесс использования метainформации в ИС.

Интеллектуальный банк данных ИБД (рис. 1.6) - это сравнительно новый способ построения ИС, при котором информация о предметной области условно делится между двумя базами.



База данных содержит сведения о количественных и качественных характеристиках конкретных объектов. База знаний содержит сведения о закономерностях в ПО, позволяющие выводить новые факты из имеющихся в БД; мета информацию; сведения о структуре предметной области; сведения, обеспечивающие понимание запроса и синтез ответа.

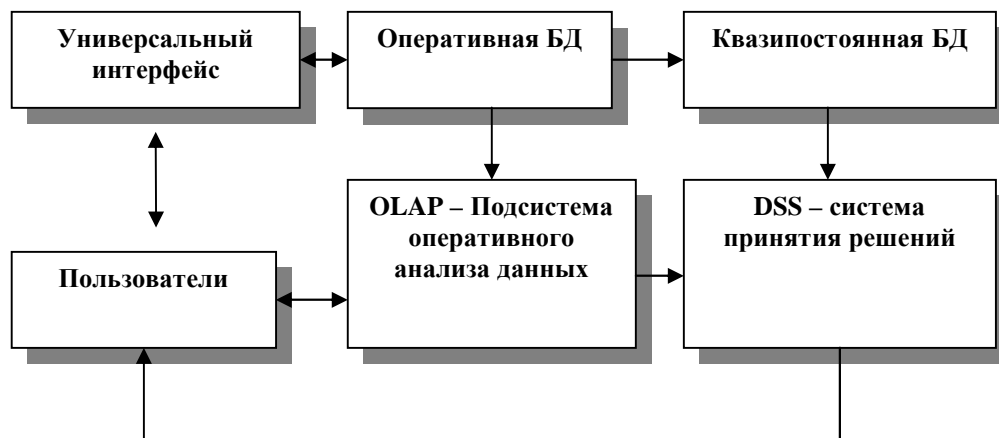
Диалоговый процессор предназначен для понимания смысла запроса и его перевода в термины знаний, заложенных в БЗ. Планировщик преобразует полученный запрос в рабочую программу, составляя ее из модулей, имеющихся в БЗ. Подсистема пополнения знаний позволяет ИС обучаться.

Если в традиционном банке данных знания о предметной области заложены программистом в каждую прикладную программу, а также в структуру БД, то в интеллектуальном банке данных они хранятся в базе знаний и отделены от прикладных программ. В отличие от данных, знания активны: на их основе формируются цели и выбираются способы их достижения. Например, ИБД в системе складского учета может автоматически реагировать на такое событие, как уменьшение количества деталей на складе до критической нормы, при этом ИБД без участия пользователя генерирует документы для заказа этих деталей и отправляет их по электронной почте поставщику.

Другое характерное отличие знаний от данных - связность, причем знания отражают как структурные взаимосвязи между объектами предметной области, так и вызванные конкретными бизнес - процессами, например такие связи, как "происходит одновременно", "следует из...", "если - то" и др.

Наконец, существенную роль в ИБД играет форма представления информации для пользователя: она должна быть как можно ближе к естественным для человека способам обмена данными (профессиональный естественный язык, речевой ввод / вывод, графическая форма).

В настоящее время в корпоративных базах данных накоплены гигантские объемы информации, однако она недостаточно эффективно используется в процессе управления бизнесом, поэтому бурно развивается новая форма построения ИС - склады (хранилища) данных.



Хранилище данных представляет собой автономный банк данных, в котором база данных разделена на два компонента: оперативная БД хранит текущую информацию, квазипостоянная БД содержит исторические данные, например, в оперативной БД могут содержаться данные о продажах за текущий год, а в квазипостоянной БД хранятся систематизированные годовые отчеты и балансы за все время существования предприятия. Подсистема оперативного анализа данных позволяет эффективно и быстро анализировать текущую информацию. Подсистема принятия решений пользуется обобщенной и исторической информацией, применяет методы логического вывода. Для общения с пользователем служит универсальный интерфейс.

Выбор того или иного класса ИС зависит от ее назначения и конкретных условий применения.

Структура программного обеспечения ИС

ПО информационной системы состоит из нескольких взаимосвязанных частей, каждая из которых определяет в системе заданные свойства. Анализ этих свойств позволяет выделить следующие блоки:

Обеспечение управления

— **Управляющий блок.** Содержит элементы, обеспечивающие управление системой и технологию работы отдельных приложений. Управляющие приложения ориентированы на предоставление информации для обработки или настройке различных параметров ИС. Кроме того, они включают в себя, приложения предоставляющие информацию для принятия решений, либо как элемент управления последовательностью выполнения технологического процесса. Управляющий блок состоит из нескольких частей:

- *Административная часть* - содержит элементы, ориентированные на управление системой, в которой принятие решений лежит в области действий пользователя. Например, управление допуском к информации в приложениях.

- *Управляющая часть* - содержит элементы, ориентированные на автоматическое управление технологическим циклом информационной системы, где принимаются решения по работе ИС без участия пользователей.

Обеспечение бизнес-процессов

- **Конвейерный блок.** Содержит элементы, составляющие группу решений обеспечивающих производственный цикл работы информационной системы. Конвейерные приложения выполняют сбор и обработку информации по заданным правилам и в определенной последовательности.
- **Учетный блок.** Содержит элементы, обеспечивающие учетные функции системы, с регистрацией происходящих изменений. Учетные приложения ориентированы на ввод, хранение и предоставление информации необходимой для выполнения технологического цикла.

Структура приложений в ИС

Пользовательские приложения могут быть классифицированы не только по их функциональной принадлежности к той или иной группе, но и по формам организации отношений между ними. Форма организации взаимодействия пользовательских приложений в единую систему может иметь следующий вид:

- Технологическая организация;
- Временная организация;
- Административная организация.

Технологическая организация взаимодействия приложений в системе основана на задаваемой управляющим блоком ИС, последовательностью работы приложений. Эта форма наиболее приемлема для построения отношений в клиент/серверной технологии. Ею достигается, необходимая “технологическая” последовательность в работе приложений. Порядок выполнения решений основан на механическом контроле работы различных частей системы.

Временная организация взаимодействия приложений в системе строится на работе элементов в строго заданные интервалы времени. Приложение может быть активизировано в четко определенное или заданное время. Порядок выполнения решений основан на управлении последовательностью выполнения приложений во временных интервалах.

Административная организация приложений основана на информационном взаимодействии, в котором одно из приложений является управляющим, а другие подчиненными. В такой структуре правила работы подчиненных приложений зависят от работы управляющего приложения. Порядок выполнения решений основан на постоянном взаимодействии подчиненных и управляющего приложения.

Структура хранения информации в ИС

При проектировании структуры хранения информации следует учитывать не только требования, предъявляемые программно-аппаратной платформой, но и “жизненный цикл” информации хранимой в БД.

“Жизненный цикл информации” отражает то, что все данные введенные в базу данных имеют целесообразный временной интервал их использования. Проектирование без учета этой особенности приводит, к излишнему хранению информации в оперативных элементах базы данных. Это может увеличивать время реакции системы на запросы и усложнить обслуживание базы данных производственными приложениями. Проектирование с учетом “жизненного цикла информации” позволяет повысить устойчивость БД, и на внести ограничения на объем хранимых данных.

Анализ различных информационных систем позволяет выделить несколько групп, по которым возможно классифицировать таблицы базы данных:

- Технологическая группа;
- Обеспечивающая группа;

— Архивная группа.

К *“технологической группе”* относятся таблицы, в которых храниться информация обеспечивающая полный цикл выполнения “бизнес-процессов”. Она актуальна только на момент выполнения процессов. Например, набор данных, который формируется и используется в определенный временной интервал. Как правило, в ИС подобный тип данных является наиболее объемным. Для построения оптимальных моделей хранения и предоставления данных, с тем, что бы иметь возможность реконструировать состояние системы на определенный момент времени, следует организовать перевод используемой информации в “архивную группу”, или уничтожать накопившиеся данные. Наличие в активных элементах БД неиспользуемой информации приводит к увеличению времени реакции системы на запросы и усложняет организацию по обслуживанию базы данных.

К *“обеспечивающей группе”* относятся таблицы, информация в которых не привязана к отдельным бизнес процессам. Например, различные справочники, классификаторы и т.п. “Жизненный цикл” этой информации можно считать бесконечным.

К *“архивной группе”* относятся таблицы, в которых использование информации носит эпизодический характер. Например, в качестве статистического материала, или как место хранения большого объема информации не участвующего в оперативной работе системы.

Единое информационное пространство

Взаимодействие базовых блоков системы основано на использовании данных из “единого информационного пространства”. **“Единое информационное пространство”** – это специальным образом организованное хранилище данных. В рамках хранилища каждое приложение может на основе уже существующей общедоступной информации создавать новый тип данных, также доступный всем элементам системы. Однако “единое информационное пространство” - это не только хранилище, но и единый механизм управления доступом, позволяющий предоставлять разным группам пользователей различный объем данных. “Единое информационное пространство” - это комплекс административных и системных мероприятий, обеспечивающих выполнение следующих правил:

- Один источник данных – имеет много потребителей;
- Каждый потребитель - собственный способ предоставления информации;

Для разных групп потребителей доступен разный объем данных. “Единое информационное пространство” позволяет вводить элементы “самоконтроля”, поддерживающие “целостность” имеющихся в системе данных.

При проектировании “единого информационного пространства” следует учитывать не только функциональность структуры базы данных, но и возможность дальнейшего развития информационной системы, модификации пользовательских приложений, оперативность проектирования. При этом проектировка структуры базы данных должна вестись с учетом “жизненного цикла” информации, “возможного” развития самой структуры, т.е. предусматривать развитие функциональности системы. Кроме того, информационная система должна иметь “развивающую основу” в эксплуатируемой модели.

Для построения “развивающейся” информационной системы необходимо выполнение следующих условий:

- Структура системы должна быть построена на “блочной” основе, где один и тот же блок может быть использован в нескольких приложениях;
- Механизм проектирования системы должен позволять производить модификацию и наращивание информационных блоков без внесения деструктивного влияния на действующую систему.

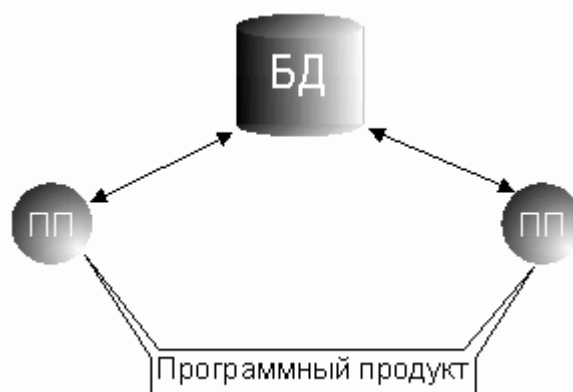
Классификация программных решений

Рассмотрим в рамках автоматизации обработки информации классификацию программных решений. Основой классификации служит объем конечных функций, реализованных в программе, и метод их объединения. Существуют три группы моделей:

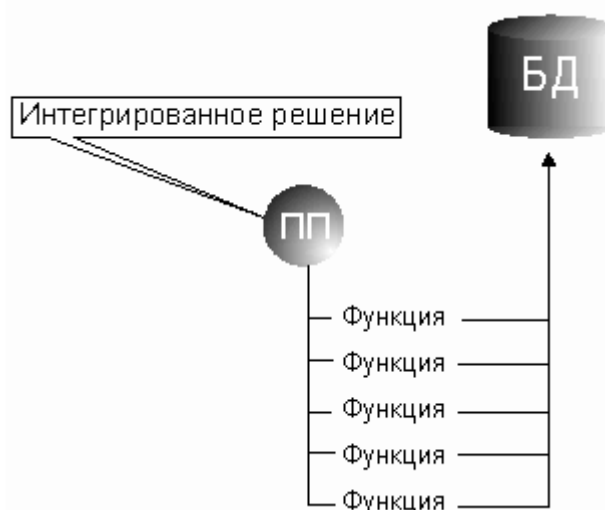
- Модель, состоящая из одного компонента и отражающая одну функцию;
- Модель, состоящая из одного компонента и отражающая несколько функций;

- Модель, состоящая из нескольких компонентов объединенных в единую систему.
- Существуют следующие типы организации информационных моделей:
- Программный продукт (ПП)
- Интегрированное решение (ИР)
- Информационная система (ИС)

“Программный продукт (ПП)” – модель, в которой решение поставленной задачи организовано в рамках единого компонента, ориентированного на выполнение единственной функции. Не смотря на то, что набор отдельных ПП, может решать и сложные задачи, отсутствие механизмов по их организации позволяет отнести такую информационную модель к простым программным решениям.

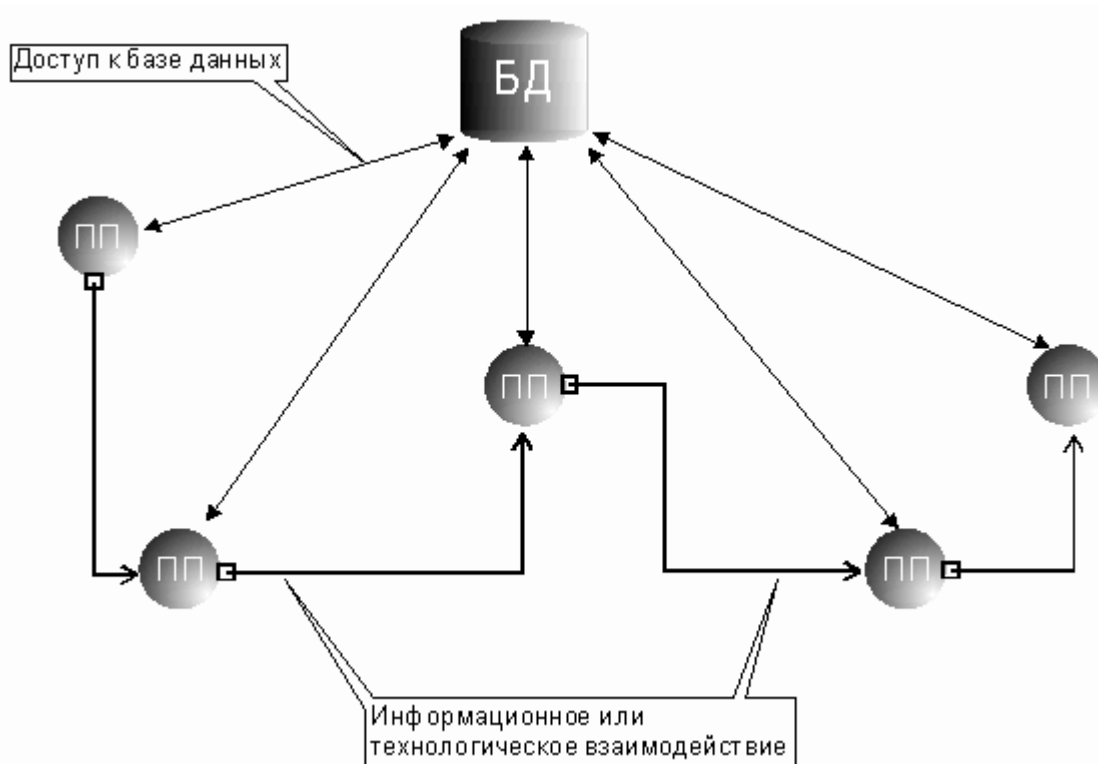


“Интегрированное решение (ИР)” – модель, в которой решение поставленной задачи организовано в рамках одного компонента, ориентированного на выполнение нескольких функций. В моделях данного типа все функции объединены под общей оболочкой осуществляющей управление их работой. Отсутствие механизмов по организации работы функций и их отношений, позволяет отнести данную информационную модель к систематизированным программным решениям.



“Информационная система (ИС)” – модель, в которой решение поставленной задачи организовано из нескольких компонентов, использующих единый информационный источник, и объединенных в технологическую цепочку. Правила работы данного объединения могут быть выражены как в логике работы отдельных приложений, так и их группы. В рамках ИС могут быть использованы приложения, функционирующие на разных программных платформах, и ориентированные на решения узкоспециализированных задач. При этом возможно применение ПП и ИР, если есть доступ к их информационным ресурсам или функциям. Наличие механизмов организации взаимодействия компонентов, в процессе выполнения, и их влияние на правила работы друг друга, позволяет отнести такую модель к *сложным системам*.

Данная классификация определяет и систематизирует отличия в методах организации и способов решения задач. Она не влияет на особенности процесса проектирования и не может быть использована как критерий оценки способов реализации. Эта классификация позволяет принимать решения по организационным методам построения программных приложений. Она потенциально дает представление об административных и производственных возможностях рассматриваемых решений.



Этапы разработки автоматизированных информационных систем

Основные этапы проектирования АИС представлены в следующей таблице.

№	Наименование этапа	Основные характеристики
1	Разработка и анализ бизнес - модели	<p>Определяются основные задачи АИС, проводится декомпозиция задач по модулям и определяются функции с помощью которых решаются эти задачи. Описание функций осуществляется на языке производственных требований (описание процессов предметной области), функциональных требований (описание форм обрабатываемых документов) и технических требований (аппаратное, программное, лингвистическое обеспечение АИС).</p> <p>Метод решения: Функциональное моделирование.</p> <p>Результат:</p> <p>1. Концептуальная модель АИС, состоящая из описания предметной области, ресурсов и потоков данных, перечень требований и ограничений к технической реализации АИС.</p> <p>2. Аппаратно-технический состав создаваемой АИС.</p>
2	Формализация бизнес - модели, разработка	<p>Разработанная концептуальная модель формализуется, т.е. преобразуется логическую модель АИС.</p>

	логической модели бизнес процессов.	<p>Метод решения: Разработка диаграммы "сущность-связь" (ER (Entity-Relationship) - CASE- диаграммы).</p> <p>Результат: Разработанное информационное обеспечение АИС: схемы и структуры данных для всех уровней модульности АИС, документация по логической структуре АИС, сгенерированные скрипты для создания объектов БД.</p>
3	Выбор лингвистического обеспечения, разработка программного обеспечения АИС.	<p>Выбирается лингвистическое обеспечение (среда разработки - инструментарий), проводится разработка программного и методического обеспечения. Разработанная на втором этапе логическая схема реализуется в виде программных объектов, при этом логические схемы реализуются в виде объектов базы данных, а функциональные схемы - в виде пользовательских форм и приложений.</p> <p>Метод решения: Разработка программного кода с использованием выбранного инструментария.</p> <p>Результат: Работоспособная АИС.</p>
4	Тестирование и отладка АИС	<p>Корректировка информационного, аппаратного, программного обеспечения, разработка методического обеспечения (документации разработчика, пользователя) и т.п.</p> <p>Результат: Оптимальный состав и эффективное функционирование АИС.</p> <p>Комплект документации: разработчика, администратора, пользователя.</p>
5	Эксплуатация и контроль версий	<p>Особенностью АИС созданных по архитектуре клиент сервер является их многоуровневость и многомодульность, поэтому при эксплуатации и развитии АИС на первое место выходят вопросы контроля версий, т.е. добавление новых и развитие имеющихся модулей с выводом из эксплуатации старых. Если ежедневный контроль версий не ведется, то БД АИС за год эксплуатации может насчитывать более 1000 таблиц, из которых эффективно используются лишь 20-30%.</p> <p>Результат: Нарастиваемость и безизбыточный состав гибкой, масштабируемой АИС</p>

Разработка и анализ бизнес-модели

При построении эффективной автоматизированной информационной системы первым этапом является исследование и формализация бизнес-процессов деятельности предприятия. Т.е. описание системы ведения делопроизводства с целью эффективного использования информации для достижения поставленных задач и решения проблем, стоящих перед организацией. Организация работы с документами является важной составной частью процессов управления и принятия управленческих решений, существенно влияющей на оперативность и качество управления. Процесс принятия управленческого решения состоит из:

- Получения информации;
- Переработки информации;
- Анализа, подготовки и принятия решения.

Все эти этапы связаны с документным обеспечением процессов управления, проектирования и производства. Если на предприятии отсутствует четкая организация работы с документами, то формируются документы низкого качества, как в оформлении, так и в полноте и ценности содержащейся в них информации и увеличиваются сроки их обработки. Это приводит к ухудшению качества управления, увеличению сроков принятия решений и увеличению числа неверных решений. С ростом масштабов предприятия и численности его сотрудников вопрос об эффективности документного обеспечения управления становится наиболее актуальным. При этом возникают следующие проблемы:

- руководство теряет целостную картину происходящего;
- структурные подразделения, не имея информации о деятельности друг друга, не могут согласованно осуществлять свою деятельность. Падает качество обслуживания клиентов и способность организации поддерживать внешние контакты;
- происходит падение производительности, что вызывает ощущение недостатка в людских, технических, коммуникационных и других ресурсах;
- расширяется штат, вкладываются деньги в оборудование новых рабочих мест, помещений, коммуникации, обучение новых сотрудников;
- для производственных предприятий увеличение штата может привести к изменению технологии производства, что потребует дополнительных инвестиций;
- штат увеличивается, производительность падает, производство требует инвестиций, возникает потребность в увеличении оборотного капитала. Это может потребовать новых кредитов и уменьшить плановую прибыль.

В итоге предприятие замедляет рост и развитие происходит экстенсивным путем за счет ранее созданной прибыли.

Основные понятия электронного документооборота

Документ – это слабоструктурированная совокупность блоков или объектов информации, понятная человеку.

Он представляет собой совокупность трех составляющих:

- Физическая регистрация информации.
- Форма представления информации
- Активизация определенной деятельности.

Документооборот может иметь один из двух типов:

- универсальный - автоматизирующий существующие информационные потоки слабоструктурированной информации;
- операционный - ориентированный на работу с документами, содержащими операционную атрибутику, вместе с которой ведется слабоструктурированная информация.

Кроме собственно документов важен еще регламент работы с ними.

Электронный документооборот

К основным преимуществам электронного документооборота можно отнести следующие:

- Полный контроль за перемещением и эволюцией документа, регламентация доступа и способов работы пользователей с различными документами и их отдельными частями.
- Уменьшение расходов на управление за счет высвобождения (на 90% и более) человеческих ресурсов, занятых различными видами обработки бумажных документов, уменьшение задержек обработки за счет маршрутизированного перемещения документов и жесткого контроля за порядком и сроками их прохождения.
- Быстрое создание новых документов из уже существующих.

- Поддержка одновременной работы многих пользователей с одним и тем же документом, предотвращение его потери или порчи.
- Сокращение времени поиска нужных документов.

Использование АИС может рассматриваться в качестве базы для общего совершенствования управления предприятием. При этом управление предприятием реализует следующие основные функции:

- обслуживание клиентов;
- разработка продукции;
- учет и контроль за деятельностью предприятия;
- финансовое обеспечение деятельности предприятия.

Модели информационного пространства предприятия

Комплексная автоматизация перечисленных функций требует создания единого информационного пространства предприятия, в котором сотрудники и руководство могут осуществлять свою деятельность, руководствуясь едиными правилами представления и обработки информации в документном и бездокументном виде.

Для этого в рамках предприятия требуется создать единую информационную систему по управлению информацией или единую систему управления документами, включающую возможности:

- удаленная работа, при которой сотрудники могут работать в разных помещениях или в разных зданиях;
- доступ к информации, при котором разные пользователи имеют доступ к одним и тем же данным без потерь в производительности независимо от своего местоположения в сети;
- средства коммуникации (электронная почта, факс, печать документов);
- сохранение целостности данных в общей базе данных;
- полнотекстовый и реквизитный поиск информации;
- открытость системы, при которой пользователи имеют доступ к привычным средствам создания документов и к уже существующим документам, созданным в других системах;
- защищенность информации;
- удобство настройки на конкретные задачи пользователей;
- масштабируемость системы для поддержки роста организации, защиты вложенных инвестиций.

Начальным этапом создания такой системы является построение модели предметной области или другими словами модели документооборота для конкретного предприятия.

Основными направлениями автоматизации документооборота являются: поддержка фактографической информации, возможность работы с полнотекстовыми документами, поддержка регламента прохождения документов. Они определяют трехмерное пространство свойств, в котором любой программный продукт данного класса движется по некоторой траектории, проходя различные стадии в своем развитии.



Ось (F) характеризует уровень организации хранения фактографической информации, которая привязана к специфике конкретного рода деятельности компании или организации. Например: при закупке материальных ценностей происходит оформление товарно-сопроводительных документов (накладных, приемо-передаточных актов, приходных складских ордеров и т.п.), регистрируемых в качестве операционных документов. Их атрибутика очень важна для принятия управленческих решений. Информация из операционных документов используется при сложной аналитической и синтетической обработке, и, может быть получена пользователем через систему отчетов.

Ось (D) - полнотекстовые документы. Она отражает необходимость организации взаимодействия при формировании и передачи товаров, услуг или информации как внутри корпорации, так и вне ее. В этих документах наряду с фактографической информацией содержится слабо структурированная информация, не подлежащая автоматизированной аналитической обработке, такая, как форс-мажорные факторы и порядок предъявления претензий при нарушении условий договора. Все взаимоотношения между субъектами бизнеса сопровождаются документами, которые являются отражением результата взаимодействия.

Ось (R) отражает регламент процессов прохождения документов, а именно: описание того какие процедуры, когда и как должны выполняться. Основу для позиционирования относительно данной оси представляет набор формальных признаков (атрибутов) и перечень выполнения операций.

Точка в пространстве (F, D, R) определяет состояние системы документооборота и имеет координаты (f,d,r), где f,d и r принадлежат множествам F,D и R, соответственно. Положение этой точки зависит от уровня развития и стадии внедрения системы документооборота на предприятии, а также от его специфики и самих масштабов бизнеса.

Представив модель документооборота таким образом, можно, зная текущее положение дел с организацией делопроизводства на предприятии, четко представить, в каких направлениях следует двигаться, чего недостает в текущий момент и каким образом использовать уже существующие системы автоматизации.

В общем случае, процесс автоматизации делопроизводства на предприятии можно представить в виде кривой в трехмерном пространстве координат F,D,R.

Эволюция модели документооборота

Рассмотренная модель прошла 3 фазы развития.

Первая фаза - фактографическая. Она характеризуется периодом накопления информации, имеющей жесткую структуру и атрибутику. Условно уровень организации этой информации откладывается вдоль одной оси.

Точка данной оси определяет текущее состояние системы документооборота. Движение по оси вверх характеризует накопление фактографической информации. Начиная с определенного момента данного этапа можно отметить второй этап первой фазы - возникновение понятия "операция". Документ представляется как некоторый привязанный к бизнес - процессам предприятия агрегат из имеющихся характеристик (атрибутов). Движение по оси приобретает более операционный характер. Привязка к конкретным бизнес - процессам делает дальнейшее развитие документооборота в одномерном пространстве невозможным - необходим качественный переход к новой фазе.

Вторая фаза - полнотекстовая. Расширение организации и увеличение круга решаемых задач требуют использования полнотекстовых документов, включающих не только тексты, но и другие способы представления информации: графики, таблицы, видео и различные виды конструкторско-технологической документации. Появляется новая ось - полнотекстовые или, мультимедийные документы. Точка в новом, двумерном, пространстве характеризует систему документооборота предприятия, где кроме фактографической базы документов имеются хранилища и архивы информации.

Хранилища позволяют накапливать документы в различных форматах, предполагают наличие их структуризации и возможностей поиска. Если на предприятии уже используется автоматизация, то хранилище - это электронный архив. Движение по оси "полнотекстовые документы" предполагает наращивание атрибутивных возможностей: разграничение доступа, расширение средств поиска, иерархию хранения, классификацию. Возникают такие понятия как электронная подпись, шифрование и т.п.



На данной оси имеются свои этапы - с определенного момента развития хранилища можно говорить не об индивидуальном, а о корпоративном архиве, обслуживающем деятельность рабочих групп. Точка на плоскости FD, достигнутой во второй фазе, характеризует систему документооборота, позволяющую отображать фактографическую информацию в виде полнотекстовых документов, имеющих необходимое количество атрибутов. Доступ к этим документам может быть осуществлен по маршруту любого уровня сложности с соблюдением различных уровней конфиденциальности. Если, например, говорить о точке "А", то соответствующее ей состояние системы документооборота позволяет осуществлять синхронизацию работы различных рабочих групп сотрудников корпорации. Система для этой точки предполагает структурирование информации по уровням управления и наличие средств репликации данных.

Третья фаза - регламентирующая. Нормальный документооборот в масштабах корпорации невозможен без решения вопросов согласования или соблюдения регламента работы. Если ранее, на второй фазе присутствовал лишь один, простейший регламент (нулевая точка) - сотрудник имел доступ к архиву или его части, либо в папку сотрудника помещалось индивидуальное задание, то теперь необходим, контроль за тем, как работник выполнил задание, или как продвигается документ в условиях нелинейного процесса его согласования (например согласования пакета конструкторско-технологической документации на сборочную единицу).

Третья ось в пространстве документооборота предприятия, как и две другие имеет свое деление на этапы. Первоначальный этап движения по оси характеризуется наличием упрощенного регламента, отображаемого появлением атрибутов, отвечающих за регламент, например: "оплатить до", "действителен для". Количественное накопление атрибутов и расширение возможностей по управлению регламента сопровождается постепенным переходом ко второму этапу, отличительная черта которого - появление системы, специально

предназначенной для отслеживания процесса соблюдения регламента. При дальнейшем движении вдоль этой оси можно говорить о появлении единой системы управления проектом. Теперь документ в системе "документооборота" становится вторичным. Первичной является цель бизнеса и процесс реализации бизнес - процедур, оставляющий после себя документы. Т.о. от модели потока документов приходим к модели потока работ (work flow).

Оси "F" и "D" определяют специфику деятельности организации, регламентируемую положением третьей координаты (R) пространства модели документооборота. При этом модель не зависит от технологии обработки документов, принятой на предприятии - все решает только цель деятельности, будь то государственная организация, торговая компания и промышленная фирма. В общем случае можно выделить три типа организаций:

- Банк и торговая компания (приобретение, наценка, продажа, получение прибыли - главный объект деятельности);
- бюджетная организация: (основная деятельность - формирование документов);
- промышленное предприятие: (закупка сырья, переработка, создание нового продукта, реализация, получение прибыли. Цель деятельности – операция).

Если задачей организации является формирование документов, например мэрия, суд или министерство, то ее позиция в модели будет занимать достаточно высокое положение относительно осей "F" и "D". Однако если рассматривать деятельность коммерческого банка или фирмы задача которой - производство операций, материальных ценностей, то здесь все три координаты должны иметь сбалансированные значения.

Глава 3

Теоретические основы информационных систем

Знаковые системы

В самом общем виде знания в памяти компьютера представляются в виде некоторой знаковой системы. Знаковые системы изучает наука семиотика. Основным понятием семиотики является понятие знак.

Знак - это объект или событие, которые способны что-то обозначать, т.е. указывать на некоторый другой объект, и что-то означать, т.е. иметь некоторый смысл. Например, «Луна» обозначает конкретный физический объект и имеет смысл «естественный спутник Земли»; знак «*P*» обозначает число 3.14, имеет смысл «отношение длины окружности к диаметру».

Не все объекты, сопоставленные другим объектам, выступают в качестве знаков. Например, паспорт, квитанции или гардеробные номерки не являются знаками.

С понятием знак непосредственно связаны понятия денотат и концепт знака.

Денотат - это объект, обозначаемый данным знаком, а концепт - свойство денотата, выражаемое знаком. Концепт определяет свой денотат. Отношение между знаком, его концептом и денотатом выражает так называемый семиотический треугольник, или треугольник Фреге (рис. 1)

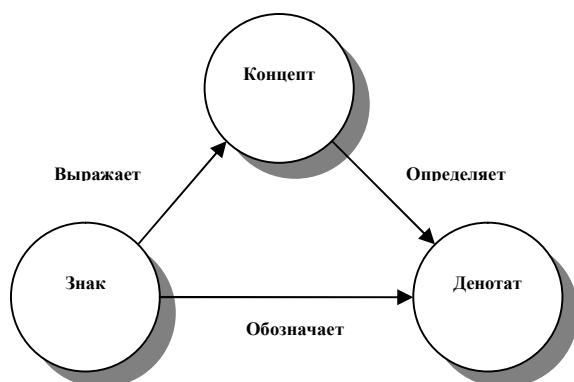


Рисунок 1. Семиотический треугольник

Когда знак реально вступает во взаимодействие со своим денотатом и концептом, возникает знаковая ситуация. Возможны знаковые ситуации, в которых одна из вершин семиотического треугольника отсутствует (рис. 2)

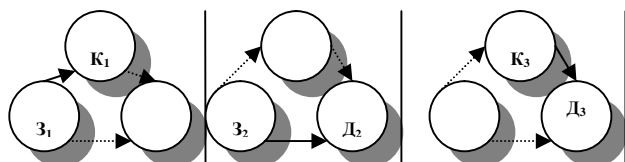


Рисунок 2. Знаковая ситуация

Поскольку, по мнению Г.Фреге, знак обозначает денотат через посредство своего концепта, изобразим семиотический треугольник вытянутым в цепь (З, К, Д) (рис. 3).

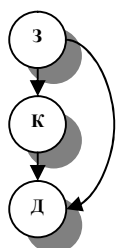


Рисунок 3. Вытянутый семиотический треугольник

Соответствия между знаками, концептами и денотатами не являются взаимно-однозначными. Отметим основные особенности употребления знаков (рис. 4).

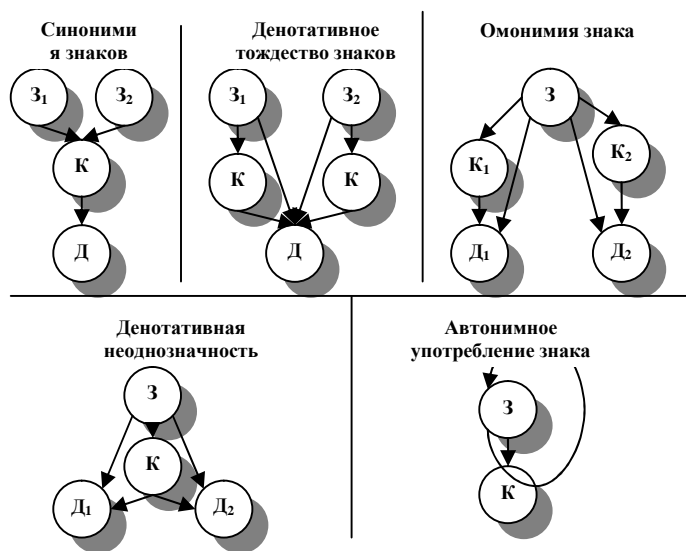


Рисунок 4. Неоднозначность элементов семиотического треугольника

Знаки Z_1, Z_2 называются синонимичными, если они выражают один и тот же концепт. Знаки Z_1, Z_2 , каждый со своим концептом, называются денотативно тождественными, если они обозначают один и тот же денотат.

Знак Z называется омонимичным (или многозначным), если он может выражать более чем один концепт. Знак Z называется денотативно неоднозначным, если он при одном и том же концепте может обозначать различные денотаты.

Знак Z автонимен, если его денотатом в данной знаковой ситуации выступает сам знак Z .

Из семиотического треугольника следует, что знак имеет две знаковые функции: обозначает не только денотат, но и его концепт - десигнат данного знака.

Как следует из рис. 4, один и тот же знак может обозначать разные денотаты. Объем знака - это объем поля денотатов. Важными понятиями, имеющими к треугольнику Фреге непосредственное отношение, являются понятия экстенсионал и интенсионал.

Экстенционал знака определяет конкретный класс всех его допустимых денотатов. Если имя есть предложение, высказывание (т.е. запись некоторого суждения), то его денотатом служит истинное значение этого предложения (высказывания, суждения), т.е. «истина» или «ложь».

Интенционал знака определяет содержание понятия и характеризует концепт. Интенциональность знака зависит не только от истинных его значений, но и от прагматических оттенков этого смысла.

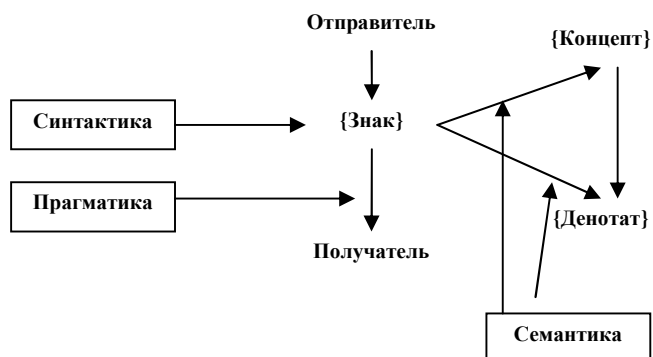


Рисунок 5. Знаковая система

Знаковая система (рис. 5) - это множество знаков с регулярными отношениями между ними, отражающими регулярные отношения между их концептами и денотатами. В знаковой системе выделяют три аспекта: синтактику, семантику и прагматику.

Синтактика изучает внутреннее устройство знаковой системы, правила построения сложных знаков из простых. Для естественных языков синтактика выступает в виде синтаксиса, определяющего правильное построение предложения и связного текста. В искусственных языках синтактика определяет правильное логическое построение потенциально осмысленных выражений.

Семантика изучает соотношения, с одной стороны, между знаками и их денотатами, с другой - между знаками и их концептами (смыслами).

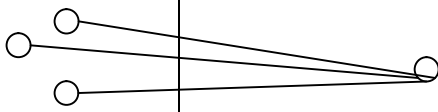
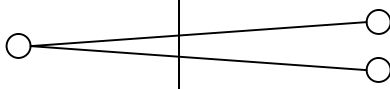
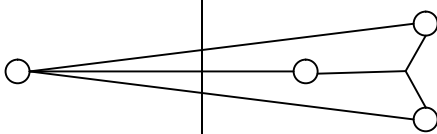
Прагматика изучает знаки с точки зрения их отношения к адресату сообщений.

КЛАССИФИКАЦИОННЫЕ СИСТЕМЫ

Классификационные системы с давних пор применяются для структурирования и обобщения знаний. В таких системах, с одной стороны, все сущности разбиваются по определенным признакам на некоторое число классов, с другой стороны, группируются вместе. При классифицировании наблюдателю дается набор объектов, которые можно описать некоторым множеством признаков. Каждый объект принадлежит одному или более классам из некоторого фиксированного множества. В задаче классификации (образов) наблюдатель должен применить установленное ранее правило, чтобы решить, к какому классу принадлежит объект. В задаче распознавания (образов) правило классификации вырабатывается на основе исследований множества объектов с известной принадлежностью различным классам. Эти объекты в совокупности называются обучающим множеством, или выборкой. В задаче формирования образов объекты предъявляются наблюдателю без указания их принадлежности классам. Наблюдатель должен самостоятельно построить соответствующее определение классов. Задача классификации эквивалентна задаче выяснения: является ли некоторая цепочка предложением в формальном языке.

Некоторые классификационные системы широко применяются при представлении декларативных знаний.

Вся совокупность употребляемых при классификации слов называется лексикой. Для обеспечения лексической однозначности должны быть учтены отношения синонимии, омонимии и полисемии (или многозначности), свойственные словам естественного языка. Отношения синонимов, омонимов и многозначных слов к обозначенным ими предметам или выражаемым ими понятиям показаны в таблице.

Отношения		
Тип отношения	Слова (имена)	Предметы (понятия)
Синонимия		
Омонимия		
Полисемия		

Например:

Синонимы: сумка - саквояж, размытые множества - расплывчатые множества;

Омонимы: мосты - мосты (строения), мосты (зубные), мосты (ходовая часть), электрические мосты;

Полисемия: отражение (нападения), отражение (света), отражение (звука), отражение (электромагнитных волн).

Между словами естественных языков существуют два вида отношений.

1. Парадигматические (базисные, имманентные, аналитические)- отношения, обусловленные наличием логических связей между предметами и явлениями, обозначаемые этими словами. Такие отношения носят вне-языковой характер и не зависят от ситуации, для описания которой используются слова. Например, СТОЛ и СТУЛ - МЕБЕЛЬ (соподчинение, координация); СТОЛ и МЕБЕЛЬ - отношение вид - род; ПЕРЕПЛЕТ и КНИГА - отношение часть - целое; ЛАМПА и СВЕТ - отношение причина - следствие; ЛОПАТА и ЭКСКАВАТОР - отношение функционального сходства.

2. Синтагматические (текстуальные, ситуативные) - линейные отношения, которые устанавливаются непосредственно при соединении слов в словосочетания и фразы. Синтагматические средства составляют синтаксис языка и относятся к его грамматическим средствам. Классом называется совокупность (множество) предметов, каждому, из которых присущи признаки, отражаемые в содержании соответствующего понятия. Слово или словосочетание, выражающее это понятие, служит именем данного класса.

Классификацией называется система распределения предметов или отношений на основании наиболее существенных признаков, присущих этим предметам или отношениям и отличающих их от других предметов или отношений. Классы могут быть простыми и сложными. Простым называется такой класс, члены которого характеризуются только одним общим признаком, выражаемым или обозначаемым именем этого класса. Такое имя обычно выражается одним словом или одним устойчивым словосочетанием. Например, самолеты; пассажиры; реактивные двигатели и т. д. Как правило, простой класс нельзя расчленить на более простые, не теряя при этом возможности однозначно восстановить исходный класс путем логического умножения полученных более узких классов.

Сложным называется такой класс, члены которого имеют не один общий признак, а сочетание признаков. Именами сложных классов являются различные словосочетания и даже целые фразы. Например, реактивные пассажирские самолеты; вычислительные машины на интегральных схемах. Сложные классы всегда можно расчленить на простые, без потери возможности их однозначного восстановления путем логического умножения полученных простых классов.

Иерархические системы классификации. Иерархическая система классификации - это такая система, в которой между классификационными группировками установлено отношение

подчинения, как правило, родовидовое. Классификационное множество объектов делится по некоторому выбранному признаку (основание деления) на крупные группировки, затем каждая группировка в соответствии с выбранным основанием деления разбивается на ряд последующих группировок, которые в свою очередь распадаются на более мелкие, постепенно конкретизируя объект классификации (рис. 2.8).

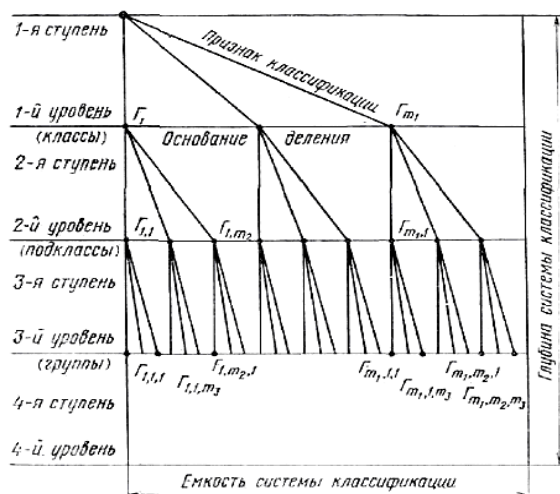


Рисунок 6. Классификационная система

При построении иерархической системы классификации необходимо соблюдать следующие наиболее важные формально-логические правила:

- каждая классификационная группировка должна делиться только по одному основанию деления;
- получаемые в результате деления группировки должны исключать друг друга (т.е. не повторяться);
- сумма подмножеств деления должна составлять делимое множество.

Основными преимуществами иерархической системы классификации являются большая информационная емкость и простота поиска (возможность ручного поиска). Недостатки заключаются в малой гибкости структуры и невозможности агрегировать объекты по любому произвольному сочетанию признаков.

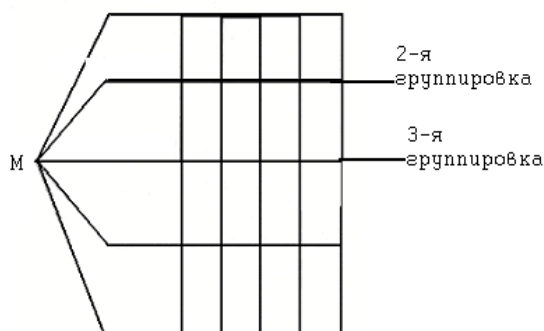


Рисунок 7. Фасетная классификация

Фасетная система классификаций. Это такая система (рис. 7), при которой классифицируемое множество образует независимые группировки по различным аспектам классификации. Классификационные группировки образуются путем комбинаций значений, взятых из соответствующих фасетов.

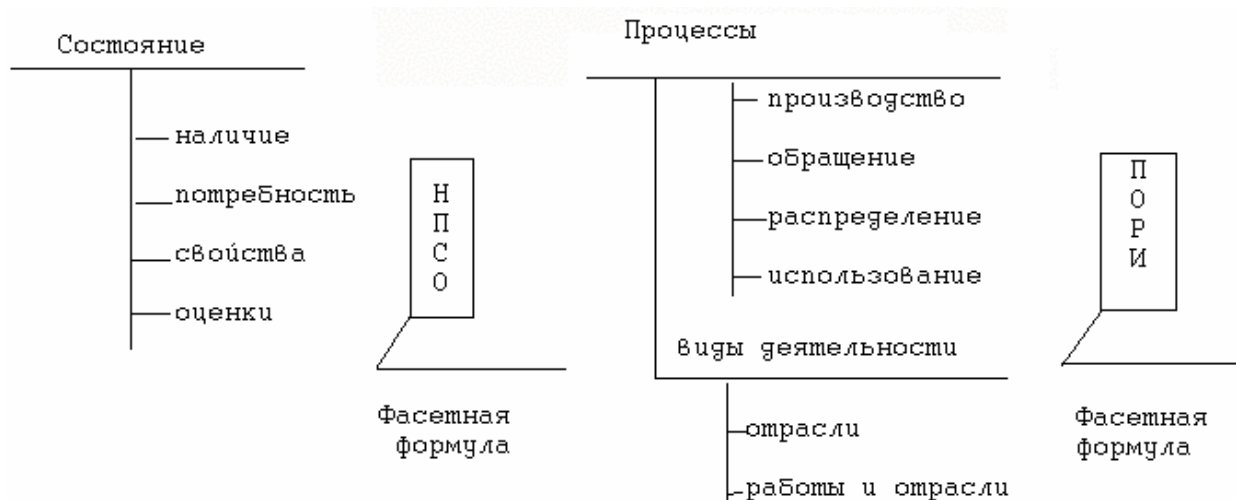


Рисунок 8. Фасетные формулы.

Последовательность расположения фасетов задается фасетной формулой, которая в каждом конкретном случае устанавливается в зависимости от характера решаемых задач и алгоритма обработки информации (рис. 8).

Порядок следования фасетов в фасетной формуле строго фиксировании определяется с учетом предмета, для которого разрабатывается фасетная классификация.

Преимущества фасетных классификаций заключаются в гибкости структуры, возможности агрегировать объекты по любому сочетанию фасетов, возможности блочного построения классификаций. Недостатки состоят в неполном использовании емкости классификации и сложности применения при ручной обработке.

Алфавитно-предметная классификация

Алфавитно-предметной классификацией называется система классов (каждый из которых соответствует одному виду предметов или факторов), расположенных в алфавитном порядке их имен, например, обычный телефонный справочник.

Тезаурус. Координатное индексирование заключается в том, что центральная тема документа или информационного запроса выражается в виде некоторого перечня (множества) слов и словосочетаний естественного языка, обычно являющихся именами простых классов. Такие слова и словосочетания рассматриваются как координаты документа в некотором n-мерном пространстве предметно-тематических признаков.

Для координатного индексирования документов или информационных запросов можно использовать полнозначные (неслужебные) слова, выбираемые непосредственно из индексируемых текстов. Такие слова и словосочетания называются ключевыми словами. Однако среди ключевых слов встречается немало синонимов, многозначных слов и омонимов (омографов). Кроме того, возможны различные написания одних и тех же ключевых слов. Из-за этого прямое (пословное) сопоставление поисковых образов документов с поисковыми предписаниями становится затруднительным, а информационный поиск - менее эффективным.

Для преодоления этих затруднений применяется лексикографический контроль за используемыми ключевыми словами. Этот контроль заключается:

- в приведении используемых ключевых слов к единой морфологической форме, к единому написанию;
- в учете синонимии, полисемии, омонимии ключевых слов.

Из одинаковых или близких по смыслу ключевых слов строится некоторый класс. Из членов этого класса выбирается наиболее представительное и стилистически нейтральное ключевое слово, которое назначается именем такого класса и становится дескриптором.

Нормативный словарь, в котором в едином алфавитном ряду приведены все важнейшие ключевые слова и дескрипторы по данной отрасли (эти ключевые слова и дескрипторы снабжены отсылочными, ограничительными и пояснительными пометками, устраняющими их синонимию, полисемию и омонимию), называется дескрипторным словарем.

Формулирование поисковых образов документов и поисковых предписаний, помимо

лексикографических сведений, требует предметно-тематических знаний. Дополнение дескрипторного словаря определенными сведениями о предметно-тематической области превращает его в нормативный словарь-справочник, называемый информационно-поисковым тезаурусом.

Информационно-поисковый тезаурус - это нормативный словарь, предназначенный для координатного индексирования документов и информационных запросов, в котором приведены в алфавитном порядке все дескрипторы и синонимичные им ключевые слова, а также отражены важнейшие парадигматические отношения между дескрипторами. В зависимости от выполняемой функции лексические единицы тезауруса делятся на дескрипторы и условные синонимы.

Дескриптор - однозначное ключевое слово или код, обозначающий (называющий) класс условной эквивалентности, в который включены эквивалентные слова и близкие по смыслу ключевые слова.

Элементарной структурной единицей тезауруса является словарная статья дескриптора, которая строится по алфавитно-структурному принципу

$$d_i < M_{i1}, M_{i2}, M_{i3}, M_{i4} >,$$

где d_i - заглавный дескриптор;

M_{i1} - упорядоченное по алфавиту множество условных синонимов данного заглавного дескриптора, образующих вместе с ним класс условной эквивалентности;

M_{i2} - упорядоченное по алфавиту множество дескрипторов, каждый из которых связан с заглавным дескриптором отношением род - вид;

M_{i3} - упорядоченное по алфавиту множество дескрипторов, каждый из которых связан с заглавным дескриптором отношением вид - род;

M_{i4} - упорядоченное по алфавиту множество дескрипторов, каждый из которых связан с заглавным дескриптором по крайней мере одним из следующих парадигматических отношений: целое - часть, часть - целое, причина - следствие, следствие - причина, функциональное сходство (ассоциативные связи).

Примеры словарных статей, построенных по указанному алфавитно-структурному принципу, имеют вид:

№ 1 - Трудящиеся - заглавный дескриптор

Син.

- Народные массы
- Народ

Вид.

- Интеллигенция
- Крестьянство
- Рабочий класс.
- Служащие

Асс.

- Производство
- Рабочее время

№ 2 - Рабочая сила - заглавный дескриптор

Син.

- Субъект труда

Род.

- Производительные силы
- Труд

Асс.

- Предмет труда

— Средства производства

Здесь: Син. - синонимичный; Вид - видовой; Род. - родовой; Асс. - ассоциативный термин.

Любое из перечисленных множеств может быть одноэлементным и даже пустым, т.е. может отсутствовать в словарной статье.

Множество M_{il} в совокупности с дескрипторами d_i образует класс условной эквивалентности, который и является дескриптором. Это множество M_{il} выполняет функцию номинального определения, которое уточняет смысл дескриптора d_i , выбранного для обозначения этого класса условной эквивалентности.

Информационно-поисковые тезаурусы по методам создания и применения делятся на синхронные и несинхронные. Синхронные методы совмещают построение тезауруса, начиная с «пустого состояния», или «нуль-тезауруса», с процессом эксплуатации систем. Несинхронные методы предусматривают предварительное, априорное построение тезауруса до начала эксплуатации систем.

Однако независимо от указанных методов для построения информационно-поискового тезауруса необходимо:

- провести отбор ключевых слов;
- построить словарь дескрипторов;
- построить словарные статьи.

При синхронных методах указанные процедуры выполняются в динамическом режиме одновременно с созданием поискового массива и выполнением процедур поиска, что обеспечивает более точное отражение лексики вводимых документов, следовательно, более высокие характеристики полноты и точности поиска. Однако за эти достоинства необходимо платить некоторым увеличением эксплуатационной трудоемкости.

Предметная область

Объекты и свойства

Каждая из автоматизированных информационных систем (АИС) соотносится определенной частью реального мира, именуемой предметной областью, и является сферой проблемной ориентации этой системы. При этом каждая автоматизированная система ориентирована на выполнение определенных функций в соответствующей ей области применения.

При определении предметной области объект должен иметь относительно целостный характер и для целей обработки информации обладать конечным набором свойств. Под свойством понимается некоторая его характеристика, позволяющая устанавливать его сходство и различие по отношению к другим объектам. При этом свойства объекта могут быть индивидуальными и общими, присущими лишь единичным экземплярам и целому классу объектов соответственно. Например, индивидуальными свойствами отличаются друг от друга животные одного вида, станки одной модели, товары одного наименования, а общие свойства имеются у одного вида растений, у материальных ценностей различных наименований, объединенных в одну группу. Существенными являются свойства объекта, по которым он может идентифицироваться в модели предметной области.

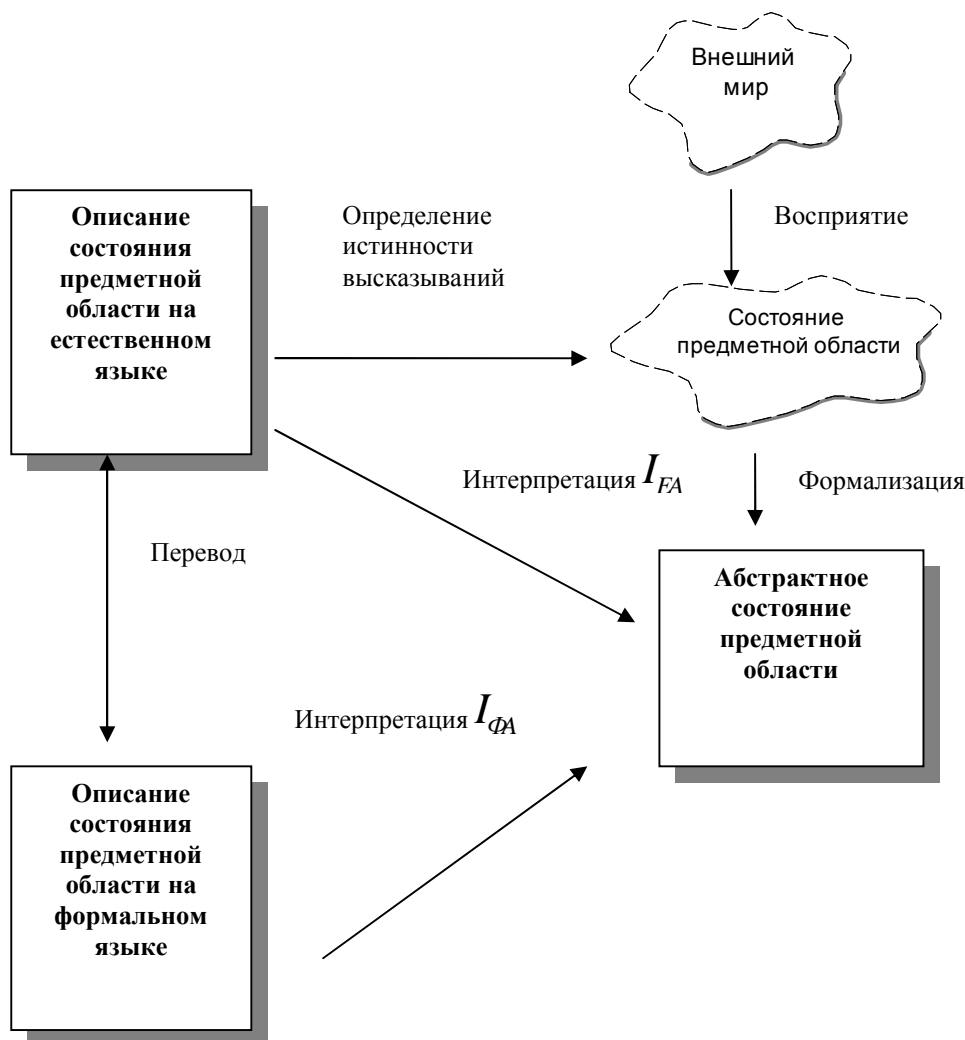
Под отношением понимается форма связи между объектами. Отношения подразделяются на внешние и внутренние. Внешние отражают связи между объектами, не затрагивая при этом их свойств. Например, внешними являются структурные отношения между элементами некоторой системы. Внутреннее отношение выявляет свойства объектов или обуславливает зависимость свойств объектов от данного отношения.

Описание предметной области должно вестись на общедоступном естественном языке. Для этого достаточно использовать предложения естественного языка, позволяющие именовать объекты и формулировать утверждения о том, что они обладают некоторыми свойствами, либо о том, что конкретные объекты находятся в определенных взаимоотношениях. Однако машины

могут оперировать только определенной формой описаний предметной области.

При этом под описанием предметной области на формальном языке всегда предполагается соотнесение с ее описанием на естественном языке. Введение формального языка приводит к необходимости однозначного перевода описания на одном языке (естественном или формальном) в описание на другом языке и к необходимости их эквивалентной интерпретации.

Эквивалентная интерпретация достигается введением абстрактных состояний предметной области, определенных формально и служащих однозначной интерпретацией описания состояния как на естественном, так и на формальном языке.



Описания на естественном и формальном языках эквивалентны, если одно из них является результатом перевода другого и если их интерпретацией служит одно и то же абстрактное состояние.

Простая модель, удовлетворяющая указанным требованиям, отражает абстрактные состояния, как совокупность множества абстрактных объектов E , множества типов T (каждый тип $T_j \in T$ есть подмножество $E = \bigcup_{j \in T} T_j$) и совокупности отношений R , в которой каждое отношение $R_j^{T_j \in T}$ имеет степень n_j .

Интерпретация описания состояний предметной области на естественном языке I_{EA} есть соответствие:

- фраз, однозначно именующих различные реальные объекты, различным абстрактным объектам из E ;
- фраз, обозначающих свойства объектов, типам из T ;
- фраз, обозначающих взаимоотношения конкретных объектов, кортежам отношений

из R.

Один и тот же объект с различными свойствами может принадлежать нескольким различным типам. Предложение естественного языка есть утверждение о том, что некоторые объекты обладают определенным свойством или что между конкретными объектами существуют определенные взаимоотношения.

В первом случае предложение становится фактом (предложению соответствует истинное значение), если объекты в абстрактном состоянии принадлежат типам, определенным фразами, обозначающими свойства объектов. Во втором случае предложение становится фактом, если объекты, определяемые связанными друг с другом в предложении именами, образуют кортеж соответствующего отношения. Если указанные условия не выполняются, предложению соответствует значение «ложь», и предложение фактом не является.

Абстрактное состояние является моделью описания состояния на естественном языке, если имеется интерпретация I_{EA} , такая, что всем предложениям этого описания соответствует значение «истина», а всем предложениям, не принадлежащим описанию, - значение «ложь».

Пример описания предметной области приведен ниже.

Описание на естественном языке.

«Отдел № 50 есть отдел с названием отдела ОЭВМ и номером отдела № 50. Проект Ш-Р1 - проект, имеющий шифр Ш-Р1. Проект Ш-Р2 - проект, имеющий шифр Ш-Р2. Ресурс Р-АВ - ресурс, имеющий шифр Р-АВ. Отдел № 50 разрабатывает проект Ш-Р1. Проект Ш-Р1 потребляет ресурс Р-АВ до конечной даты 1.06.84».

Абстрактные состояния $\langle E, T, R \rangle$, представляющие состояние рассматриваемой предметной области, а также соответствие элементов описания предметной области обозначениям объектов, типов и отношений, приведены ниже:

$$E = \{l_1, l_2, \dots, l_{10}\}, \text{ где}$$

$$l_1 = \text{«отдел № 50»}$$

$$l_2 = \text{«№ 50»}$$

$$l_3 = \text{«ОЭВМ»}$$

$$l_4 = \text{«проект Ш-Р1»}$$

$$l_5 = \text{«Ш-Р1»}$$

$$l_6 = \text{«проект Ш-Р2»}$$

$$l_7 = \text{«Ш-Р2»}$$

$$l_8 = \text{«ресурс Р-АВ»}$$

$$l_9 = \text{«Р-АВ»}$$

$$l_{10} = \text{«1.06.84»}$$

$$T = \{T_1, T_2, \dots, T_8\}, \text{ где}$$

$$T_1 = \text{«быть отделом»}$$

$$T_2 = \text{«быть номером отдела»}$$

$$T_3 = \text{«быть названием отдела»}$$

$$T_4 = \text{«быть проектом»}$$

$$T_5 = \text{«быть шифром проекта»}$$

$$T_6 = \text{«быть ресурсом»}$$

$$T_7 = \text{«быть шифром ресурса»}$$

$$T_8 = \text{«быть датой»}$$

$$T_1 = \{l_1\}, T_2 = \{l_2\}, T_3 = \{l_3\}, T_4 = \{l_4, l_6\}, T_5 = \{l_5, l_7\}, T_6 = \{l_8\}, T_7 = \{l_9\}, T_8 = \{l_{10}\}$$

$$R = \{R_1, R_2, \dots, R_6\},$$

$$R_1 = \text{«отдел имеет номер отдела»}$$

$$R_2 = \text{«отдел имеет название»}$$

$$R_3 = \text{«отдел разрабатывает проект»}$$

$$R_4 = \text{«проект потребляет ресурс до конечной даты»}$$

$$R_5 = \text{«проект имеет шифр проекта»}$$

$$R_6 = \text{«ресурс имеет шифр ресурса»}$$

$$R_1 = \{<l_1, l_2>\},$$

$$R_2 = \{<l_1, l_3>\},$$

$$R_3 = \{<l_1, l_4>\},$$

$$R_4 = \{<l_4, l_8, l_{10}>\},$$

$$R_5 = \{<l_4, l_5>, <l_6, l_7>\},$$

$$R_6 = \{<l_8, l_9>\}.$$

Основные принципы описания состояния предметной области на некотором формальном языке следующие:

- информационное содержание раскрывается посредством описания состояния предметной области на естественном языке, позволяющем соотносить элементы описания реальным объектам;
- формальное установление факта эквивалентности описаний состояния на различных языках требует их абстрактной интерпретации в терминах некоторой эталонной модели;
- извлечение информации требует обязательного перехода от формального описания к описанию на естественном языке.

В автоматизированных системах сведения об окружающем нас мире представляются посредством определенного набора понятий. В качестве таких понятий выступают единицы информации и информационные отношения. Рассмотрим их на примере экономических данных.

Единицы информации

Информационные единицы бывают элементарными и составными. Рассмотрим их на примере накладной. В качестве элементарных единиц информации выступают реквизиты - логически неделимые элементы, соотносимые с определенным свойством отображаемого объекта или процесса. Форма реквизита определяет полное его наименование, тип, описание множества допустимых значений и другие характеристики.

	R_1	R_2	R_3	R_4	
C_{11}	Приказ-накладная на отпуск готовых изделий № 19	Дата 8.11	Вид операций 51	Склад 4	
C_1	Получатель				
	R_5	R_6	R_7		
C_{12}	Наименование	Шифр	Адрес		
	Завод МЭЛЗ	132	г. Москва, ул. 1 Мая, 1		
	R_8	R_9			
C_{13}	Платежное требование	№899 от 8.02.2002			C_{11}
	Вид упаковки	ящики			R_{10}
	Станция назначения	Москва-товарная 11			R_{11}
	Основание	Договор № 20 от 6.01.2002			R_{12}
	R_{13}	R_{14}	Q_1	C_{21}	Q_4
	Наименование, сорт, размер	Номенклатурный номер	Стоимость, руб-коп	Количество	Стоимость, руб-коп
				По наряду	Отпущено
C_{21}	Подшипники	11250	212-30	100	100
C_{22}	Кольца С4-15	11781	147-25	30	27
C_{23}	Сепараторы	11261	123-25	180	180
C_{24}					
C_{25}					
C_{26}					
			Q_2	Q_3	

Существует ряд типов реквизитов в зависимости от значений, которые они могут принимать. Наиболее распространенными являются числовой и текстовый реквизиты.

Числовые реквизиты характеризуют количественные свойства экономических явлений, полученные в результате подсчета натуральных единиц, измерения, взвешивания, вычисления и т. п. Значениями таких реквизитов являются числа. Примерами служат числовые реквизиты Q_1 и Q_4 .

Текстовые реквизиты отражают качественные свойства экономических явлений, дают характеристику тем обстоятельствам, при которых протекало то или иное экономическое явление или процесс и были получены те или иные числовые значения. Текстовые реквизиты называют также реквизитами-признаками, их использование позволяет однозначно трактовать тот или иной хозяйственный процесс, не допуская смыслового искажения или смещения различных процессов или явлений. В качестве текстовых реквизитов в примере выступают СКЛАД, НАИМЕНОВАНИЕ ПОЛУЧАТЕЛЯ, АДРЕС ПОЛУЧАТЕЛЯ. Над числовыми реквизитами, как правило, в процессе обработки выполняются арифметические операции, а над текстовыми - логические. Кроме текстового и числового типов, часто применяется также логический тип, который определяет, какое из двух значений имеет величина - истину или ложь.

Каждый из наблюдаемых "объектов или процессов характеризует ряд присущих ему свойств. Изолированно взятый реквизит, характеризующий одно из свойств, не может служить полной информацией о наблюдаемом объекте или процессе. Для воспроизведения некоторого

сообщения об объекте, определенной информации о процессе требуется некоторая взаимосвязанная совокупность реквизитов.

Единицу информации, состоящую из совокупности других единиц, связанных между собой, называют составной единицей информации (СЕИ). Единицу информации, входящую в СЕИ, называют составляющей единицей информации. Для $C = (R_1, R_2, \dots, R_m)$ составляющими являются реквизиты R_1, R_2, \dots, R_m . Составной единицей является, например, ДАТА:

ДАТА = (ЧИСЛО, МЕСЯЦ, ГОД).

Хотя во многих случаях она рассматривается как один реквизит. Следует отметить, что существуют и другие реквизиты, кроме даты, которые могут быть составными. К таким реквизитам относятся множественные реквизиты, отражающие тот факт, что отображенное свойство объекта может, иметь несколько значений при представлении объекта, процесса или явления. В качестве множественного реквизита может выступать реквизит ЗНАНИЕ ИНОСТРАННЫХ ЯЗЫКОВ в кадровой анкете. Отдельная личность может владеть несколькими иностранными языками.

Структуру приказа-накладной можно представить составными единицами информации следующим образом:

$$\begin{aligned}C &= (C_1, C_2); \\C_1 &= (C_{11}, C_{12}, C_{13}); \\C_{11} &= (R_1, R_2, R_3, R_4); \\C_{12} &= (R_5, R_6, R_7); \\C_{13} &= (C_{131}, R_{10}, R_{11}, R_{12}); \\C_{131} &= (R_8, R_9); \\C_2 &= (R_{13}, R_{14}, Q_1, C_{21}, Q_4); \\C_{21} &= (Q_3, Q_4).\end{aligned}$$

В составных единицах информации выделяют промежуточные единицы - группы. Группа, состоящая только из реквизитов, называется простой, а группы, имеющие в своем составе другие группы, называются составными. Различают повторяющиеся и неповторяющиеся группы. Повторяющиеся группы допускают, как и множественные реквизиты, определенное количество реализаций:

простые группы $C_{11}, C_{12}, C_{131}, C_{21}$;

составные группы C, C_1, C_{13}, C_2 ;

повторяющиеся группы C_2 ;

неповторяющиеся - остальные (кроме C_2).

Из возможных структурных образований - групп особого внимания заслуживает группа, называемая показателем.

Показатель - это составная единица информации, состоящая из одного реквизита числового типа, именуемого основанием показателя и отражающего тот или иной факт в количественной оценке, и ряда характеризующих его и связанных с ним реквизитов-признаков текстового типа (времени, места действия, действующих лиц, предметов, продуктов труда и т. д.).

Общий вид показателя можно представить следующим образом: $\Pi(1:N)(P_1, P_2, \dots, P_n, Q)$,

где P_1, P_2, \dots, P_n - реквизиты-признаки; Q - реквизит-основание показателя.

Одной из причин выделения показателей в особую разновидность составных единиц информации является то, что показатель, по существу, служит минимальной по составу информационной совокупностью, достаточной для образования самостоятельного документа. Этот документ может существовать изолированно.

Показатели представляются, с одной стороны, простейшими СЕИ, способными к

документированию, а с другой - сложными образованиями информации, охватывающими описание многообразных качественных свойств и количественной характеристики объекта и состоящими в результате этого из целого ряда реквизитов. Например, в приказе-накладной показателем является

ЦЕНА ИЗДЕЛИЯ: $\Pi_y(1:N)(R_{13}, R_{14}, Q_1)$.

Составную единицу информации любой сложности можно свести в конечном итоге к определенной совокупности различных показателей. Например, приказ-накладную можно представить в виде четырех показателей: цена изделия (Π_y), количество отпущенных со склада

изделий ($\Pi_{ко}$), количество изделий, предназначенных на отпуск ($\Pi_{кн}$), сумма стоимости отпущенных со склада изделий (Π_c):

$\Pi_y(1:N)(R_{13}, R_{14}, Q_1)$.

$\Pi_{кн}(1:N)(P_2, P_3, P_4, P_5, P_6, P_7, P_{10}, P_{11}, P_{13}, P_{14}, Q_2)$.

$\Pi_{ко}(1:N)(P_2, P_3, P_4, P_5, P_6, P_7, P_{10}, P_{11}, P_{13}, P_{14}, Q_3)$.

$\Pi_c(1:N)(P_2, P_3, P_4, P_5, P_6, P_7, P_{10}, P_{11}, P_{13}, P_{14}, Q_4)$.

Особый интерес представляют показатели со значением основания, равным 1. Такие показатели внешне выступают с составом, включающим одни признаки. При основании, равном 1, количество информации будет больше в том показателе, который содержит большее число признаков, поскольку он дает более разностороннюю характеристику данному экономическому хозяйственному процессу или явлению.

Можно ввести условие, что форма показателя с минимальным составом, включающая одно основание и один признак, содержит единичное количество информации. Это позволяет ввести логарифмическую меру и выразить количество информации, содержащейся в показателе в следующем виде:

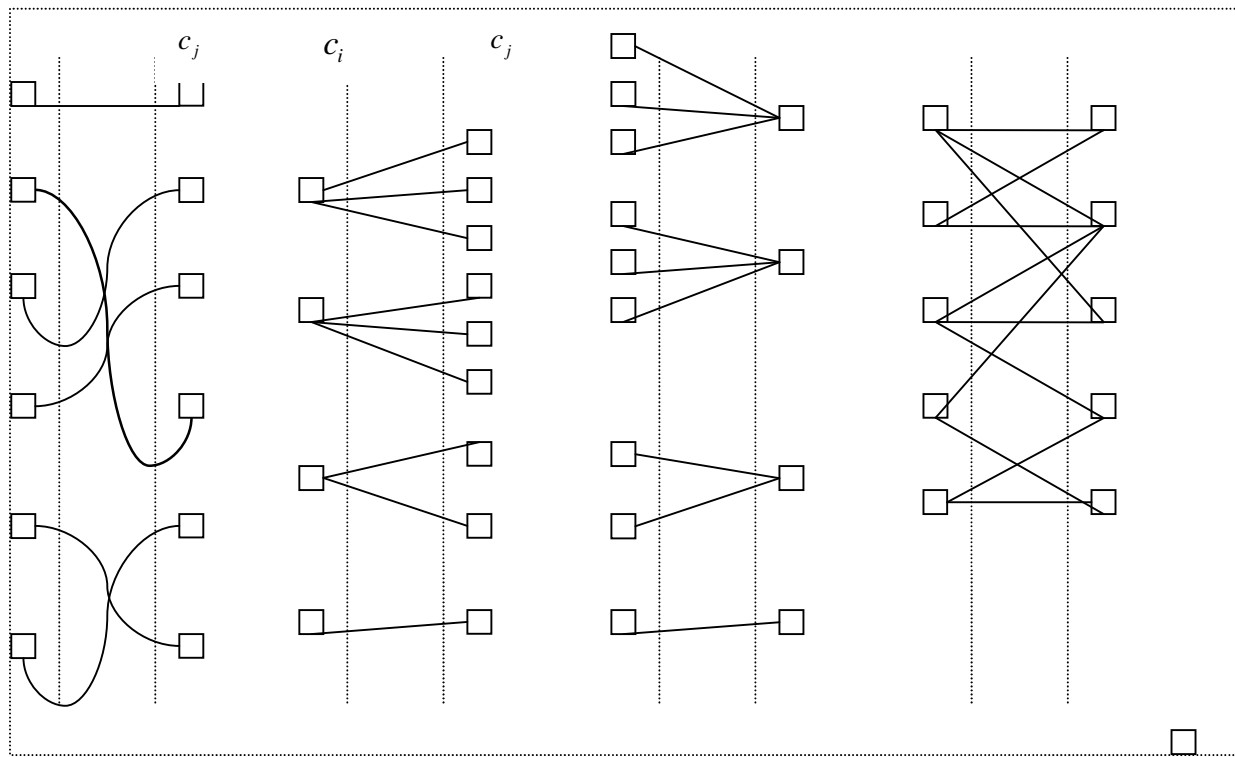
$$I(\Pi) = \log_2(1+Q) \log_2\left(1 + \sum_{q=1}^n q\right), \text{ где } n - \text{число реквизитов-признаков в показателе.}$$

В литературе часто показатели подразделяются по единице измерения основания на абсолютные и относительные. К абсолютным отнесены показатели, основания которых получают прямым счетом, измерением, взвешиванием, алгебраическим суммированием других абсолютных показателей, различные средние абсолютные показатели. В число относительных входят показатели, значения оснований которых получены отношением двух других показателей {удельный вес, отношение двух частей и т. д.).

Информационные отношения

Между различными объектами, свойствами объектов и объектами существуют всевозможные объективные отношения, определяемые предметной областью (пространственные, временные, функциональные и т. п.). При информационном отображении объектов и присущих им свойств эти отношения переносятся на отношения между составными единицами информации, с помощью которых представляются объекты и их свойства.

Для установления связи между составными единицами информации используется понятие «групповое отношение». Групповым отношением называется бинарное отношение $S_k \subseteq G_i \times G_j$, заданное на двух множествах групп G_i и G_j . Групповое отношение позволяет устанавливать связи между группами и тем самым выражать отношения между объектами. Групповые отношения различаются прежде всего тем, как связаны между собой конкретные представители множеств групп G_i и G_j в групповом отношении. Часто основные виды групповых отношений называют отношениями 1:1, 1 : п, п: 1 и m : п, а указанные пары чисел коэффициентом группового отношения.



В отношении с коэффициентами группового отношения вида 1:1, 1 : п, т: п группа q_i называется родительской, а q_j - подчиненной. В отношении п:1, наоборот, q_j - родительская, а q_i - подчиненная.

Иерархическим называется групповое отношение, в котором каждая подчиненная группа может быть связана только с одной родительской. Отношения вида 1 : 1, 1 : п, п : 1 являются иерархическими.

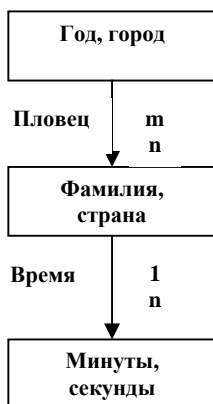
Неиерархическое отношение, напротив, позволяет соотносить каждую подчиненную группу с произвольным числом родительских групп. Иногда групповые отношения вида 1:1, 1 :п, п: 1 называют функциональными. Действительно, в первом и третьем случаях S_k - функция, а во втором - функцией является отношение S_k^{-1} .

Реализацией группового отношения S_k называют группу $q_i \in G_i$ вместе с совокупностью групп $I_{q_i} = \{q_j^1, q_j^2, \dots, q_j^l\} \subset G_j$, таких, что $\langle q_i, q_j \rangle \in S_k$ для любых $q_j \in I_{q_i}$. Говорят, что q_i - владелец реализации группового отношения, а $q_j \in I_{q_i}$ - члены реализации.

Графически группы и групповые отношения удобно изображать в виде диаграмм, используя следующие соглашения. На диаграммах группы изображаются прямоугольником, над левым верхним углом которого помещается имя группы. Внутри прямоугольника могут быть показаны имена реквизитов. Групповые отношения обозначаются стрелками, проведенными от родительского к подчиненному типу группы с возможным указанием коэффициента группового отношения (1:1, 1 : п, л:1, m: п) и имени группового отношения.

Если диаграмму трактовать как ориентированный граф и группы в диаграмме отождествлять с вершиной графа, а отношения - с дугой графа, то путем в диаграмме называется такая последовательность групповых отношений (дуг), что конец каждой предыдущей дуги совпадает с началом следующей.

Олимпиада



Модель "сущность-связь"

Многоуровневые представления данных

При изучении модели данных следует распознавать логические представления данных, к которым имеет отношение эта модель. Определим четыре уровня представления данных:

- (1) Информация, относящаяся к объектам и связям, которые существуют в нашем воображении.
- (2) Структура информации - организация информации, в которой объекты и связи представляются данными.
- (3) Структура данных, независимая от способа доступа, - структуры данных, которые не связаны со схемами поиска, индексации и др.
- (4) Структура данных, зависящая от способа доступа.

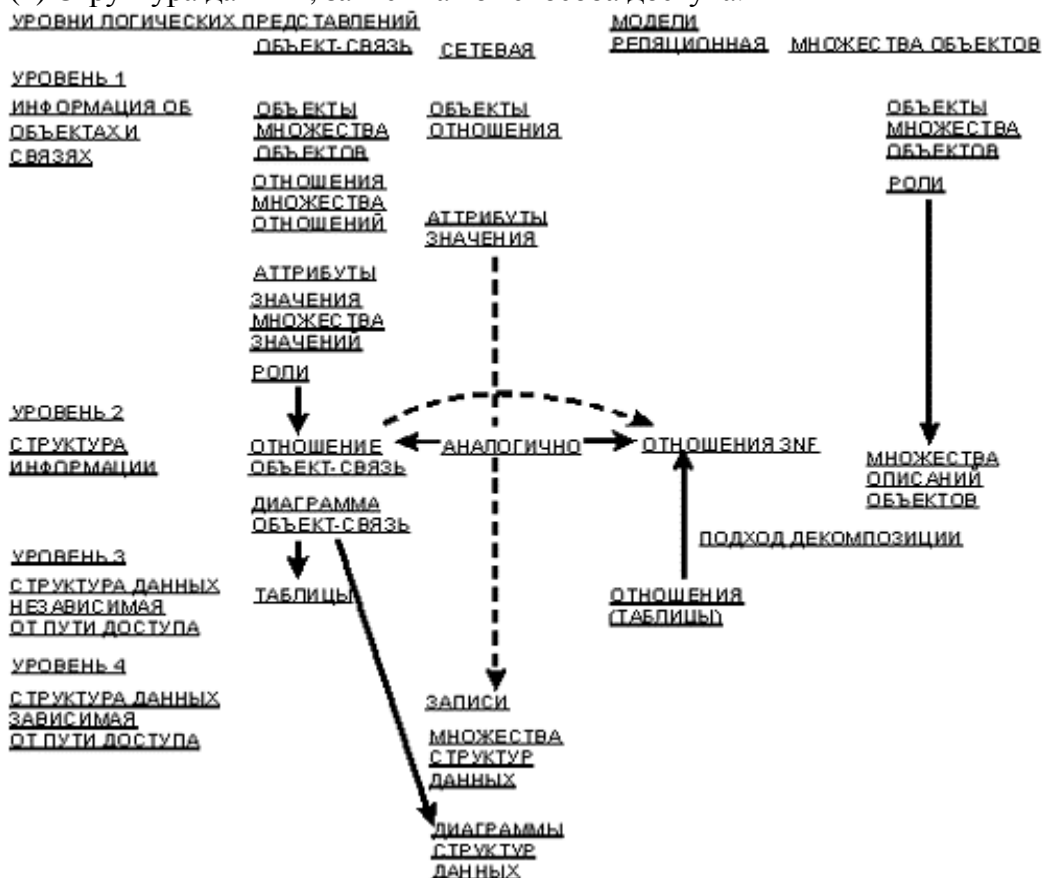


Рисунок 9. Анализ моделей данных с использованием нескольких уровней логических представлений

Информация о сущностях и связях (уровень 1)

На этом уровне мы рассматриваем сущности и связи. **Сущность (entity)** - это "предмет", который может быть идентифицирован некоторым способом, отличающим его от других "предметов". Конкретные человек, компания или событие являются примерами сущности. **Связь (relationship)** - это ассоциация, устанавливаемая между сущностями. Например, "отец - сын" - это связь между двумя сущностями "человек". База данных предприятия содержит информацию о сущностях и связях, которые представляют интерес для этого предприятия. В базу данных предприятия не может быть занесено полное описание сущности или связи. Невозможно и не обязательно сохранять всю потенциально доступную информацию о сущностях и связях. Будем рассматривать только те сущности и связи (и информацию о них), которые должны войти в проект базы данных.

Сущность и набор сущностей

Пусть E обозначает сущность, которая существует в воображении. Каждая сущность относится к некоторому отличному от других **набору сущностей (entity set)**, таким как Служащий, Проект или Отдел. С каждым набором сущностей связывается предикат, позволяющий проверить, принадлежит ли сущность данному набору. Например, если сущность относится к набору сущностей Служащий, то мы также знаем, что эта сущность обладает свойствами, общими с другими сущностями из набора объектов Служащий. В число этих свойств входит упомянутый предикат. Пусть E_i обозначает набор сущностей. Наборы сущностей не обязаны быть непересекающимися. Например, сущность, принадлежащая набору объектов Мужчины, принадлежит также и набору сущностей Человек. В этом случае набор объектов Мужчины является подмножеством набора объектов Человек.

Связь, роль и набор связей

Рассмотрим ассоциации сущностей. *Набор связей (relationship set)* R_i - это математическое отношение между n сущностями, каждая из которых относится к некоторому набору сущностей

$$\{[e_1, e_2, \dots, e_n] \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}.$$

и каждый кортеж сущностей $[e_1, e_2, \dots, e_n]$ является *связью (relationship)*. В этом определении наборы не обязаны быть различными, так, например, Брак - это связь между двумя сущностями из набора сущностей Человек.

Роль (role) сущности в связи - это функция, которую сущность выполняет в данной связи. Так, в примере выше, Муж и Жена - это роли. Упорядочивание сущностей в определении связи может отсутствовать, если в связи явно указаны роли сущностей: $(r_1 / e_1, r_2 / e_2, \dots, r_n / e_n)$, где r_i - это роль сущности e_i в данной связи.

Атрибут, значение и набор значений

Информацию об объекте или связи получают путем наблюдения или измерения и выражают множеством пар "атрибут-значение". , "3", "красный", "Петя" и "Иванов" - это значения. Значения классифицируются в различные **наборы значений (value sets)**, такие как Цифра, Цвет, Имя и Фамилия. С каждым набором значений связывается предикат для проверки того, принадлежит ли значение этому набору. Значение из некоторого набора значений может быть эквивалентно другому значению из другого набора значений. Например, "12" из набора значений «дюймы» эквивалентно "1с" в множестве значений «футы».

Атрибут (attribute) может быть формально определен как функция, отображающая набор сущностей или набор связей в набор значений или декартово произведение наборов значений:

$$f : E_i \vee R_i \rightarrow V_i \vee V_{i_1} \times V_{i_2} \times \dots \times V_{i_n}$$

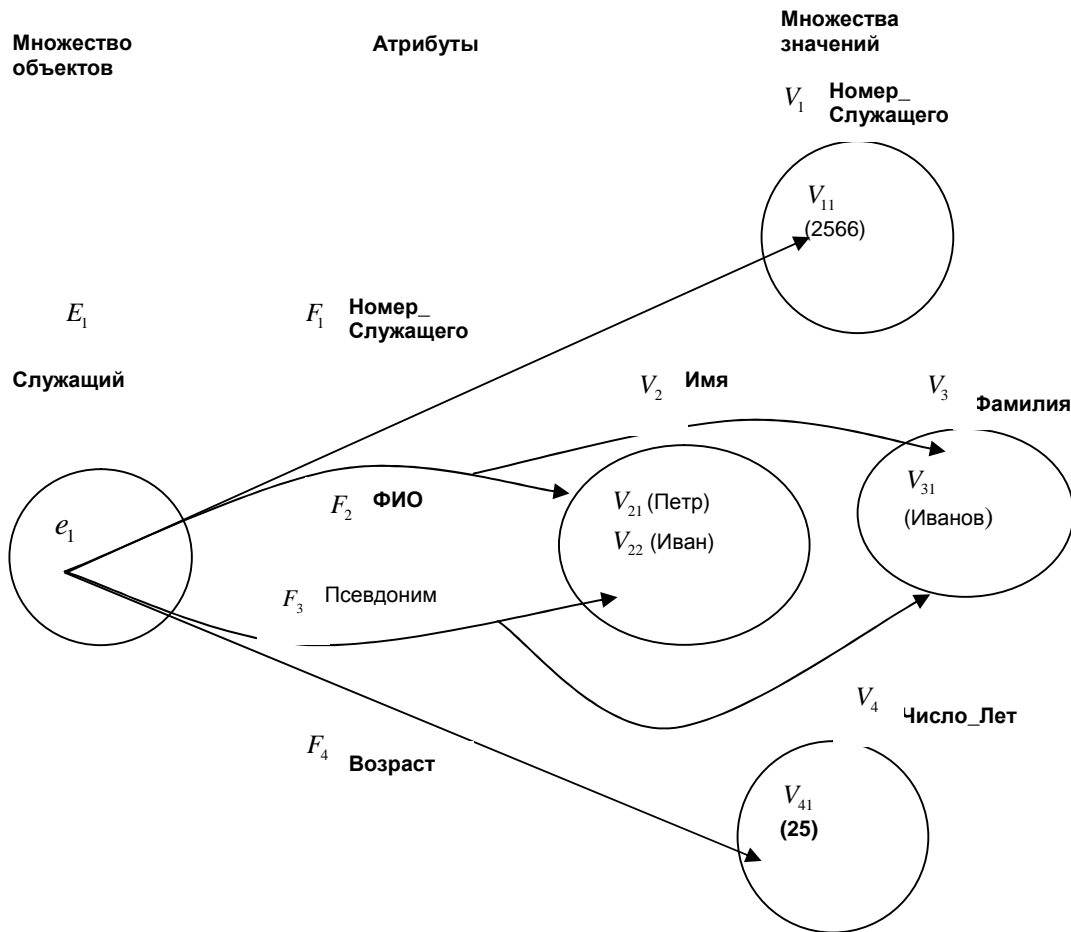


Рисунок 10 Атрибуты, определенные на наборе сущностей Человек

На рис. 10 показаны несколько атрибутов, определенных на множестве объектов Человек. Атрибут Возраст производит отображение в набор значений Число_Лет. Атрибут может задавать отображение в декартово произведение наборов значений. Например, атрибут ФИО задает отображение во множества значений Имя и Фамилия. Несколько атрибутов могут задавать отображение одного и того же набора объектов в один и тот же набор значений (или одну и ту же группу набора значений). Например, атрибуты ФИО и Псевдоним задают отображение из множества объектов Служащие в наборы значений Имя и Фамилия. Тем самым, атрибут и набор значений являются различными понятиями, хотя в некоторых случаях они могут иметь одно и то же имя (например, атрибут Номер_Служащего задает отображение из Служащие в набор значений Номер_Служащего). Это различие не является явным в сетевой модели и во многих существующих системах управления данными. Атрибут определяется как функция. Он отображает данный объект в одно значение (или один набор значений в случае декартова произведения наборов значений). Связи также имеют атрибуты. Рассмотрим множество связей Исполнитель_Проекта (рис. 11). Атрибут Количество_Времени, представляющий долю времени, выделенную конкретному служащему на конкретный проект, - это атрибут, определенный на наборе связей Исполнитель_Проекта. Он не является ни атрибутом сущности Служащий, ни атрибутом сущности Проект, так как его смысл зависит и от служащего, и от проекта. Понятие атрибута связи важно для понимания семантики данных и определения функциональных зависимостей между данными.

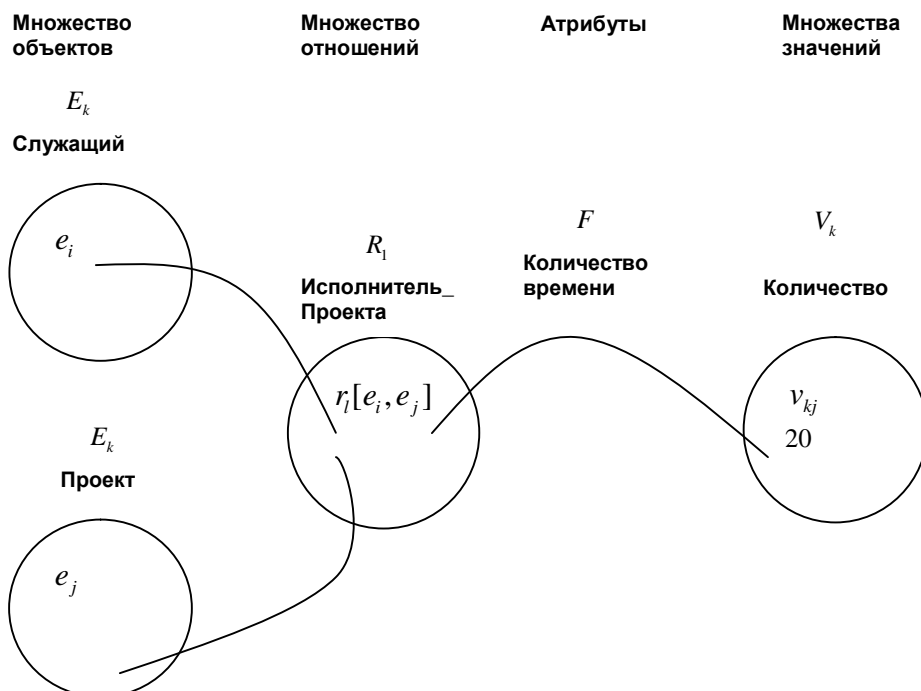


Рисунок 11. Атрибуты, определенные на наборе связей Исполнитель_Проекта.

Концептуальная структура информации

Рассмотрим организацию информации о сущностях и связях посредством разделения информации о сущностях и информации о связях. Такое разделение полезно для идентификации функциональных зависимостей между данными.

На рис. 12 в форме таблицы приведена информация о сущностях в наборе сущностей. Каждая строка значений относится к одной и той же сущности, а каждый столбец относится к множеству значений, которое, в свою очередь, относится к атрибуту. Порядок строк и столбцов неважен.

Атрибуты	F_1	F_2	F_3	F_4
Множество объектов	Номер служащего	Имя	Псевдоним	Возраст
Множество значений	E_1	V_2	V_3	V_4
Служащий	Номер служащего	Имя	Имя	Число лет
e_1	v_{11} (2556)	v_{21} Петр	v_{31} Иванов	v_{41} (25)
e_2	v_{12} (3378)	v_{22} Мария	v_{32} Чен	v_{42} (23)
.
.
.

Рисунок 12. Информация о сущностях из набора сущностей (табличная форма)

Служащий	Проект	
		Количество времени
E_i Служащий	E_j Проект	V_k Количество
e_{i1}	e_{j1}	v_{k1} (20)
.	.	.
.	.	.
.	.	.

Рисунок 13. Информация о связях из набора связей (табличная форма)

На рис. 13 приведена информация о связях в наборе связей. Каждая строка значений относится к связи, которая показана группой сущностей, где каждая сущность играет определенную роль и принадлежит определенному набору объектов.

На рис. 12 и 10 (а также на рис. 13 и 11) представлены различные формы одной и той же информации. Форма таблицы используется для простоты связи с реляционной моделью.

Структура информации (уровень 2)

Сущности, связи и значения на уровне 1 являются концептуальными объектами, существующими в воображении. На уровне 2 рассматриваются представления концептуальных объектов. Предполагается, что существуют непосредственные представления значений



Рисунок 14. Представление сущностей значениями (номерах служащих)

Первичный ключ

На рис. 2 значения атрибута Номер_Служащего могут использоваться для идентификации сущностей в наборе сущностей Служащий, если каждый служащий имеет отдельный номер служащего. Для идентификации сущностей в наборе сущностей может понадобиться более одного атрибута. Для идентификации сущностей могут использоваться несколько групп атрибутов. По существу, **ключ сущности (entity key)** - это группа атрибутов, такая, что отображение из набора сущностей в соответствующую группу наборов значений является взаимнооднозначным отображением. Если не удастся найти такое отображение на доступных данных, или если желательна простота в идентификации объектов, можно искусственно определить атрибут и набор значений, чтобы добиться наличия взаимнооднозначного отображения. В случае существования нескольких ключей обычно выбирается семантически значимый ключ в качестве **первичного ключа сущности (entity primary key - PK)**.

Рис. 14 получен объединением набора сущностей Служащий с набором значений Номер_Служащего с рис. 2. Обратим внимание на некоторые семантические импликации на рисунке 14. Каждое значение в множестве значений Номер_Служащего представляет объект (служащего). Атрибуты задают отображение из множества значений Номер_Служащего в другие множества значений. Атрибут Номер_Служащего задает отображение из множества значений Номер_Служащего в самого себя.

Отношения "объект/связь"

Информация об объектах в наборе объектов может быть организована в форме, показанной на рис. 7. Рис. 7 аналогичен рис. 12, за исключением того, что объекты представлены значениями их первичных ключей. Вся таблица на рис. 7 представляет *отношение сущностей (entity relation)*, а каждая строка представляет *кортеж сущностей (entity tuple)*.

<div>Атрибуты</div> <div>Множество значений (домен)</div> <div>Объект (набор)</div> <div>реляционное представление</div>	<div>← Основной ключ →</div>					
	Номер служащего	Имя		Псевдоним		Возраст
	Номер служащего	Имя	Фамилия	Имя	Фамилия	Число лет
	2366	Петр	Иванов	Виктор	Иванов	25
	3378	Мария	Чен	Маргарита	Чен	23
	⋮	⋮	⋮	⋮	⋮	⋮

Рисунок 15.Регулярное отношение сущностей Служащий

Так как связь идентифицируется вовлекаемыми в нее сущностями, *первичный ключ связи (primary key of a relationship)* может быть представлен основными ключами вовлеченных в связь сущностей. На рис. 8 вовлеченные сущности представлены их первичными ключами Номер_Служащего и Номер_Проекта. Имена ролей задают семантический смысл значений в соответствующих столбцах. Номер_Служащего - это первичный ключ вовлеченных в связь сущностей, а не атрибут связи. Количество_Времени - это атрибут связи. Таблица на рис. 8 представляет *отношение связей (relationship relation)*, а каждая строка значений - *кортеж связей (relationship tuple)*.

Имя отношения объектов	Основной ключ			
	Служащий	Проект		
Роль	Исполнитель	Проект		
Атрибут объекта	Номер служащего	Номер проекта	Количество времени	Атрибут связи
Множество значений (домен)	Номер служащего	Номер проекта	Количество	
Набор связей	представление	2566	31	20
	2173	25	100	
	.	.	.	
	.	.	.	
	.	.	.	

Рисунок 16.Регулярное отношение связей Исполнитель_Проекта.

В некоторых случаях сущности в наборе сущностей нельзя уникально идентифицировать значениями их собственных атрибутов; следовательно, для их идентификации следует использовать связи. Например, рассмотрим сущности служащих-подчиненных и служащих-начальников: подчиненные идентифицируются своими именами и значениями основного ключа служащих-начальников (т.е. их связями с этими служащими). На рис. 9 Номер_Служащего не является атрибутом объекта в наборе Подчиненный, а представляет собой первичный ключ служащих, которые имеют подчиненных. Каждая строка значений на рис. 9 - это кортеж сущностей с Номер_Служащего и ФИО в качестве первичных ключей. Вся таблица является отношением сущностей.

Теоретически, любой вид связи может использоваться для идентификации сущностей. Для простоты мы ограничимся только одним видом связи: бинарными связями с отображением 1:n, в которых существование n сущностей на одной стороне связи зависит от существования одной сущности на другой стороне связи. Например, один служащий может иметь n ($n = 0, 1, 2, \dots$) подчиненных, и существование подчиненных зависит от существования соответствующего служащего.

Этот метод идентификации сущностей связями с другими сущностями можно применять рекурсивно до тех пор, пока не встретятся сущности, которые могут быть идентифицированы значениями своих собственных атрибутов. Например, основной ключ департамента компании может состоять из номера департамента и основного ключа отделения, который в свою очередь состоит из номера отделения и имени компании.

Следовательно, мы имеем две формы отношений сущностей. Если связи используются для идентификации объектов, будем называть это **слабым отношением сущностей (weak entity relation)** (рис. 9). Если связи не используются для идентификации сущностей, будем называть это **регулярным отношением сущностей (regular entity relation)** (рис. 16). Если некоторые сущности в связи идентифицируются другими связями, будем называть это **слабым отношением связей (weak relationship relation)**. Например, любые связи между сущностями Подчиненный и другими сущностями приведут к образованию слабых отношений связи, так как сущность Подчиненный идентифицируется своим именем и связью с сущностью Служащий. Проведение различия между регулярными (сущность/связь) и слабыми (сущность/связь) отношениями полезно для поддержки целостности данных.

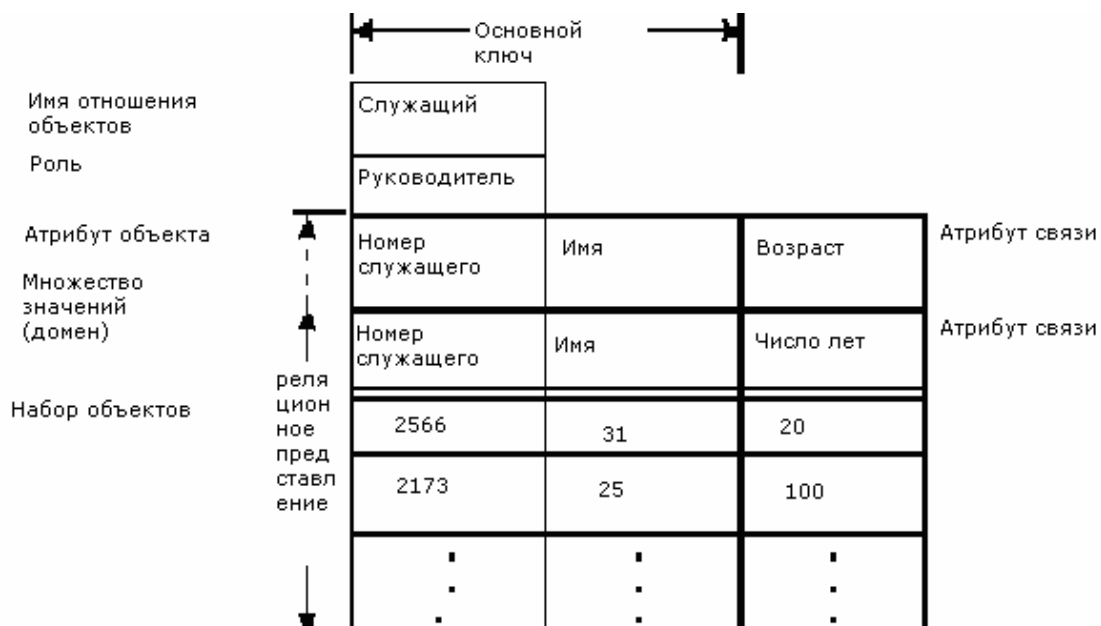


Рисунок 17. Слабое отношение сущностей Подчиненный

Диаграмма "сущность - связь" и включение семантики описания данных и манипулирование данными

Системный анализ с использованием диаграммы "сущность-связь"

Рассмотрим диаграмму "сущность-связь".

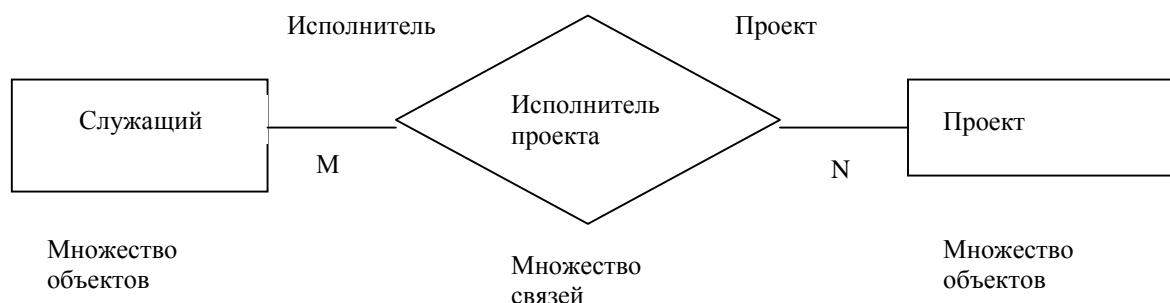


Рисунок 18. Простая диаграмма сущность-связь

На рис. 18 с использованием диаграммного метода демонстрируются набор связей Исполнитель_Проекта и наборы связей Служащий и Проект. Каждый набор сущностей представляется прямоугольником, а каждый набор связей - ромбом. Линии, соединяющие прямоугольники, демонстрируют тот факт, что набор связей Исполнитель_Проекта определен на наборах сущностей Служащий и Проект. Указаны роли сущностей в связи.

На рис. 11 приведена более полная диаграмма некоторых наборов сущностей и наборов связей, которые могли бы представлять интерес для производственной компании. Отдел, Служащий, Подчиненный, Проект, Руководитель и Деталь - это множества сущностей. Служащий_Отдела, Служащий_Подчиненный, Исполнитель_Проекта, Руководитель_Проекта, Поставщик_Деталей_Проекта, Детали_Проекта и Компонент - множества связей. Связь Компонент показывает, какие детали (и в каком количестве) требуются для создания составных деталей.

Рассмотрим несколько важных общих характеристик связей:

(1) Набор связей может быть определен на более чем двух наборах сущностей. Например, множество связей Поставщик_Деталей_Проекта определяется на трех наборах сущностей: Поставщик, Проект и Деталь.

(2) Набор связей может быть определен и только на одном наборе сущностей. Например, набор связей Компонент определяется на единственном наборе объектов Деталь.

(3) Может существовать несколько наборов связей, определенных на заданных наборах сущностей. Например, наборы связей Исполнитель_Проекта и Руководитель_Проекта определяются на наборах сущностей Проект и Служащий.

(4) В диаграмме могут различаться отображения 1:n, m:n и 1:1. Набор связей Служащий_Отдела является отображением 1:n, т.е. в одном департаменте может работать n ($n = 0, 1, 2, \dots$) служащих, и каждый служащий работает только на один департамент. Набор связей Исполнитель_Проекта является отображением m:n, т.е. в каждом проекте могут участвовать ноль, один и более служащих, и каждый служащий может участвовать в ноле, одном или более проектах. Можно выразить также отображение 1:1, такое, каким является набор связей Брак. Информация о допустимом набором связей числе сущностей в каждом наборе сущностей задается с помощью указания в диаграмме "1", "m", "n". Реляционная модель и модель множества объектов не включают этот тип информации; в сетевой модели невозможно просто выразить отображение 1:1.

(5) В диаграмме можно показать *зависимость существования (existence dependency)* одного типа сущностей от другого. Например, стрелка в наборе связей Служащий_Подчиненный показывает, что существование сущности в наборе сущностей Подчиненный зависит от соответствующей сущности в наборе сущностей Служащий. Другими словами, если служащий покидает компанию, его подчиненные, возможно, больше не будут представлять интерес.

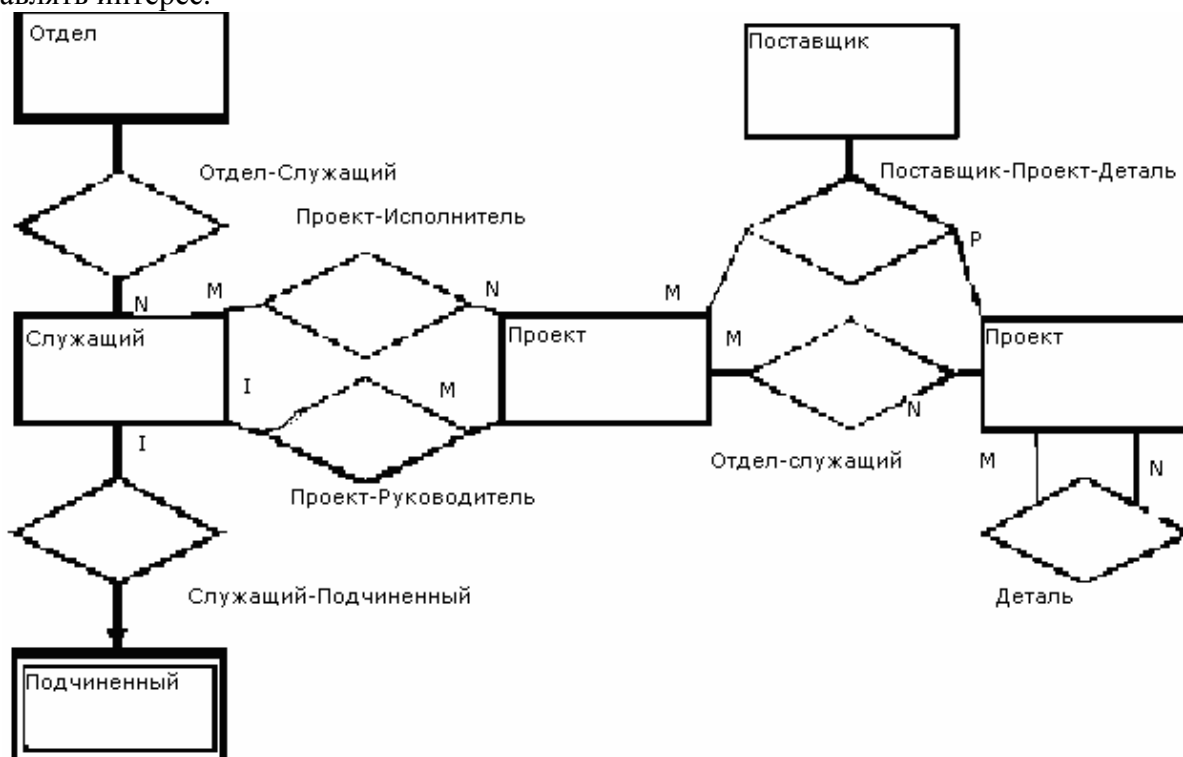


Рисунок 19. Диаграмма сущность-связь для анализа информации в производственной компании

Набор сущностей Подчиненный показан специальным прямоугольником. Это значит, что на уровне 2 информация о сущностях из этого набора организована как слабое отношение сущностей (с использованием первичного ключа Служащий как части первичного ключа Подчиненный).

Анализ других моделей данных и их порождение из модели "сущность - связь"

Реляционная модель

Реляционное представление данных и неоднозначность семантики

В реляционной модели *отношение (relation)* R - это математическое отношение, определенное на множествах X_1, X_2, \dots, X_n :

$$R = \{(x_1, x_2, \dots, x_n) \mid x_1 \text{ принадл } X_1,$$

x_2 принадл X_2, \dots, x_n принадл X_n

Множества X_1, X_2, \dots, X_n называются **доменами (domains)**, а (x_1, x_2, \dots, x_n) называется **кортежем (tuple)**. На рис. 12 показано отношение Служащий. Доменами отношения являются Номер_Служащего, Имя, Фамилия, Имя, Фамилия, Число_Лет. Порядок строк и столбцов в отношении не является существенным. Чтобы избежать неоднозначности столбцов отношения с одним и тем же доменом, имена доменов уточняются **ролями (roles)** (для выделения роли домена в отношении). Например, в отношении Служащий, домены Имя и Фамилия можно уточнить ролями Истинное или Псевдоним. **Имя атрибута (attribute name)** в реляционной модели - это имя домена и присоединенное к нему имя роли [10]. Сравнивая рис. 12 и 15, видим, что "домены" почти эквивалентны наборам значений. Хотя может показаться, что "роль" и "атрибут" в реляционной модели служат той же цели, что и "атрибут" в модели "сущность-связь", семантика этих терминов различается. "Роль" и "атрибут" в реляционной модели используются главным образом для того, чтобы различать домены с одним и тем же именем в одном и том же отношении, в то время как "атрибут" в модели "сущность-связь" - это функция, задающая отображение из набора сущностей (или связей) в наборы значений.

Роль		Истинное	Истинное	Псевдоним	Псевдоним	
Домен	Номер служащего	Имя	Фамилия	Имя	Фамилия	Число лет
Набор	2566	Петр	Иванов	Виктор	Иванов	25
	3378	Мария	Чен	Маргарита	Чен	23

Рисунок 20.Отношение Служащий

Номер проекта	Номер служащего
7	2566
3	3566
7	3378

Рисунок 21.Отношение Служащий_Проект

Использование реляционных операторов в реляционной модели может вызывать семантические неоднозначности. Например, в результате соединения отношения Служащий с отношением Служащий_Проект (рис. 21) по домену Номер_Служащего получается отношение Служащий_Проект (рис. 14). Рассмотрим соединение отношения Служащий с отношением Корабль по домену Число_Лет. Одно и то же имя домена может иметь различную семантику в различных отношениях (роль предназначена для различения доменов в данном отношении). Если не допускается, чтобы значения домена Число_Лет отношения Служащий были сравнимыми со значениями домена Число_Лет отношения Корабль, должны быть объявлены различные имена доменов.

		Истинное	Истинное	Псевдоним	Псевдоним	
Номер проекта	Номер служащего	Имя	Фамилия	Имя	Фамилия	Число лет
7	2566	Петр	Иванов	Виктор	Иванов	25
3	2566	Петр	Иванов	Виктор	Иванов	25
7	3378	Мария	Чен	Маргарита	Чен	23

Рисунок 22.Отношение Служащий_Проект как "соединение" отношений Служащий и Служащий_Проект

Номер корабля	Имя	Число Лет
037	Ушаков	25
056	Арктика	18

Рисунок 23.Отношение Корабль

В модели "сущность-связь" семантика данных более очевидна. Например, один столбец в рассмотренном примере содержит значения значения Возраст объекта Служащий, а другой столбец содержит Возраст объекта Корабль. Поскольку система баз данных содержит семантическую информацию, ей следует обладать возможностью предупреждать пользователя о потенциальных проблемах операции типа соединения.

Семантика функциональных зависимостей между данными

В реляционной модели "атрибут" В отношения *функционально зависим (functionally dependent)* от "атрибута" А того же отношения, если каждому значению А соответствует не более, чем одно значение В. Семантика функциональных зависимостей между данными становится очевидной в модели "сущность-связь". По существу, имеется два основных типа функциональных зависимостей:

(1) Функциональные зависимости, относящиеся к описанию сущностей или связей. Поскольку атрибут определяется как функция, он отображает сущность из набора сущностей в одно значение из набора значений (см. рис. 2). На уровне 2 для представления сущностей используются значения первичного ключа. Следовательно, неключевые наборы значений функционально зависимы от наборов значений первичного ключа (например, на рис. 22 и 23 Число_Лет функционально зависим от Номер_Служащего). Поскольку отношение может иметь несколько ключей, неключевые наборы значений будут функционально зависимыми от любого ключевого набора значений. Ключевые наборы значений будут взаимно функционально зависеть друг от друга. Аналогично, в отношении связей неключевые наборы значений будут функционально зависимы от наборов значений первичного ключа (например, на рис. 16 Количество_Времени функционально зависим от Номер_Служащего и Номер_Проекта).

(2) Функциональные зависимости, относящиеся к сущностям в связи. Заметим, что на рис. 19 мы указываем типы отображений (1:n, m:n, и т.д.) для наборов связей. Например, Руководитель_Проекта является отображением 1:n. Допустим, что Номер_Проекта - это первичный ключ в отношении сущностей Проект. В отношении связей Руководитель_Проекта набор значений Номер_Служащего будет функционально зависим от набора значений Номер_Проекта (т.е. для каждого проекта имеется только один менеджер).

Проведение различия между уровнем 1 (рис. 2) и уровнем 2 (рис. 22 и 23) и разделение отношения сущностей (рис. 23) и отношения связей (рис. 16) вносит ясность в семантику функциональных зависимостей между данными.

3NF-отношения 3NF в сравнении с отношениями "сущность-связь"

Из определения "отношения" следует, что любое группирование доменов можно рассматривать как отношение. Чтобы избежать нежелательных свойств в поддерживаемых отношениях, проводится процесс нормализации, в ходе которого произвольные отношения преобразуются в первую нормальную форму, затем во вторую нормальную форму и, наконец, в третью нормальную форму (3NF). Отношения сущностей и связей в модели "сущность-связь" похожи на 3NF-отношения, но обладают более ясной семантикой и не требуют выполнения операций преобразования.

Следующие три отношения находятся в первой нормальной форме (т.е. не включают домена, элементы которого сами являлись бы отношениями):

Служащий (Номер Служащего)

Деталь (Номер Детали, Описание_Детали, Количество_Имеющееся)

Детали_Проекта (Номер Детали, Номер Проекта, Описание_Проекта,
Номер_Руководителя_Проекта, Количество_Используемое)

Домен Номер_Руководителя_Проекта содержит Номер_Служащего менеджера проекта. Первичные ключи подчеркнуты.

В соответствии с определенными правилами эти отношения преобразуются в третью нормальную форму:

Служащий (Номер Служащего)

Деталь (Номер Детали, Описание_Детали, Количество_Имеющееся)

Проект (Номер Проекта, Описание_Проекта, Номер_Руководителя_Проекта)

Детали_Проекта (Номер Детали, Номер Проекта, Количество_Используемое)

Используя диаграмму "сущность-связь" с рис. 19, получим следующие отношения сущностей и связей:

сущность

Деталь (Номер Детали, Описание_Детали, Количество_Имеющееся)

отношения

Проект (Номер Проекта, Описание_Проекта)

Служащий (Номер Служащего)

связь Детали_Проекта (Деталь/Номер Детали, Проект/Номер Проекта,
Количество_Используемое)

отношения

Руководитель_Проекта (Проект/Номер Проекта, Руководитель/Номер Служащего)

Указываются имена ролей сущностей в связях (такие как Руководитель). Имена отношений сущностей, ассоциированных с первичными ключами сущностей в связях, и имена наборов значений пропущены.

В приведенном примере отношения "сущность-связь" похожи на 3NF-отношения. При использовании подхода 3NF Номер_Руководителя_Проекта включается в отношение Проект, поскольку предполагается, что Номер_Руководителя_Проекта функционально зависит от Номер_Проекта. В модели "сущность-связь" Номер_Руководителя_Проекта (т.е. Номер_Служащего менеджера проекта) включается в отношение связи Руководитель_Проекта, так как в этом случае Номер_Служащего рассматривается как первичный ключ сущности

При использовании подхода 3NF изменения функциональных зависимостей данных могут приводить к тому, что некоторые отношения перестают находиться в 3NF. Например, если для одного проекта могут быть несколько менеджеров, отношение Проект больше не будет 3NF-отношением, и его нужно будет разбить на два отношения - Проект и

Руководитель_Проекта. При использовании модели "сущность-связь" такие изменения не являются необходимыми. Следовательно, используя модель объект-связь можно привести данные к форме, аналогичной 3NF-отношениям, но с отчетливым семантическим смыслом.

Описанный подход декомпозиции (или преобразования) для нормализации отношений является подходом "снизу-вверх" в проектировании базы данных используется некоторая семантическая информация (функциональные зависимости данных) для преобразования отношений в 3NF-отношения. В модели "сущность-связь" принят подход "сверху-вниз", в котором используется семантическая информация для организации данных в отношения "сущность/связь".

Сетевая модель

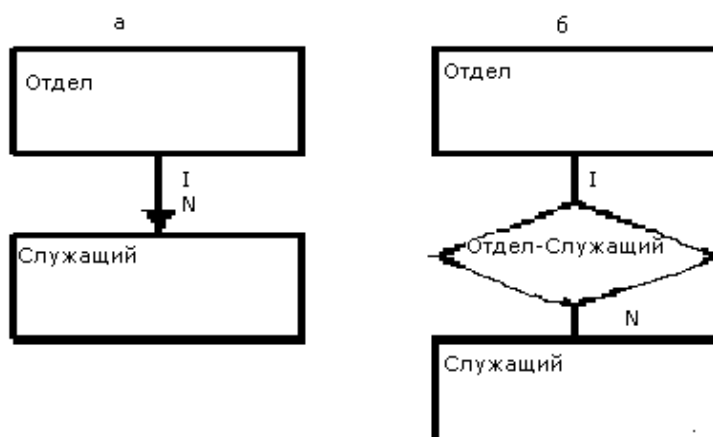


Рисунок 24.Связь Отдел_Служащий (а) диаграмма структур данных, (б) диаграмма "сущность-связь"

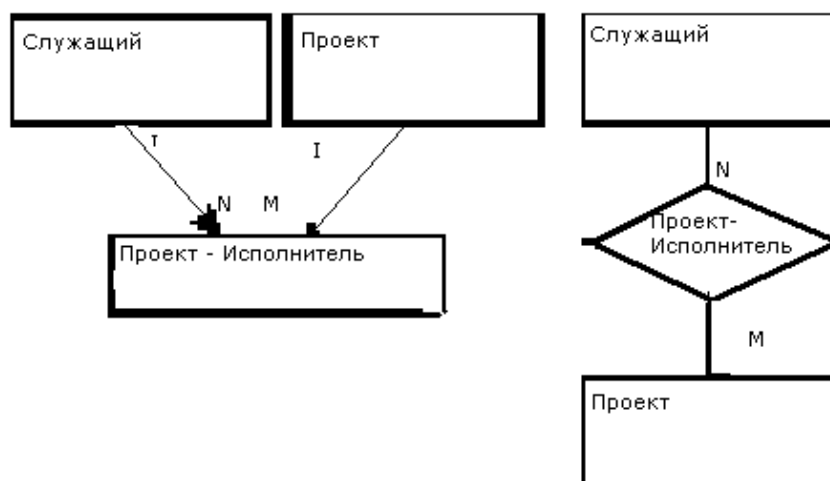


Рисунок 25.Связь Исполнитель_Проекта (а) диаграмма структур данных, (б) диаграмма "сущность-связь"

Семантика диаграммы структур данных

Один из лучших способов объяснения сетевой модели состоит в использовании *диаграммы структур данных (data-structure diagram)*. На рис. 24 показана диаграмма структур данных. Каждый прямоугольник представляет тип записи. Стрелка представляет набор структур данных, в котором запись Отдел является *записью-владельцем (owner-record)*, и одной записи-владельцу может соответствовать n (n = 0, 1, 2, ...) *записей-членов (member-records)*. На рис. 24(б) приведена диаграмма "сущность-связь". Стрелка на диаграмме структур данных не всегда показывает связь между сущностями в двух наборах сущностей. На рис. 25(а) и 25(б) показана диаграмма структур данных и диаграмма "сущность-связь", выражающие связь Исполнитель_Проекта между двумя типами сущностей - Служащий и Проект. На рис. 25(а)

связь Исполнитель_Проекта становится другим типом записи, и стрелки не представляют связь между сущностями. Стрелка представляет связь 1:n между двумя типами *записи* (а не сущности), и из ее наличия также следует существование пути доступа из записи-владельца к записям-членам. Диаграмма структур данных - это представление организации записей; оно не является точным представлением объектов и связей.

Порождение диаграммы структур данных

Сравнение диаграмм структур данных с соответствующими диаграммами "сущность-связь" позволяет вывести следующие правила:

- Для бинарных связей 1:n стрелка используется для представления связи (см. рис. 24(а)).
- Для бинарных связей m:n для представления связи создается тип "запись связи", и рисуются стрелки из типа "запись сущности" в тип "запись связи" (см. рис. 24(а)).
- Для k-арных связей ($k \geq 3$) применяется то же правило, что и в случае (2) (т.е. создание типа "запись связи").



Рисунок 26. Набор структур данных, определенных на одном типе записи

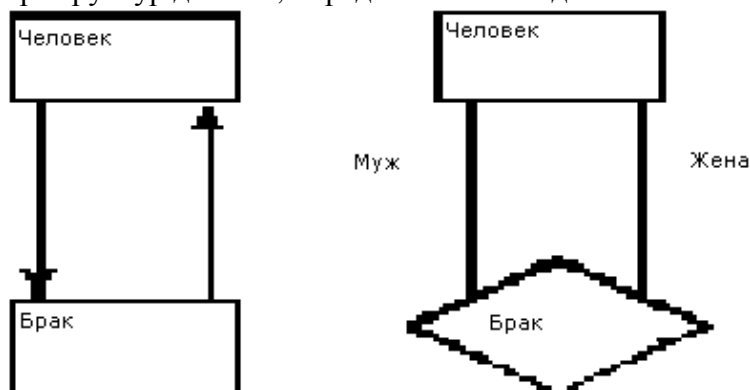


Рисунок 27. Связь Брак (а) диаграмма структур данных, (б) диаграмма "сущность-связь"

Поскольку в DBTG не разрешается определить набор структур данных на одном типе записи, для реализации таких связей необходима "запись связи" (см. рис. 27(а)). Соответствующая диаграмма "сущность-связь" показана на рис. 27(б).

Стрелки на диаграмме структур данных не всегда представляют связи сущностей. Даже в том случае, когда стрелка представляет связь 1:n, она показывает только однонаправленную связь (хотя имеется возможность найти запись-владельца по записи-члену). В модели "сущность-связь" представляются оба направления связи (специфицируются роли обеих сущностей). Помимо семантической неоднозначности стрелок, недостатком сетевой модели является неудобство обработки изменений семантики. Например, если связь между Отдел и Служащий из отображения 1:n превращается в отображение m:n (т.е. один служащий может принадлежать нескольким департаментам), в сетевой модели мы должны создать запись связи Отдел_Служащий. В модели "сущность-связь" все типы отображений обрабатываются единообразно.

Модель "сущность-связь" можно применять как средство при структурном проектировании баз данных с использованием сетевой модели. Вначале рисуется диаграмма

"сущность-связь" (рис. 27). Затем она переводится в диаграмму структур данных (рис. 20). Можно следовать дисциплине, в соответствии с которой каждая сущность или связь должны отображаться в запись (то есть, "записи связей" создаются для всех типов связей, независимо от того, являются они отображениями 1:n или m:n). Таким образом, нужно на рис. 27 заменить ромбы на прямоугольники и добавить концы стрелок к соответствующим линиям. При использовании данного подхода на рис. 20 появятся еще три прямоугольника - Отдел_Служащий, Служащий_Подчиненный и Руководитель_Проекта (см. рис. 21).

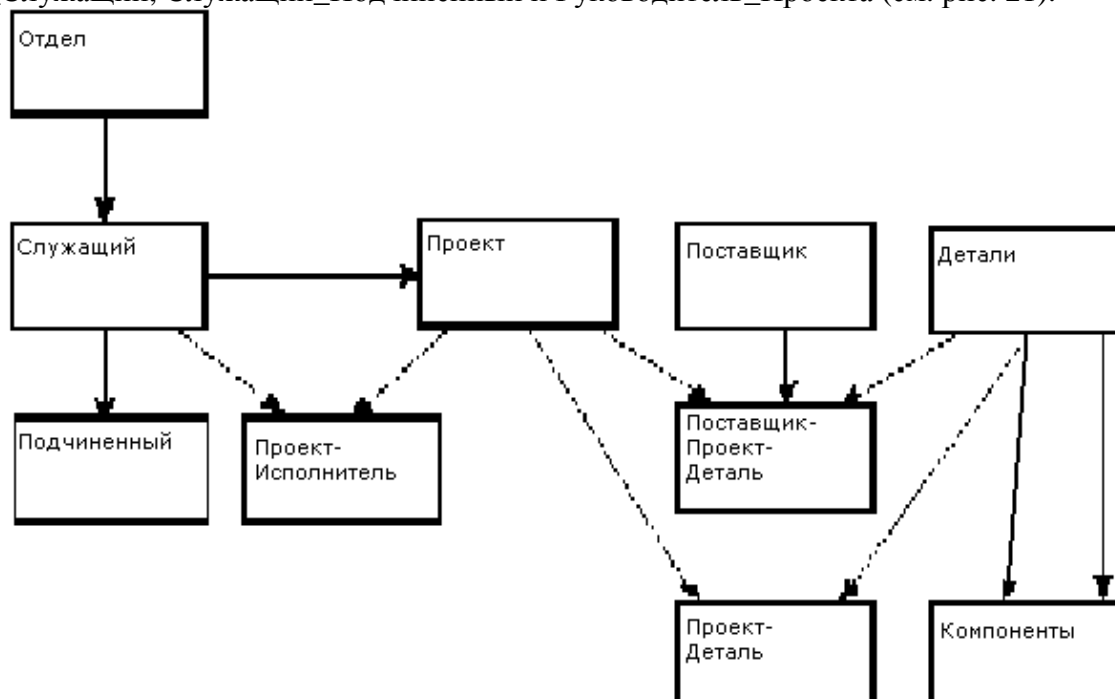


Рисунок 28.Диаграмма структур данных, выведенная из диаграммы "сущность-связь" с рис. 27

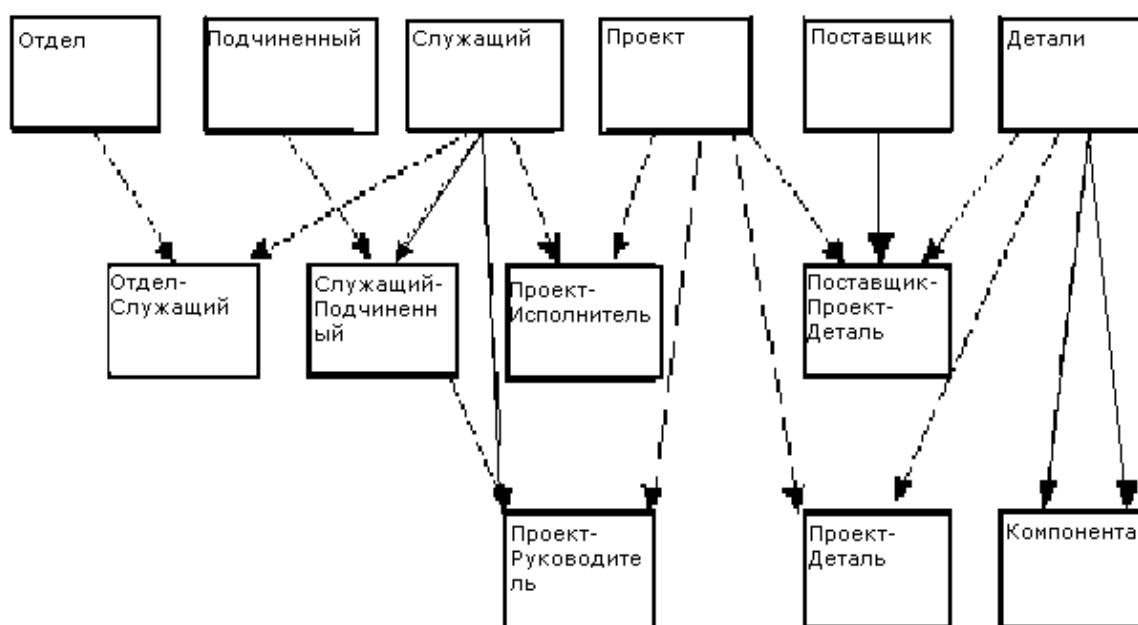


Рисунок 29."Упорядоченная" диаграмма структур данных, выведенная из диаграммы "сущность-связь" с рис. 27

Модель множества сущностей

Представление множества сущностей

Базовый элемент модели множества сущностей - это "сущность". Сущности имеют имена, такие как "Петр Иванов", "синий" или "22". Имена сущностей, имеющих некоторые общие свойства, собираются в *множества-имен-сущностей*, которые именуются *именами-множеств-имен-сущностей*, например, "Имя", "Цвет" и "Количество".

Сущность представляется парой "имя-множества-имен-сущностей/имя-сущности", например, Имя/ Петр Иванов, Номер_Служащего/2566 и Число_лет/20. Сущность описывается своей связью с другими сущностями. Рис. 30 иллюстрирует представление данных в модели множества сущностей. "Отдел" сущности Номер_Служащего /2566 - это сущность Номер_Отдела/405. Другими словами, "Отдел" - это роль, которую играет сущность Номер_Отдела /405 для описания сущности Номер_Служащего /2566. Аналогично, "Имя", "Псевдоним" или "Возраст" сущности Номер_Служащего /2566 - это " Имя/ Петр Иванов ", " Имя/ Александр Иванов " или "Число_Лет/20" соответственно. Описание сущности Номер_Служащего /2566 - это набор связанных сущностей и их ролей (сущности и роли обведены пунктирной линией). Пример *описания сущности* "Номер_Служащего /2566" представлен на рис. 31 набором триплетов имя-роли/имя-множества-имен-сущностей/имя-сущности. Концептуально, модель множества сущностей отличается от модели "сущность-связь" следующим:

(1) В модели множества сущностей все трактуется как объекты. Например, "Цвет/Черный" и "Число_Лет/45" - это сущности. В модели "сущность-связь" "синий" и "36" обычно трактуются как "значения". Заметим, что обработка значений как объектов может порождать семантические проблемы. Например, на рис. 30, какова разница между "Номер_Служащего /2566", " Имя/ Петр Иванов " и " Имя/ Виктор Иванов "? Представляют ли они различные объекты?

(2) В модели множества сущностей используются только бинарные связи, в то время как в модели "сущность-связь" можно использовать n-арные связи.

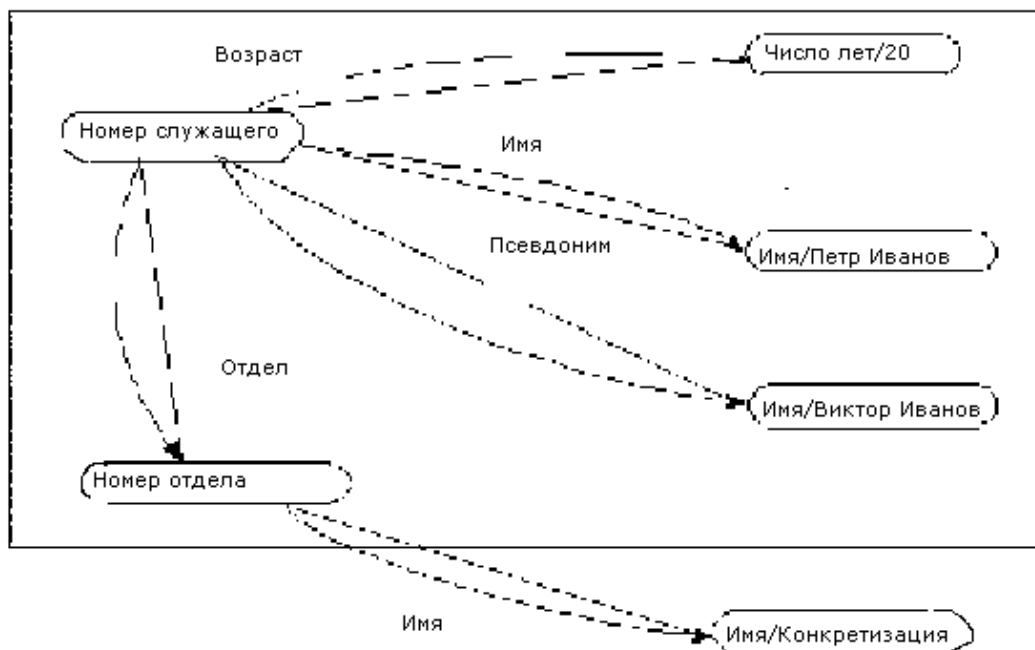


Рисунок 30. Представление множества объектов

Терминология модели объект-связь	Атрибут или роль	Множество значений	Значение
Терминология модели множества объектов	"Имя роли"	"Имя множества имен объектов"	"Имя объекта"
Идентификатор	Номер служащего	2566	
Имя	Имя	Петр Иванов	
Имя	Имя	Виктор Иванов	
Возраст	Число лет	25	
Отдел	Номер отдела	405	

Рисунок 31. Описание сущностей в модели множества сущностей

Глава 4

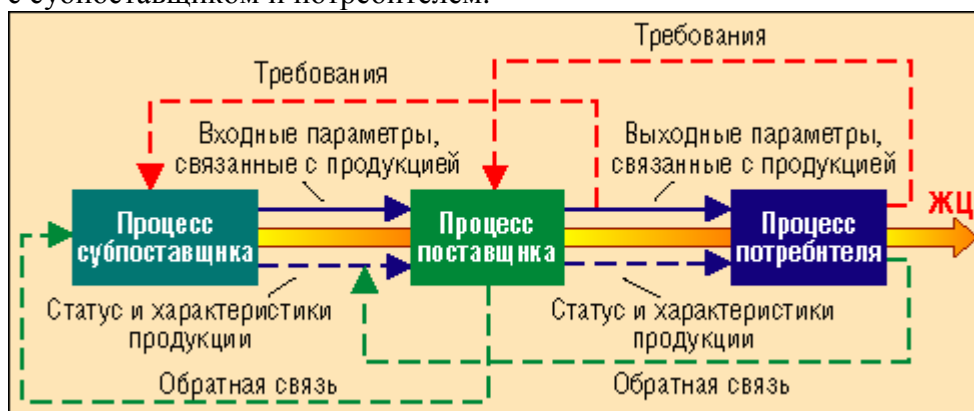
Информационные системы как средство автоматизации бизнес – процессов

Cals-технологии

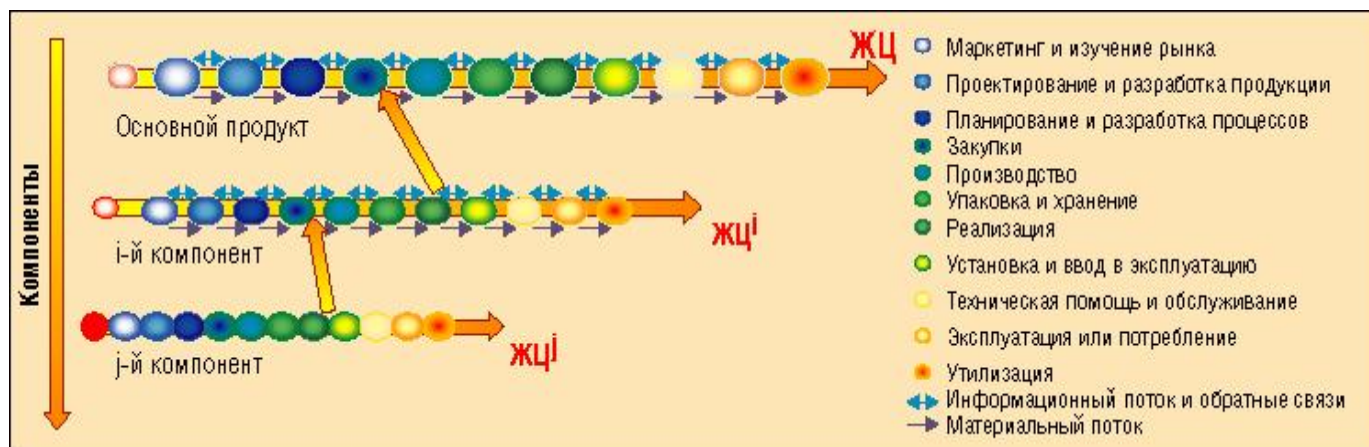
Информационная поддержка жизненного цикла продукта

Одним из направлений повышения эффективности промышленного сектора экономики является применение современных информационных технологий для обеспечения процессов, протекающих в ходе всего жизненного цикла продукции и ее компонентов. Жизненный цикл (ЖЦ) продукта, как его определяет стандарт ISO 9004-1, - это совокупность процессов, выполняемых от момента выявления потребностей общества в определенной продукции до удовлетворения этих потребностей и утилизации продукта.

Все многообразие этих процессов можно представить в виде прямых и обратных связей поставщика с субпоставщиком и потребителем.



В общем случае ЖЦ необходимо рассматривать как совокупность ЖЦ конечного продукта и ЖЦ входящих в него компонентов, результатов деятельности субпоставщиков. С этой точки зрения ЖЦ представляет собой древовидную структуру.



Информационное взаимодействие субъектов, участвующих в поддержке ЖЦ, должно осуществляться в едином информационном пространстве. В основе концепции единого информационного пространства лежит использование открытых архитектур, международных стандартов и апробированных коммерческих продуктов обмена данными. Стандартизации подлежат форматы представления данных, методы доступа к данным и их корректной интерпретации.

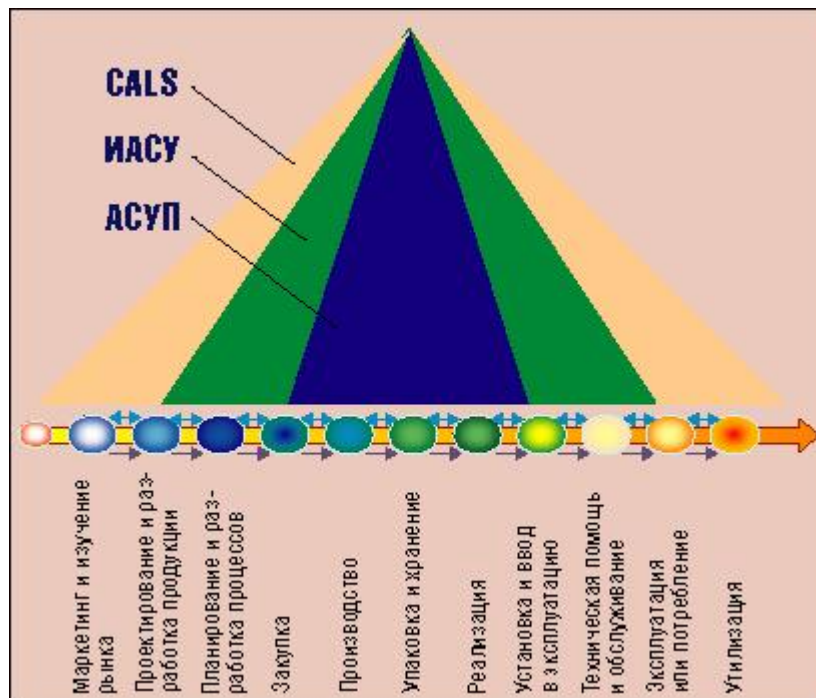
Первые шаги в организации единого информационного пространства были предприняты еще в 80-х годах в оборонном комплексе США. Возникла необходимость в обеспечении оперативного обмена данными между заказчиком, производителем и потребителем вооружений и военной техники (ВВТ), а также в повышении управляемости, сокращении бумажного документопотока и связанных с ним затрат.

Данная концепция изначально базировалась на понятия ЖЦ средств ВВТ и охватывала в основном фазы производства и эксплуатации. На первоначальном этапе инициатива получила обозначение CALS (Computer Aided Logistic Support - компьютерная поддержка поставок).

Доказав свою эффективность, концепция CALS начала активно применяться в промышленности, строительстве, транспорте и других отраслях экономики, расширяясь и охватывая все этапы ЖЦ продукта - от маркетинга до утилизации.

Новая концепция сохранила существующую аббревиатуру (CALS), но получила более широкую трактовку: Continuous Acquisition and Life cycle Support - непрерывная информационная поддержка жизненного цикла продукта. Таким образом, идея, возникшая в Министерстве обороны США и связанная только с поддержкой логистических систем, быстро превратилась в глобальную бизнес-стратегию перехода на безбумажную электронную технологию и повышения эффективности бизнес-процессов, выполняемых в ходе ЖЦ продукта за счет информационной интеграции и совместного использования информации на всех этапах ЖЦ. В настоящее время в мире действует более 25 национальных организаций, координирующих вопросы развития CALS-технологий, в том числе в США, Канаде, Японии, Великобритании, Германии, Швеции, Норвегии, Австралии, а также в рамках НАТО.

В отличие от интегрированной автоматизированной системы управления производством (ИАСУ), CALS-система охватывает все стадии ЖЦ.



Информационная интеграция на основе единой модели продукта

Предметом CALS являются технологии совместного использования информации (информационной интеграции) в процессах, выполняемых в ходе ЖЦ продукта. В основе CALS лежит комплекс единых информационных моделей, стандартизация способов доступа к информации и ее корректной интерпретации, обеспечение безопасности информации, а также юридические вопросы совместного использования информации (в том числе интеллектуальной собственности).

Информационная интеграция базируется на применении следующих интегрированных моделей:

- продукта;
- ЖЦ продукта и выполняемых в его ходе бизнес-процессов;
- производственной и эксплуатационной среды.

Классификация информационных моделей и их связь со стадиями ЖЦ продукта приведены в таблице.

С позиций системной архитектуры базовые информационные модели - это фундамент, на котором могут быть построены автоматизированные системы управления различного уровня. На основе одной и той же модели ЖЦ и бизнес-процессов решаются задачи анализа эффективности бизнес-процессов и обеспечения качества продукции. Интегрированная модель продукта обеспечивает обмен конструкторскими данными между проектировщиком и производителем, является источником информации для расчета потребности в материалах и создания электронных справочников по эксплуатации продукта и т. д.

Применение совместно используемых информационных моделей, являющихся единым источником информации и стандартизованных методов доступа к данным, - основа эффективной информационной кооперации всех участников ЖЦ.

Примеры задач, решаемых при помощи CALS-технологий

Моделирование жизненного цикла продукта и выполняемых бизнес-процессов. Первый шаг к повышению эффективности организационной структуры, поддерживающей одну или несколько стадий ЖЦ продукта, - моделирование и анализ ее функционирования.

Цель бизнес-анализа - выявить существующее взаимодействие между составными частями и оценить его рациональность и эффективность. Для этого с использованием CALS-технологий разрабатываются функциональные модели, содержащие детальное описание выполняемых процессов в их взаимосвязи. Формат описания регламентирован стандартами

IDEF/0 и ISO 10303 AP208. Полученная функциональная модель не только является детальным описанием выполняемых процессов, но также позволяет решать целый ряд задач, связанных с оптимизацией, оценкой и распределением затрат, оценкой функциональной производительности, загрузки и сбалансированности составных частей, т. е. вопросов анализа и реинжиниринга бизнес-процессов (Business Process Reengineering, BPR).

Проектирование и производство изделия. Совместное, кооперативное, проектирование и производство изделия может быть эффективным в случае, если оно базируется на основе единой информационной модели изделия.

Разрабатываемая на данной фазе конструкторско-технологическая информационная модель должна базироваться на использовании стандарта ISO 10303 STEP. Созданная однажды модель изделия используется многократно. В нее вносятся дополнения и изменения, она служит отправной точкой при модернизации изделия. Модель изделия в соответствии с этим стандартом включает: геометрические данные, информацию о конфигурации изделия, данные об изменениях, согласованиях и утверждениях.

Стандарт ISO 10303 построен таким образом, что помимо базовых элементов (интегрированных ресурсов) в его состав входят так называемые прикладные протоколы, определяющие конкретную структуру информационной модели для различных предметных областей (автомобилестроение, судостроение, строительство, электроника и т. д.). Все прикладные протоколы (прикладные информационные модели) базируются на стандартизованных интегрированных ресурсах. Таким образом, при создании нового прикладного протокола обеспечивается преемственность с уже существующими решениями.

Стандартный способ представления конструкторско-технологических данных позволяет решить проблему обмена информацией между различными подразделениями предприятия, а также участниками кооперации, оснащенными разнородными системами проектирования. Использование международных стандартов обеспечивает корректную интерпретацию хранимой информации, возможность оперативной передачи функций одного подрядчика другому, который, в свою очередь, может воспользоваться результатами уже проделанной работы. Это особенно важно для изделий с длительным ЖЦ, когда необходимо обеспечить преемственность информационной поддержки продукта, независимо от складывающейся рыночной или политической ситуации.

Эксплуатация изделия. Известно, что объемы разрабатываемой документации для сложного наукоемкого изделия очень велики. Поэтому традиционное бумажное документирование сложных изделий требует огромных затрат на поддержку архивов, корректировку документации, а также снижает эксплуатационную привлекательность и конкурентоспособность изделия.

Решение проблемы заключается в переводе эксплуатационной документации на изделие, поставляемой потребителю, в электронный вид. При этом комплект электронной эксплуатационной документации следует рассматривать как составную часть единой интегрированной информационной модели изделия.

Электронная документация может поставляться на электронных носителях, например компакт-дисках, или размещаться в глобальной сети Интернет.

Стадия ЖЦ	Информационная модель		
	продукта	ЖЦ продукта и выполняемых в его ходе бизнес-процессов	производственной и эксплуатационной среды
Маркетинг	Маркетинговая (концептуальная)	Модель процесса маркетинга продукта	Модель маркетинговой среды
Проектирование и разработка продукта	Конструкторская	Модель процессов проектирования, разработки продукта	Модель проектно-конструкторской среды
Производство или предоставление услуг, упаковка и хранение	Технологическая	Модель процессов производства	Модель технологической среды
Реализация	Сбытовая (цены, условия продажи и пр.)	Модель процессов продаж	Модель среды, в которой осуществляются продажи
Установка и ввод в эксплуатацию, техническая помощь и обслуживание, эксплуатация, утилизация	Эксплуатационная	Модель процессов эксплуатации	Модель эксплуатационной среды

Эксплуатационная документация может содержать информацию различных типов в соответствии со стандартами CALS: ISO 8879 (SGML), ISO 10744 (HyTime) и MIL-PRF-28001C - для текстовой и мультимедийной информации, MIL-PRF-28000A, MIL-PRF-28002C, MIL-PRF-28003A - для векторных и растровых графических иллюстраций.

Стандарты MIL-PRF-87268 и MIL-PRF-87269 определяют стиль, формат и технологию создания электронных справочников по изделиям. Стандартизация гарантирует применимость такой электронной документации на любых компьютерных платформах.

Важно отметить, что в электронный вид может быть преобразована эксплуатационная документация, созданная ранее без использования компьютерных систем. Для изделий, уже находящихся в эксплуатации длительный период и спроектированных традиционными методами, задача поддержки документации не менее актуальна. В качестве примера можно привести опыт проектов, выполняемых в ВМФ и ВВС США по массовому переводу миллионов страниц руководств и листов чертежей в стандартизованный электронный вид. Полученная электронная документация размещается в специальных хранилищах на базах ВМФ и ВВС или непосредственно у производителей и доступна через компьютерные сети. Одновременно информация может распространяться на компакт-дисках. Данные работы выполняются уже в течение ряда лет. При этом используются современные технологии сканирования, распознавания текста, векторизации чертежей и схем, создаются электронные справочники на целые изделия и отдельные системы.

Что дают CALS-технологии

Во многих развитых странах CALS рассматривается как стратегия выживания в рыночной среде, позволяющая:

- расширить области деятельности предприятий (рынки сбыта) за счет кооперации с другими предприятиями, обеспечиваемой стандартизацией представления информации на разных стадиях и этапах жизненного цикла. Благодаря современным телекоммуникациям, уже не принципиально географическое положение и государственная принадлежность партнеров. Новые возможности информационного взаимодействия позволяют строить кооперацию в форме виртуальных предприятий, действующих в течение ЖЦ продукта. Становится возможной кооперация не только на уровне готовых компонентов, но и на уровне отдельных этапов и задач: в процессах проектирования, производства и эксплуатации;
- повысить эффективность бизнес-процессов, выполняемых в течение ЖЦ продукта; за счет информационной интеграции и сокращения затрат на бумажный документооборот, повторного ввода и обработки информации обеспечить преемственность результатов работы в комплексных проектах и возможность

- изменения состава участников без потери уже достигнутых результатов;
- повысить “прозрачность” и управляемость бизнес-процессов путем их реинжиниринга, на основе интегрированных моделей ЖЦ и выполняемых бизнес-процессов, сократить затраты в бизнес-процессах за счет лучшей сбалансированности звеньев;
 - повысить привлекательность и конкурентоспособность изделий, спроектированных и произведенных в интегрированной среде с использованием современных компьютерных технологий и имеющих средства информационной поддержки на этапе эксплуатации;
 - обеспечить заданное качество продукции в интегрированной системе поддержки ЖЦ путем электронного документирования всех процессов и процедур.

По инициативе и при поддержке Сводного департамента экономики оборонных отраслей промышленности в сети Интернет создан информационный сервер по вопросам разработки и применения CALS-технологий в России (www.cals.ru), содержащий, помимо новостей, описаний продуктов и технологий, информацию о международных CALS-стандартах (STEP, SGML, HyTime, Plib, MANDATE).

Системы класса MRP - CSRP

В конце 60-х годов, в связи с бурным развитием вычислительной техники, ее возможности перестали быть востребованы только отдельными наукоёмкими отраслями, компьютерные системы прочно входили в повседневную деловую жизнь. Повсюду начались активные попытки оптимальной автоматизации и информатизации бизнеса, создавались новые концепции управления и совершенствовались уже существующие. Основными целями автоматизации производственных компаний являлись: точный расчет актуальной себестоимости продукции, ее анализ, понижение затрат в процессе производства и повышение производительности в целом, благодаря эффективному планированию производственных мощностей и ресурсов. Результатом оптимизации этих параметров являлись понижение конечной цены готовых изделий и повышение общей производительности, что соответственно немедленно отражалось на конкурентоспособности и рентабельности компании. В результате поиска решений в области автоматизации производственных систем родилась парадигма планирования потребностей в материалах (MRP). По сути, MRP-методология представляет собой алгоритм оптимального управления заказами на готовую продукцию, производством и запасами сырья и материалов, реализуемый с помощью компьютерной системы. Другими словами, MRP система позволяла оптимально загружать производственные мощности, и при этом закупать именно столько материалов и сырья, сколько необходимо для выполнения текущего плана заказов и именно столько, сколько возможно обработать за соответствующий цикл производства. Тем самым планирование текущей потребности в материалах позволяло разгрузить склады как и сырья и комплектующих (сырье и комплектующие закупались ровно в том объеме, который можно обработать за один производственный цикл и поступали прямо в производственные цеха), так и склады готовой продукции (производство шло в строгом соответствии с принятым планом заказов, и продукция, относящаяся к текущему заказу, должна быть произведена ровно к сроку его исполнения (отгрузки)). Собственно методология MRP является реализацией двух известных принципов JIT (Just In Time – Вовремя заказать) и KanBan (Вовремя произвести). Разумеется, идеальная реализация концепции MRP невыполнима в реальной жизни. Например, из-за возможности срыва сроков поставок по различным причинам и последующей остановки производства в результате этого. Поэтому в жизненных реализациях MRP-систем на каждый случай предусмотрен заранее определенный страховой запас сырья и комплектующих (safety stock), объем которого определяется компетентным руководством компании.

После появления концепции MRP, казалось бы, все основные проблемы производства были решены, активно создавались и продавались компьютерные программы, реализующие ее нехитрые принципы. Однако в процессе дальнейшего анализа существующей ситуации в мировом бизнесе и ее развития, выяснилось, что всю большую составляющую себестоимости

продукции занимают затраты напрямую не связанные с процессом и объемом производства. В связи с растущей от года к году конкуренцией, конечные потребители продукции становятся все более “избалованными”, ощутимо увеличиваются затраты на рекламу и маркетинг, уменьшается жизненный цикл изделий. Всё это требует пересмотра взглядов на планирование коммерческой деятельности. Отныне нужно не “что-то производить и стараться потом продать”, а “стараться производить, то, что продается”. Таким образом, маркетинг и планирование продаж должны быть непосредственно связаны с планированием производства. Исходя из этих предпосылок, и зародилась новая концепция корпоративного планирования. Концепция MRPII.

Состав систем класса MRPII (Manufacturing Resource Planning)

Очевидно, на любом производственном предприятии существует набор стандартных принципов планирования, контроля и управления функциональными элементами. Такими элементами являются производственные цеха, функциональные отделы, аппарат руководства и т.д. На основании этих принципов, попытаемся создать замкнутую логическую систему, которая позволяет отвечать на следующие тривиальные вопросы:

- Что собираемся производить?
- Что для этого нужно?
- Что имеем в данный момент?
- Что должны получить в итоге?

Эти вопросы должны иметь ясные ответы для руководящего состава любого коммерческого (производственного и непроизводственного) предприятия. Одной из основ эффективной деятельности любого предприятия является правильно поставленная система планирования. Она призвана содействовать ответам на эти вопросы.

Эта система планирования должна чётко отвечать на вопрос: "Что конкретно нужно в тот или иной момент времени в будущем?". Для этого она должна планировать потребности в материале, производственные мощности, финансовые потоки, складские помещения и т.д., принимая во внимание текущий план производства продукции (или услуг - здесь и далее) на предприятии. Такая система называется системой планирования ресурсов предприятия, или MRPII-системой (Manufacturing Resource Planning System. Окончание аббревиатуры - римская цифра "II" не несет никакого лексического смысла.)

Таким образом, MRPII-система должна состоять из следующих функциональных модулей:

- Планирование развития бизнеса (Составление и корректировка бизнес-плана)
- Планирование деятельности предприятия
- Планирование продаж
- Планирование потребностей в сырье и материалах
- Планирование производственных мощностей
- Планирование закупок
- Выполнение плана производственных мощностей
- Выполнение плана потребности в материалах
- Осуществление обратной связи

Схематический план работы MRPII-системы можно отобразить следующей диаграммой:



Модуль планирования развития бизнеса определяет миссию компании: её нишу на рынке, оценку и определение прибылей, финансовые ресурсы. Фактически, он утверждает, в условных финансовых единицах, что компания собирается произвести и продать, и оценивает, какое количество средств необходимо инвестировать в разработку и развитие продукта, чтобы выйти на планируемый уровень прибыли. Таким образом, выходным элементом этого модуля является бизнес-план.

Модуль планирования продаж оценивает (обычно в единицах готового изделия), какими должны быть объем и динамика продаж, чтобы был выполнен установленный бизнес-план. Изменения плана продаж, несомненно, влекут за собой изменения в результатах других модулей.

Модуль планирования производства утверждает план производства всех видов готовых изделий и их характеристики. Для каждого вида изделия в рамках выпускаемой линии продукции существует своя собственная программа производства. Таким образом, совокупность производственных программ для всех видов выпускаемых изделий, представляет собой производственный план предприятия в целом.

Модуль планирования потребности в материалах (или видах услуг) на основе производственной программы для каждого вида готового изделия определяет требуемое расписание закупки и/или внутреннего производства всех материалов комплектующих этого изделия, и, соответственно, их сборку.

Модуль планирования производственных мощностей преобразует план производства в конечные единицы загрузки рабочих мощностей (станков, рабочих, лабораторий и т.д.)

Модуль обратной связи позволяет обсуждать и решать возникающие проблемы с поставщиками комплектующих материалов, дилерами и партнерами. Тем самым, этот модуль собственно и реализует знаменитый "принцип замкнутой петли" (Closed loop principle) в системе. Обратная связь особенно необходима при изменении отдельных планов, оказавшихся невыполнимыми и подлежащих пересмотру.

Механизм работы МРП-системы

Составление производственного плана (Master Production Schedule) и общего плана деятельности (Production plan)

Логика работы МРП системы достаточно проста. Рассмотрим её на конкретном примере. Первым этапом является составления плана деятельности предприятия. Для этого, сначала определим производственную программу (Master Production Schedule-MPS) в виде следующего выражения: "Будем производить 30 автомобилей в неделю". Далее, при определении плана деятельности, принимаем во внимание следующие факторы:

- Текущий инвентарный запас изделий на складе
- Определение необходимого количества поддерживаемого инвентарного запаса на складе в тот или иной момент времени в течение всего периода планирования.
- Прогнозы продаж автомобилей на планируемый период

Следующая таблица, представляет собой типичный план деятельности предприятия:

Дата (конец месяца)		План продаж	План производства (MPS)	Объем запасов
31.03	По плану			60
	Реальный			
30.04	По плану	30	35	65
	Реальный	25	36	71
30.6	По плану	30	35	75
	Реальный			

Далее, показан типичный бизнес-план, который, по сути, является отображением плана деятельности, только в финансовом эквиваленте.

Дата (конец месяца)		План продаж	План производства (MPS)	Объём запасов
31.3	По плану			6000
	Реально			
30.4	По плану	3000	3500	6500
	Реально	2500	3600	7100
31.5	По плану	3000	3500	7000
	Реально	3800	3200	6500
30.6	По плану	3000	3500	7500

	Реально	3200	3700	7000
31.12	По плану	3000	3500	10500
	Реально			

Полный бизнес-план на производственном предприятии, разумеется, включает в себя затраты на новые разработки и развитие, а также ряд других затрат, напрямую не связанных с производством и продажами, но нам для начала достаточно рассмотреть его облегченный вариант. С точки зрения MRPII-системы, план деятельности и бизнес-план не являются независимыми, и, каждый раз, при обновлении плана деятельности, вносятся изменения и в бизнес-план. На основании главной программы производства ("Что собираемся производить?"), MRPII-система составляет инвентарный список (Bill of materials file) материалов-комплектующих ("Что для этого нужно?") и, сравнивая его с инвентарными запасами имеющимися в наличии (на складе или в позициях активных заказов - "Что имеем в данный момент?"), определяет потребность в материалах ("Что должны приобрести?").

Следующий список представляет собой пример инвентарного списка комплектующих для простого автомобильного двигателя:

Инвентарный номер	Наименование материалов-комплектующих	Кол-во
789887	Блок цилиндров	1
678767	Коленчатый вал	1
678776	Поршень в сборе	4
787987	Поршень	4
789877	Кольцо поршневое	4
.....
.....
567765	Свеча зажигания	4

Такой инвентарный список обычно называется списком с отступом. Это означает тот факт, что элементы списка высшего уровня (комплектующие высшего порядка) располагаются левее, чем их составляющие - комплектующие более низкого порядка. На основании инвентарных списков происходит планирование потребностей в материалах.

Планирование потребностей в материалах (MRP - Materials Requirements Planning)

Модуль планирования потребностей в материалах (MRP - Materials Requirements Planning) исторически является тем самым зерном, из которого выросла концепция MRPII (Manufacturing Resources Planning, Римская цифра "II" появилась на конце ввиду аналогичности аббревиатур с MRP). Цель этого модуля - так спланировать поставку всех комплектующих, чтобы исключить простои производства и минимизировать запасы на складе. Уменьшение запасов материалов-комплектующих, кроме очевидной разгрузки складов и уменьшения затрат на хранение дает ряд неоспоримых преимуществ, главное из которых - минимизация замороженных средств, вложенных в закупку материалов, не сразу идущих на конвейер, а подолгу ожидающих своей участи.

Входными элементами MRP-модуля являются:

- **Описание состояния материалов (Inventory Status File)**

Этот элемент является основным входным элементом MRP-модуля. В нем должна быть отражена максимально полная информация о всех типах сырья и материалах-комплектующих, необходимых для производства конечного продукта. В этом элементе должен быть указан статус каждого материала, определяющий, имеется ли он на руках, на складе, в текущих заказах или его заказ только планируется, а также описания, его запасов, расположения, цены, возможных задержек поставок, реквизитов поставщиков. Информация по всем вышеперечисленным позициям должна быть заложена отдельно по каждому материалу, участвующему в производственном процессе.

- **Программа производства (Master Production Schedule)**

Этот элемент представляет собой оптимизированный график распределения времени для производства необходимой партии готовой продукции за планируемый период или диапазон периодов.

- **Перечень составляющих конечного продукта (Bills of Material File)**

Этот элемент представляет собой список материалов и их количество, требуемое для производства конечного продукта. Таким образом, каждый конечный продукт имеет свой перечень составляющих. Кроме того, здесь содержится описание структуры конечного продукта, т.е. он содержит в себе полную информацию по последовательности его сборки. Чрезвычайно важно поддерживать точность всех записей в этом элементе и соответственно корректировать их всякий раз при внесении изменений в структуру и/или технологию производства конечного продукта.

Принцип работы MRP-модуля состоит в следующем:

1. Для каждого отрезка времени (обычно таким отрезком являются неделя или сутки) в течение всего периода планирования на основании инвентарных списков, плана производства и текущих запасов на складе создаётся полная потребность в материалах. Она представляет собой интегрированную таблицу, выражающую потребность в каждом материале, (суть элементе списка) в каждый конкретный момент времени.
2. Далее, вычисляется чистая потребность. Это делается путем вычитания из полной потребности тех материалов-комплектующих, которые имеются в текущих запасах или занесены, в качестве позиций, в активные заказы. Другими словами, чистая потребность определяет: какое количество материалов нужно заказать (или произвести, в случае внутреннего производства комплектующих) в каждый конкретный момент времени, чтобы удовлетворить текущие потребности производственного процесса. Очевидно, что чистая потребность тоже представляет собой определенную таблицу, элементы которой рассчитываются по формуле:
$$\text{Чистая потребность} = \text{Полная потребность} - \text{Инвентаризовано на руках} - \text{Страховой запас} - \text{Резервирование для других целей}$$
3. Последний этап работы заключается в том, что чистая потребность в материалах конвертируется в соответствующий план заказов на требуемые материалы и, в случае необходимости, вносятся поправки в уже действующие планы. При этом строго учитывается время выполнения каждого заказа, другими словами MRP-система, автоматически составляя план заказов, руководствуется известным временем выполнения каждого из них (lead time). Это время, как правило, определяется Поставщиком данного материала. Этот план заказов является руководящим документом отдела закупок.

Результатами работы MRP-модуля являются следующие основные элементы:

- **План Заказов (Planned Order Schedule)**

Этот элемент определяет, какое количество каждого материала должно быть заказано в каждый рассматриваемый период времени в течение срока планирования. План заказов является руководством для дальнейшей работы с поставщиками и, в частности, определяет производственную программу для внутреннего производства комплектующих, при наличии такового.

- **Изменения к плану заказов (Changes in planned orders)**

Этот элемент несёт в себе модификации к ранее спланированным заказам. Некоторые заказы могут быть отменены, изменены или задержаны, а также перенесены на другой период.

Планирование потребностей в производственных мощностях (CRP-Capacity Requirements Planning)

Для того чтобы производственная программа была осуществима, необходимо, чтобы имеющиеся в наличие производственные мощности смогли обработать то количество сырья и материалов-комплектующих, которое предписывает составленный MRP модулем план заказов, и изготовить из них готовые изделия. Собственно MRP-план является основным входным элементом модуля планирования потребностей в производственных мощностях (CRP-модуля). Другим немаловажным входным элементом является технологическая схема обработки/сборки конечного готового изделия (routing plan). Эта схема является определенной таблицей, аналогичной инвентарному списку, только с точки зрения этапов обработки и их длительности, а не комплектующих и их количества. На рисунке 5 представлена типичная технологическая схема обработки. Обычно, производственные мощности предприятия классифицируются на производственные единицы (work center). Такой производственной единицей может быть станок, инструмент, рабочий и т.д. Результатом работы CRP-модуля является план потребности в производственных мощностях (Capacity requirements plan). Этот план определяет, какое количество стандартных часов должна работать каждая производственная единица, чтобы обработать необходимое количество материалов.

Шаг	Номер производственной единицы	Название работы	Название производственной единицы	Кол-во рабочих часов
1	456676	Расточка	Токарный станок	1
2	56787, 345	Шлифовка		5
2.1	56787	Станочн. шлиф.	Шлифовальный станок	4
	345	Ручн Шлиф	Рабочий Петров Е. Н.	1

Также очень важно заметить, что модули MRPII-системы являются четко и однозначно взаимосвязанными (Lock step principle). Это в свою очередь означает собой тот факт, что в любом случае, если потребности в материалах (MRP-план, являющийся следствием изначально составленной программы производства (MPS)) не могут быть удовлетворены ни за счет внутреннего производства, ни за счет закупок на стороне, в план производства, очевидно, должны быть внесены изменения. Однако подобные явления должны быть исключениями. Одной из основных задач является составление успешного производственного плана с самого

начала.

Здесь представлен сокращенный вариант типичного плана потребности в производственных мощностях. Этот план является выходным элементом CRP-модуля.

План потребности в производственных мощностях. Производственная единица № 1500							
Номер материала	Номер заказа на пр-во	Кол-во	1.03.99	2.03.99	3.03.99	4.03.99	5.03.99
91234	12378	50		3.5			
80902	9870	500			16.5		
Суммарное количество часов			294	201	345	210	286

Таким образом, заметим еще раз: если в результате работы CRP-модуля установлено, что MRP-план неосуществим, то производственная программа(MPS) должна быть пересмотрена, более того, вероятно, необходимо пересмотреть весь план деятельности. Однако важно осознавать, что такой шаг должен быть сделан в самом крайнем случае, так как планировщик, работающий с CRP-системой должен быть компетентен и сам осознавать производственные возможности своего предприятия, понимая, что задача компьютера - лишь оптимально распределить загрузку производственных мощностей на период планирования. Тем самым, планировщик должен стараться определить и опротестовать заведомо неосуществимый MRP-план, до отправления его в CRP-систему, или найти пути для расширения производственных мощностей до необходимого уровня.

Контроль выполнения производственного плана. Контрольные отчёты по производительности и потреблению (input/output reports)

В тот момент, когда определено, что план потребностей в производственных мощностях может быть осуществлен, начинает функционировать контроль поддержания установленной производительности. Для этого в течение всего срока планирования системой регулярно создаются контрольные отчеты по производительности (Output control reports). Пример такого отчета приведен ниже.

Контрольный отчёт для производственной единицы №1500. Дата отчета - 23.05.1999,Пн				
Единица измерения – Стандартный час работы				
Статус/Дата	2.05.99	9.05.99	16.05.99	23.05.99
По плану	270	270	270	270
Реально	250	220	190	
Отклонение	-20	-70	-150	

Из вышеприведенного контрольного отчета становится видно, что отклонение реального темпа производства от производственного плана в первую неделю составляло 20 часов, во вторую-50 и в третью - 80 часов работы. Таким образом, суммарное отклонение достигло 150 стандартных часов.

Для адекватной работы системы необходимо определить величину допустимого

отклонения от плана производства. Например, если установлено, что величина допустимого отклонения на начало третьей недели равна половине планового недельного количества часов, то для примера на рисунке 7 это отклонение будет равняться 135 часам. И, в тот момент, когда величина реального отклонения превышает 135 часов, система сигнализирует о необходимости немедленного вмешательства в работу данной производительной единицы, и принятия мер к повышению ее производительности, вплоть её выхода на плановый уровень. Такими мерами может быть привлечение дополнительных рабочих, допустимое увеличение общего времени её работы и т.д.

Кроме контрольных отчетов производительности, для каждой производительной единицы существуют контрольные отчеты потребления материалов-комплектующих. Эти отчеты существуют для быстрого определения ситуаций, когда та или иная производительная единица не развивает плановой мощности из-за недостаточного снабжения материалами. Контрольный отчет потребления внешне абсолютно идентичен с отчетом, изображенным на рисунке 7, только вместо соотношения плановых и реальных часов работы, в нем отображается разница между реальным и плановым потреблением материалов рассматриваемой производственной единицей.

Списки операций (Dispatch lists)

Еще одним необходимым документом, регулярно (как правило, ежедневно) создаваемым MRP II-системой является список операций (operation lists). Списки операций обычно формируются в начале дня и передаются (или пересылаются) мастерам соответствующих производственных цехов. В этих документах отображена последовательность проведения рабочих операций над сырьем и комплектующими материалами на каждой производственной единице и их длительность. Списки операций позволяют каждому мастеру получать актуальную информацию, и фактически делают его частью MRP II-системы. На рисунке 8 изображен пример списка операций для одной из производственных единиц.

Список операций для производственной единицы № 1500 (Токарный станок), на 23.05.99				
Номер производственного Заказа	Инвентарный номер материала	Количество материала	Дата обработки по плану пр-ва	Количество часов обработки
17678	98769	50	20.05.99	3.5
16789	89769	500	23.05.99	19.2
18784	56307	1100	23.05.99	28.6
67830	78567	500	23.05.99	16.5
47890	87300	120	26.05.99	8.4
Суммарное количество часов				76.2

Как видно из таблицы, приведенный список определяет приоритет выполнения операций. Например, запоздавший по каким-то причинам производственный заказ от 20.05, был поставлен MRP II-системой в очередь первым. И наоборот, заказ от 26.05.99 имеет минимальный приоритет. Сразу стоит отметить, что список операций НЕ является суточным планом (это очевидно хотя бы из того, что суммарное количество часов превышает 24), а является лишь законом для мастера, определяющим последовательность и содержание производственных операций.

Обратная связь (feedback) и её роль в MRPII-системе

Чрезвычайно важно обратить внимание на функции обратной связи (feedback) в MRPII-системе. Например, если Поставщики не способны поставить материалы-комплектующие в оговоренные сроки, они должны послать отчет о задержках, сразу, как только они узнают о существовании этой проблемы. Обычно, стандартная компания имеет большое количество просроченных заказов с поставщиками. Но, как правило, даты этих заказов не отражают в достаточной степени дат реальной потребности в этих материалах. На предприятиях же, управляемых системами класса MRPII, даты поставки являются максимально близкими к времени реальной потребности в поставляемых материалах. Поэтому крайне важно заранее поставить систему в известность о возможных проблемах с заказами. В этом случае система должна сгенерировать новый план работы производственных мощностей, в соответствии с новым планом заказов. В ряде случаев, когда задержка заказов далеко не является исключением, в MRPII-системе задаётся объем минимального поддержания запасов "ненадежных" материалов на складе (safety stock).

В настоящее время, системы MRPII класса прочно входят в жизнь крупных и средних производственных организаций. Основной и эффективной чертой этих систем является возможность планировать потребности предприятия на короткие промежутки времени (недели и даже дни) и осуществлять обратную связь (например, автоматически изменять ранее построенные планы производства при сбоях поставок или поломке оборудования) внося в систему данные о проблемах в реальном времени.

Алгоритм работы MRPII-системы нацелен на внутреннее моделирование всей области деятельности предприятия. Его основная цель - учитывать и с помощью компьютера анализировать все внутрекоммерческие и внутрепроизводственные события: все те, что происходят в данный момент и все те, что запланированы на будущее. Как только в производстве допущен брак, как только изменена программа производства, как только в производстве утверждены новые технологические требования, MRPII-система мгновенно реагирует на произошедшее, указывает на проблемы, которые могут быть результатом этого и определяет, какие изменения надо внести в производственный план, чтобы избежать этих проблем или свести их к минимуму. Разумеется, далеко не всегда реально полностью устранить последствия того или иного сбоя в производственном процессе, однако MRPII-система информирует о них за максимально длительный промежуток времени, до момента их возникновения.

Таким образом, предвидя возможные проблемы заранее, и создавая руководству предприятия условия для предварительного их анализа, MRPII-система является надежным средством прогнозирования и оценки последствий внесения тех или иных изменений в производственный цикл.

Любая MRPII-система обладает определенным инструментарием для проведения планирования. Нижеперечисленные системные методологии - являются фундаментальными рычагами управления любой MRPII-системы:

- Методология расчёта и пересчета MRP и CRP планов.
- Принцип хранения данных о внутрипроизводственных и внутрекоммерческих событиях, которые необходимы для планирования.
- Методология описания рабочих и нерабочих дней для планирования ресурсов.
- Установление горизонта планирования (planning horizon)

Эти методологии и принципы не являются универсальными и определяются исходя из постановки конкретной задачи, применительно к конкретному коммерческому предприятию.

Эволюция стандартов планирования. От MRPII к ERP и CSRP.

Стандарты корпоративного планирования, как и любые стандарты со временем проходят через процесс эволюции. С годами в мире меняются принципы управления бизнесом и, соответственно, изменяются подходы к корпоративному планированию. В последнее время гиганты мировой индустрии распространили по всему миру сеть удаленных производственных и непроизводственных объектов управления, значительно усложнилась организационная

структура самих крупных компаний и холдингов. Это в свою очередь повлекло за собой увеличение управленческих издержек и затрат на поддержание сложных и запутанных логистических структур поставок продукции. Возникла необходимость искать методики, позволяющие оптимизировать решение и этих задач. В середине 90-х был введен в обращение термин ERP-системы. ERP-методология до настоящего времени должным образом не систематизирована, и представляет собой надстройку над MRPII, нацеленную на оптимизацию работы с удаленными объектами управления. В настоящее время, под широко используемым термином “ERP-система”, как правило подразумевается MRPII-система, с расширенными возможностями работы с сетью филиалов и зависимых компаний, расположенных по всему свету.

Для оптимизации управления логистическими цепочками была создана концепция SCM (Supply Chain Management), которую поддерживает большинство систем класса MRPII. SCM, положенная, как компонент общей бизнес стратегии компании, позволяет существенно снизить транспортные и операционные расходы, путем оптимального структурирования логистических схем поставок.

Одной из последних тенденций в бизнес-планировании, стало обращение усиленного внимания на качество обслуживания конечных потребителей продукции. Для того чтобы процветать, производители должны разрабатывать новые технологии и бизнес-процессы, которые позволяли бы им удовлетворять индивидуальные покупательские нужды и ожидания, отвечать на эти нужды товарами и услугами, которые представляют уникальную ценность для каждого покупателя. Производители должны совершить частичное изменение в стратегии и интегрировать покупателя в центр процесса планирования деятельности организации. Интеграция покупателя с ключевыми бизнес-процессами организации изменяет ее стратегию и реализацию этой стратегии, требует новую модель управления деятельностью: планирование ресурсов, синхронизированное с покупателем. Так появилась концепция CSRP (Customer Synchronized Resource Planning). Используя принцип CSRP, дистрибьютер продукции способен записать специфические требования к продукту, зафиксировать цену и автоматически послать эту информацию в головную организацию, где информация о требованиях к продукту динамически превращается в детальные инструкции по производству и планированию. Создается список материалов и комплектующих для производства, автоматически определяются производственные маршруты, материалы планируются и заказываются и, наконец, создается рабочий заказ. Критичная для покупателя информация динамически интегрируется в основную деятельность предприятия. После этого информация о критичных предпочтениях покупателя сохраняется в центральной базе данных о потребителях, которую могут использовать подразделения обслуживания покупателей, технического обслуживания, исследований, планирования производства и другие. Таким образом, деятельность предприятия синхронизируется с потребностями покупателей.

Технология CSRP



Архитектура открытых технологий поддерживает интеграцию прикладных систем

Покупатель использует браузер Интернет для доступа к Web-серверу производителя чтобы ввести заказ - стандартный или видоизмененный - в любое время дня или ночи. Покупатель может изменить предыдущие заказы, проверить состояние еще не выполненных заказов или запросить новые возможности. Потому что такое взаимодействие интегрировано в основные бизнес-системы предприятия, деятельность по планированию, производству и/или обслуживанию покупателей может автоматически изменяться действиями покупателя. И деятельность предприятия синхронизируется с покупателем.

Открытые технологии делают оба эти сценария и методологию CSRP реальностью. Как показано на рисунке F, для CSRP требуется использование открытых технологий, которые могут интегрировать стратегические приложения подразделений в масштабируемые, защищенные приложения масштаба предприятия. Успешное внедрение CSRP возможно только при использовании открытых технологий. Требуется переход от закрытых систем, включая системы ERP.

Организация имеет оптимизированную деятельность, интегрировала покупателя и внедрила архитектуру открытых технологий. Подразделения, ориентированные на покупателя, интегрированы в сердце системы планирования бизнесом.

Внедрение CSRP позволяет ответить на вопросы:

- Определить наиболее многообещающие и прибыльные рынки для компании.
- Установить какие рынки и товары наиболее прибыльны.
- Предсказать какие рынки будут наиболее прибыльными в течение одного года. В течение шести месяцев.
- Планировать и работать в направлении к более прибыльным рынкам.
- Гарантировать своевременную поставку наиболее ценным покупателям. Всем покупателям.
- Точно предсказать время поставки для уникальных заказов.
- Удовлетворить запросы покупателя в течение 24 часов. В течение 8 часов. В течение 1 часа.
- С прибылью видоизменять продукты и услуги.

Точно также как уменьшение числа дефектов становится возможным благодаря оптимизации процессов и сфокусированности на производственной деятельности (никто больше не удивляется бездефектным производством), также увеличение доли рынка и улучшение способности удерживать покупателя становится практичным и предсказуемым.

Планирование ресурсов, синхронизированное с покупателем (CSRP) предлагает модель бизнеса и набор инструментов, которые способны сделать партнерство с покупателем и достижимым, и поддерживаемым.

Если ERP была правилами игры для производителей прошедшего десятилетия, CSRP - это план игры на десятилетие нынешнее. Планирование ресурсов, синхронизированное с покупателем (CSRP), предлагает новый набор бизнес правил, которые делают возможным для производителя создать покупательскую ценность - разработать решения и услуги, которые сделают их (производителей) необходимыми для покупателей. Все больше и больше, конкурентные преимущества определяются как способность производителей удовлетворить уникальные потребности каждого уникального покупателя, каждого и каждый день.

Список литературы

1. Е.Судов. Информационная поддержка жизненного цикла продукта.//PC Week/RE № (169)45 от 17.11.1998
2. Г.Верников. Основы систем класса MRP-MRP II. www.cfin.ru/vernikov
3. Г.Верников. Стандарт MRP II. Структура и основные принципы работы систем поддерживающих этот стандарт. <http://www.cfin.ru/vernikov>
4. К.Де Роза Планирование ресурсов, синхронизированное с покупателем (CSRП). <http://www.socap.ru>
5. Стандарт MRP II. <http://www.cfin.ru>.
6. Питер Пин-Шен-Чен. Модель "Сущность-Связь" - шаг к единому представлению данных. //СУБД №3/1995
7. Шемакин Ю.И. Введение в информатику. - М.: Финансы и статистика, 1985
8. Информатика. Энциклопедический словарь для начинающих. Под ред. Д.А. Поспелова - М.: Педагогика-Пресс, 1994