

М.Л. Долженкова

О.В. Караваева

**Использование CASE-средств для
проектирования информационных систем**

Учебное пособие

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
КАФЕДРА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

М.Л. Долженкова
О.В. Караваева

Использование CASE-средств для проектирования информационных систем

Рекомендовано Ученым
советом ВятГУ в качестве
учебного пособия

Киров 2010

Печатается по решению редакционно–издательского совета
Вятского государственного университета

УДК 004.4¹22(07)

Д640

Долженкова М.Л. Использование CASE-средств для проектирования информационных систем: Учебное пособие / М.Л. Долженкова, О.В. Караваева - Киров: Изд-во ВятГУ, 2010.- 73 с.

В учебном пособии рассматриваются основы методологии функционального моделирования и построение моделей IDEF0, IDEF3 и DFD с помощью BPwin 4.1. Рассматриваются основы методологии информационного моделирования и построение моделей данных в нотации IDEF1X с помощью инструментального средства ERwin 4.1. Предлагается обзор интегрированной среды разработки. Приводятся примеры использования стандартных компонентов в приложении.

Пособие рассчитано на студентов, обучающихся по специальности 220100 «Вычислительные машины, комплексы, системы и сети» при изучении дисциплин "Технология программирования". Оно может быть полезным студентам других специальностей при знакомстве с методами структурного проектирования информационных систем.

Рецензент: кандидат педагогических наук, заведующая кафедрой Московского гуманитарно-экономического института Л.А. Страбыкина

© Вятский государственный университет, 2010

© М.Л. Долженкова, 2010

© О.В. Караваева, 2010

Оглавление

Введение	5
1. Постановка задачи.....	6
2. Анализ требований к системе.....	6
3. Функциональная декомпозиция системы	7
3.1 Построение модели AS-IS	12
3.1.1 Построение контекстной диаграммы	12
Установка цвета и шрифта объектов	14
Model Explorer	15
3.1.2. Детализация контекстной диаграммы.....	19
Рекомендации по рисованию диаграмм.....	19
3.1.3. Создание диаграммы декомпозиции в нотации DFD	30
3.1.4. Создание диаграммы декомпозиции в нотации IDEF3.....	32
3.2. Определение типов связей между функциями в модели	38
4. Построение ролевой диаграммы	39
5. Функционально-стоимостной анализ процессов (ABC)	44
6. Моделирование данных	51
6.1. Моделирование в ERwin.....	51
6.1.1. Физическая и логическая модели данных.....	52
6.1.2. Интерфейс ERwin. Уровни отображения модели.....	53
Установка цвета и шрифта	55
Подуровни логического уровня модели данных	57
Сущности.....	57
Атрибуты	58
Определение зависимостей между сущностями	61
Правила ссылочной целостности	65
Model Explorer	65
6.2. Построение логической модели данных	68
6.3. Построение физической модели данных	72

Выбор сервера	72
Таблицы	73
Колонки	74
6.4. Прямое и обратное проектирование.....	77
7. Связывание функциональной модели и модели данных	80
7.1. Присвоение импортированных сущностей потокам данных и использование данных активностями	81
7.2. Задание свойств, определяемых пользователем.....	84
Библиографический список.....	86

Введение

Создание программного обеспечения уже свыше трёх десятков лет является отраслью промышленности, значимость которой в мировой экономике и в экономиках отдельно взятых развитых стран возрастает с каждым годом. Программный продукт – особый вид продукта, требующий особого подхода в рассмотрении проблемы его промышленного производства. Когда речь идёт о таком виде программного продукта, как информационные системы, то очень сложно создать массовый продукт, который удовлетворял бы потребностям всех пользователей-организаций. Кроме того, подобные системы, будучи качественными и детально проработанными, не могут быть дешёвыми. Дороговизна продукта определяет существенный риск вложения финансовых средств в его создание. К этому следует добавить грандиозную скорость развития информационных технологий и постоянно эволюционирующие предметные области.

Вышесказанное можно представить в качестве одной проблемы – проблемы формальности при разработке информационных систем. Любая разумная активность предполагает наличие некоторого алгоритма действий. В число основополагающих свойств алгоритма входит его формальная определённость, которая гарантирует достижение однозначного результата при соблюдении всех требований и условий алгоритма, даже формальном. Но в большинстве случаев результат бывает неоднозначным. С одной стороны, с точки зрения разработчика система удовлетворяет всем требованиям и представляет собой именно то, что требовалось. С другой стороны, заказчику нужно совсем иное.

Потребность в формализованном процессе разработки привела к созданию ряда методологий, которые с различных точек зрения определяют весь цикл работы. Это своеобразные «алгоритмы», созданные для разработчиков информационных систем с целью перевести их работу в область формальной деятельности, которая гарантирует получение поставленного результата.

Ключевая проблема создания ИС – это их постоянно возрастающая сложность, обусловленная множеством решаемых задач, отсутствием близких аналогов, невозможностью быстрого и мгновенного перехода на новую систему, ограниченными интеллектуальными человеческими ресурсами, являющимися барьером на пути осознания всей системы целиком, и ещё множеством причин.

Специализированные программные комплексы CASE-средств, поддерживающие стандартную методологию проектирования, позволяют в той или иной степени преодолеть сложность и организовать управляемый процесс, который своевременно и адекватно адаптируется к изменяющейся предметной области, что является заявкой на поддержание полного жизненного цикла систем.

Стандарты в области проектирования предусматривают обязательное применение CASE-технологий, а их изучение позволяет освоить высокоуровневый абстрактный «инструмент» профессиональной деятельности разработчика-аналитика.

1. Постановка задачи

Необходимо разработать проект прикладной системы, поддерживающей основную деятельность организации по продаже программного обеспечения и автоматизирующей рабочее место менеджера по продажам. Разработка должна осуществляться в соответствии с основными стадиями жизненного цикла продукта с применением структурного подхода.

Система должна обеспечивать упорядоченное хранение данных, связанных с продажей программ, таких, как:

- сведения о самих программах, их поставщиках, покупателях;
- учёт наличия программ, приходящих от разных поставщиков;
- оформление договоров с поставщиками и покупателями;
- сбор статистики о продажах и обновление данных на сайте организации.

Кроме этого, система также должна предоставлять возможности простейшего статистического анализа.

2. Анализ требований к системе

Требования к системе позволяют отнести её к системам эксплуатационного уровня. Организация может быть разделена на уровни: стратегический, управленческий, уровень знания и эксплуатационный, который далее делится на функциональные области, такие, как продажи и маркетинга, производства, финансов, бухгалтерского учёта и учета человеческих ресурсов.

Различные организационные уровни обслуживают четыре главных типа информационных систем: системы с эксплуатационным уровнем, системы уровня знания, системы уровня управления и системы со стратегическим уровнем.

Системы эксплуатационного уровня обеспечивают элементарные действия организации типа продажи, платежей и т.д.

На этом уровне задачи, данные и действия предопределены и высоко формализованы.

3. Функциональная декомпозиция системы

Первоначально необходимо определить область моделирования. Описание области моделирования как системы в целом, так и ее компонентов является основой построения модели. Хотя предполагается, что в течение моделирования область может корректироваться, она должна быть в основном сформулирована изначально, поскольку именно область определяет направление моделирования и сроки окончания работы над моделью. При формулировании области необходимо учитывать два компонента - широту и глубину. Широта подразумевает определение границ модели: определяется, что будет рассматриваться внутри системы, а что снаружи. Глубина определяет, на каком уровне детализации модель является завершенной. При определении глубины системы необходимо не забывать об ограничениях времени - трудоемкость построения модели растет в геометрической прогрессии от глубины декомпозиции. После определения границ модели предполагается, что новые объекты не должны вноситься в моделируемую систему; поскольку все объекты модели взаимосвязаны, внесение нового объекта может быть не просто арифметической добавкой - оно может изменять существующие взаимосвязи. Внесение таких изменений в готовую модель является, как правило, очень трудоемким процессом.

Дальнейшее проведение анализа системы осуществляется в терминах бизнес процессов организации с использованием методологий IDEF0, IDEF3 и DFD.

Первый шаг в построении модели IDEF0 – это определение цели модели, то есть ряда вопросов, на которые призвана ответить модель. Эти вопросы следуют из формулировки задачи и анализа требований. Сама цель может быть сформулирована так: построить процессную модель функционирования

организации по продаже программного обеспечения (ПО). Из этой обобщенной формулировки следуют такие общие вопросы:

- из каких процессов состоит деятельность организации;
- какие основные цели организации;
- какие данные используются организацией;
- каковы общие алгоритмы их обработки.

Второй шаг – определение масштаба модели, то есть степени детальности или уровней декомпозиции диаграмм. Здесь применяются стандартные способы вынесения решения: следует ли дальше декомпозировать диаграмму.

Третий шаг – определение точки зрения на модель. Точку зрения можно представить как взгляд человека, который видит систему в нужном для моделирования аспекте. Точка зрения должна соответствовать цели моделирования. Часто при выборе точки зрения на модель важно задокументировать дополнительные альтернативные точки зрения. Так как с системой будет работать менеджер по продажам, то вполне целесообразно проводить рассмотрение с его точки зрения. Необходимо отметить, что точка зрения должна оставаться постоянной в течение всего процесса построения модели. Таким образом достигается согласованность и непротиворечивость.

Основу методологии IDEF0 составляет графический язык описания бизнес-процессов. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм (каждая диаграмма располагается на отдельном листе). Модель может содержать четыре типа диаграмм:

- 1) контекстная диаграмма (в каждой модели может быть только одна контекстная диаграмма);
- 2) диаграммы декомпозиции;
- 3) диаграммы дерева узлов;
- 4) диаграммы только для экспозиции (FEO).

Контекстная диаграмма является вершиной древовидной структуры диаграмм и представляет собой самое общее описание системы и ее взаимодействия с внешней средой.

После описания системы проводится функциональная декомпозиция (разбиение системы на крупные фрагменты). Диаграммы, описывающие каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции.

После декомпозиции контекстной диаграммы производится декомпозиция каждого большого фрагмента системы на более мелкие и так далее до достижения нужного уровня подробности описания. После каждого сеанса декомпозиции проводятся сеансы экспертизы - эксперты указывают на соответствие реальных бизнес-процессов созданным диаграммам.

Диаграмма дерева узлов показывает иерархическую зависимость работ, но не взаимосвязи между работами.

Диаграммы для экспозиции строятся для иллюстрации отдельных фрагментов модели, альтернативной точки зрения либо для специальных целей.

Работы (активности) обозначают поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Работы изображаются в виде прямоугольников. Все работы должны быть названы и определены. Имя работы должно быть выражено отглагольным существительным, обозначающим действие.

В процессе декомпозиции функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы, и называется дочерней. В свою очередь, функциональный блок-предок называется родительским блоком по отношению к дочерней диаграмме, а диаграмма, к которой он принадлежит, – родительской диаграммой. Каждая из подфункций дочерней диаграммы может быть далее детализирована путем аналогичной декомпозиции соответствующего ей функционального блока. Важно отметить, что в каждом случае декомпозиции функционального блока все интерфейсные дуги (стрелки), входящие в данный блок или исходящие из него, фиксируются на дочерней диаграмме. Этим достигается структурная целостность IDEF0–модели. Следует обратить внимание на взаимосвязь нумерации функциональных блоков и диаграмм - каждый блок имеет свой уникальный порядковый номер на диаграмме, а обозначение под правым углом указывает на номер дочерней для этого блока диаграммы.

Отсутствие этого обозначения говорит о том, что декомпозиции для данного блока не существует.

Взаимодействие работ с внешним миром и между собой описывается в виде интерфейсных дуг (стрелок). Стрелки представляют собой некую информацию и

именуются существительными. В IDEF0 различают четыре основные типы стрелок:

1. *Вход (Input)* - материал или информация, которые используются или преобразуется работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Зачастую сложно определить, являются ли данные входом или управлением. В этом случае подсказкой может служить то, перерабатываются/изменяются ли данные в работе или нет. Если изменяются, то скорее всего это вход, если нет, - управление.
2. *Управление (Control)* - правила, стратегии, процедуры или стандарты, которыми руководствуется работа. Управление влияет на работу, но не преобразуется работой.
3. *Выход (Output)* - материал или информация, которые производятся работой. Работа без результата не имеет смысла и не должна моделироваться.
4. *Механизм (Mechanism)* - ресурсы, которые выполняют работу, например персонал предприятия, станки, устройства и т.д. По усмотрению аналитика стрелки механизма могут не изображаться в модели.

Любой функциональный блок по требованиям стандарта должен иметь по крайней мере одну управляющую стрелку и одну исходящую, так как каждый процесс должен происходить по каким-то правилам (отображаемым управляющей стрелкой) и должен выдавать некоторый результат (выходящая стрелка), иначе его рассмотрение не имеет никакого смысла.

Часто бывают случаи, когда отдельные стрелки не имеет смысла продолжать рассматривать в дочерних диаграммах ниже какого-то определенного уровня в иерархии, или наоборот - отдельные стрелки не имеют практического смысла выше какого-то уровня. С другой стороны, бывает необходимо избавиться от отдельных “концептуальных” стрелок и не детализировать их глубже некоторого уровня. Для решения подобных задач в стандарте IDEF0 предусмотрено понятие туннелирования. Обозначение “туннеля” (Arrow Tunnel) в виде двух круглых скобок вокруг начала стрелки обозначает, что она не была унаследована от функционального родительского блока и появилась (из “туннеля”) только на этой диаграмме. В свою очередь, такое же обозначение вокруг конца стрелки непосредственно около блока-приёмника означает, что в дочерней по отношению к этому блоку диаграмме эта стрелка отображаться и

рассматриваться не будет. Чаще всего отдельные объекты и соответствующие им стрелки не рассматриваются на некоторых промежуточных уровнях иерархии.

Последним из понятий IDEF0 является словарь (Dictionary). Для каждого из элементов IDEF0: диаграмм, функциональных блоков, стрелок – существующий стандарт подразумевает создание и поддержание набора соответствующих определений, ключевых слов, повествовательных изложений и т.д., которые характеризуют объект, отображенный данным элементом. Этот набор называется словарем и является описанием сущности данного элемента. Словарь дополняет наглядный графический язык, снабжая диаграммы необходимой дополнительной информацией.

Обычно сначала строится модель существующей организации работы – AS-IS (как есть). Анализ функциональной модели позволяет определить:

- наиболее слабые места;
- преимущества новых бизнес-процессов;
- глубину изменений, которым подвергнется существующая структура организации бизнеса.

Признаками неэффективной работы деятельности могут быть:

- бесполезные, неуправляемые и дублирующиеся работы;
- неэффективный документооборот;
- отсутствие обратных связей по управлению;
- отсутствие обратных связей по входу.

Найденные в модели AS-IS недостатки можно исправить при создании модели TO-BE (как будет) - модели новой организации бизнес-процессов. Модель TO-BE нужна для анализа альтернативных путей выполнения работы и документирования того, как компания будет делать бизнес в будущем.

Технология проектирования ИС подразумевает сначала создание модели AS-IS, затем ее анализ и улучшение бизнес-процессов. И только на основе модели TO-BE строится модель данных, прототип и затем окончательный вариант ИС.

Иногда текущая AS-IS и будущая TO-BE модели различаются очень сильно, так что переход от начального состояния к конечному состоянию становится неочевидным. В этом случае необходима третья модель, описывающая процесс перехода от начального состояния системы к конечному.

3.1 Построение модели AS-IS

3.1.1 Построение контекстной диаграммы

При запуске *BPwin* в открывшемся диалоговом окне в поле *Name* вносится имя модели и тип (*Type*) – *IDEF0* (рис. 3.1). Автоматически создается контекстная

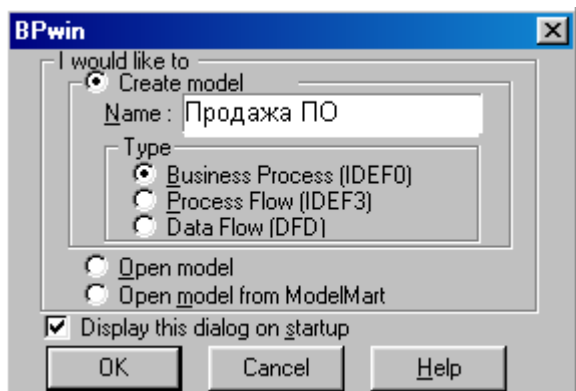


Рис. 3.1. Диалог *BPwin*

диаграмма. Процесс моделирования какой-либо системы в *IDEF0* начинается с определения контекста. В контекст входит определение субъекта моделирования, цели и точки зрения на модель. Для внесения цели и точки зрения в модели следует выбрать пункт меню *Editor/Model Definition*. В появившемся диалоговом окне (рис. 3.2) вносятся данные о модели.

1. *Project Name* - название проекта, которое будет показано в верхней части рамки диаграммы.
2. *Model Name* - название модели.
3. *Model Definition* - описание модели.
4. *Model Scope* - диапазон модели, содержащий информацию о том, что отражено в модели. Диапазон описывает ширину и глубину раскрытия процесса, описываемого моделью.
5. *Model Viewpoint* - точка зрения. Содержит информацию об эксперте, точка зрения которого рассматривается как основная при построении модели.
6. *Model Status* - статус модели.
7. *Model Time Frame* - вид модели.
8. *Purpose* - цель моделирования.
9. *Source* - источники информации, используемой при моделировании.
10. *Creation and Revision Dates* - дата создания и последнего изменения модели.
11. *Author Name and Initials* - фамилия и инициалы автора модели.

В меню *Model =>Model Properties* на вкладке *General* вводятся: имя модели “Продажа ПО”, имя проекта “Модель деятельности фирмы по продаже ПО”, имя автора и тип модели – *Time Frame: AS – IS (Как есть)*.

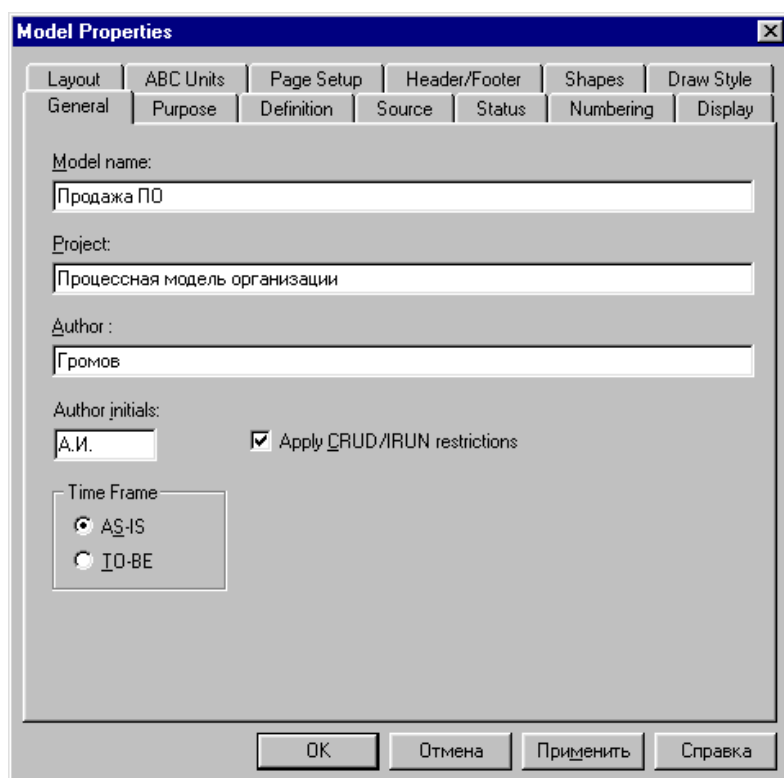


Рис. 3.2. Окно свойств модели











На вкладке *Purpose* вносится цель - “Purpose: Моделировать текущие (AS-IS) бизнес-процессы фирмы по продаже ПО” и точку зрения - “*Viewpoint*: Менеджер по продажам”.

На вкладке *Definition* вносится определение “Это учебная модель, описывающая деятельность фирмы по продаже ПО” и цель “*Scope*: Общее управление деятельностью фирмы: исследование рынка, поставка товара,

определение возможности продажи, хранение, сделка”. При запуске BPwin появляется основная панель инструментов, вид которой зависит от выбранной нотации (табл. 3.1).

Таблица 3.1

Основная панель инструментов.

Кнопка	Назначение	Соответствующий пункт меню
	Создать новую модель	File => New
	Открыть модель	File => Open
	Сохранить модель	File => Save
	Напечатать модель	File => Print
	Вызов генератора отчетов Report Builder	File => Report Builder
	Выбор масштаба	File => Zoom
	Масштабирование	File => Zoom
	Проверка правописания	Tools => Spelling
	Включение / выключение Model Explorer	View => Model Explorer
	Включение / выключение дополнительной панели инструментов работы с Model Mart	ModelMart

Основная деятельность организации заключается в продаже ПО. Таким образом, контекстная диаграмма содержит единственную работу (активность) «Продажи».

Под работами понимают процессы, функции или задачи. Работы обозначают в виде прямоугольников. При создании новой модели автоматический создается контекстная диаграмма с единственной работой, изображающей систему в целом. Двойной щелчок мыши на прямоугольнике вызывает диалоговое окно задания свойств работы.

Для определения свойств работы заполняются соответствующие поля:

- Activity Name - наименование работы (если не определено ранее)
- Definition - описание работы
- Source - источник информации о работе
- Status - статус работы (Working, Draft, Recommended, Publication, или Other)
- Author Name - имя автора (в поле автоматически вписываются данные из описания модели)

Установка цвета и шрифта объектов

Пункты контекстного меню *Font* и *Color* вызывают диалог *Activity Properties* или *Arrow Properties* для установки шрифта и цвета объекта. В нижней части вкладки Font данных диалогов (рис, 3.3) находятся:

- группа опций *Apply setting to*, позволяющих изменить шрифт для всех работ или стрелок на текущей диаграмме модели;
- группа *Global*, позволяющая изменить шрифт одновременно для всех объектов модели.

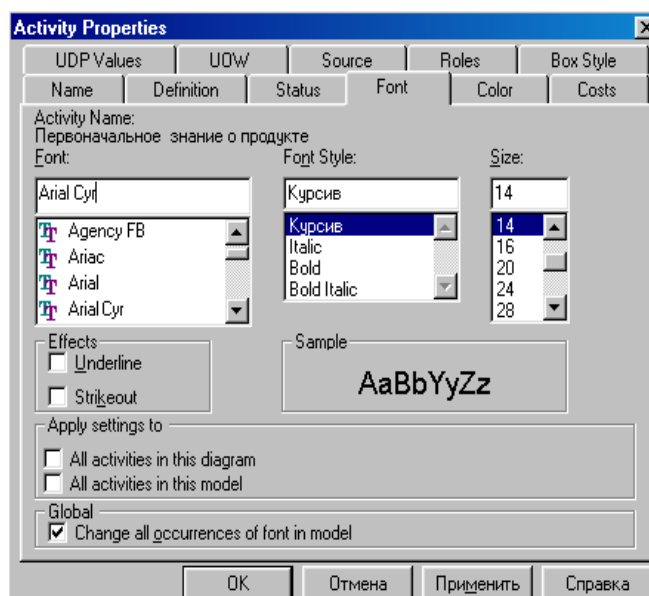



Рис. 3.3. Вкладка Font диалога *Activity Properties*

Кроме того, VPwin позволяет установить шрифт по умолчанию для объектов определенного типа на диаграммах и в отчетах. Для этого следует выбрать: *Model => Default Fonts*, после чего откроется следующее каскадное меню, каждый пункт которого служит для установки шрифтов для определенного типа объектов:

- *Context Activity* – работа на контекстной диаграмме;
- *Context Arrow* – стрелки на контекстной диаграмме;
- *Decomposition Activity* - работа на диаграмме декомпозиции;
- *Decomposition Arrow* - стрелки на диаграмме декомпозиции;
- *Node Tree Text* – текст на диаграмме дерева узлов;
- *Frame User Text* – текст, вносимый пользователем на каркасе диаграмм;
- *Frame System Text* – системный текст в каркасе диаграмм;
- *Text Blocks* – текстовые блоки;
- *Parent Diagram Text* – текст родительской диаграммы;
- *Parent Diagram Title Text* - текст заголовка родительской диаграммы;
- *Report Text* – текст отчетов.

Model Explorer

Для включения и выключения инструмента просмотра и навигации (*Model Explorer*) необходимо щелкнуть кнопку  на панели инструментов. Model Explorer (рис. 3.4) имеет три вкладки:

1. *Activities* – показывает в виде раскрывающегося иерархического списка все работы модели. Работы диаграмм *IDEF0* показываются зеленым цветом, *IDEF3* – желтым, *DFD* – голубым. Щелчок по работе в этой вкладке переключает левое окно VPwin на диаграмму, на которой эта работа размещена. Щелчок правой кнопкой по объекту позволяет редактировать его свойства.

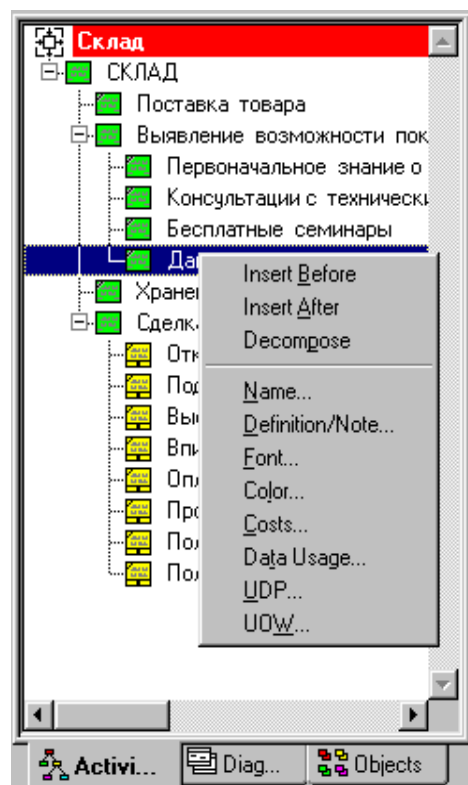


Рис. 3.4. Вкладка *Activities* навигатора Model Explorer

2. *Diagrams* – служит для перехода на любую диаграмму модели.
3. *Object* – после перехода в данную вкладку на ней отображаются все объекты, соответствующие выбранной во вкладке *Diagrams* диаграмме.

При щелчке правой кнопкой мыши по работе появляется контекстное меню (рис. 3.4), содержащее следующие пункты:


- *Insert Before* – создать новую работу на той же самой диаграмме. В списке работ новая работа будет вставлена перед текущей;
- *Insert After* - создать новую работу на той же самой диаграмме. В списке работ новая работа будет вставлена после текущей;
- *Decompose* – декомпозировать работу. В результате будет создана новая диаграмма декомпозиции;
- *Name* – вызов редактора имени работы;
- *Definition / Note* – вызов редактора определения и примечания к работе;
- *Font* – изменение шрифта работы;
- *Color* – изменение цвета работы;
- *Costs* – задание стоимости работы;
- *Data Usage* – ассоциация работы с данными;
- *UDP* – задание свойств, определяемых пользователем;
- *UOW* – задание свойств для работ IDEF3.

В соответствии с методом IDEF0 для любой активности необходимо определить входные данные, выходные данные, управление и механизм, которые изображаются на диаграмме стрелками:

- *Входные данные*: данные о потенциальных покупателях, данные о поставщиках, о программном обеспечении, маркетинговая информация;
- *Выходные данные*: результат работы организации – выручка, действительные покупатели, проданные программы;
- *Управление*: инструкции и нормативные документы;
- *Механизм*: персонал организации (офисные работники, бухгалтерия, отдел маркетинга, отдел продаж).

Стрелки на контекстной диаграмме служат для описания взаимодействия системы с окружающим миром. Они могут начинаться у границы диаграммы и заканчиваться у работы, и наоборот. Такие стрелки называются граничными.

Для внесения граничной стрелки входа нужно:

- щелкнуть по кнопке  в палитре инструментов и перенести курсор к левой стороне экрана, пока не появится начальная тёмная полоса;
- щелкнуть один раз по полоске (откуда выходит стрелка) и ещё раз в левой части работы со стороны входа (где

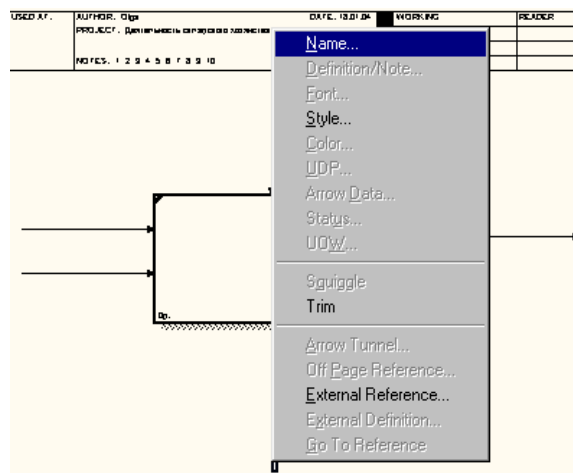



Рис. 3.5. Описание стрелки

- заканчивается стрелка);
- вернуться в палитру инструментов и выбрать опцию редактирования стрелки ;
- щелкнуть правой кнопкой мыши на линии

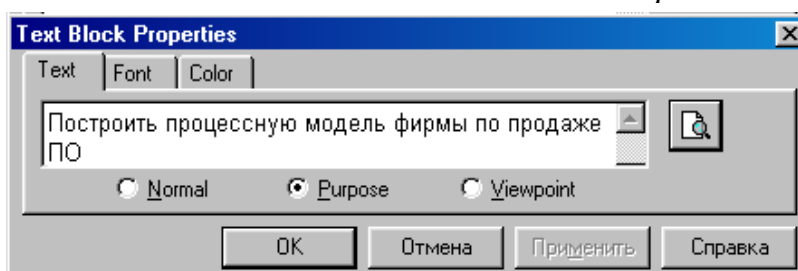


Рис. 3.6. Внесение текста в поле диаграммы с помощью Редактора Text Block Properties

стрелки и во всплывающем меню выбрать *Name* (рис. 3.5). В открывшемся диалоге *Arrow Properties* на вкладке *Name* добавить имя стрелки (*Arrow Name*).

На контекстной диаграмме создаются граничные стрелки, описанные в табл.3.2.

Таблица 3.2

Стрелки контекстной диаграммы



Рис. 3.7. Контекстная диаграмма

Имя стрелки (Arrow Name)	Определение стрелки (Arrow Definition)	Тип стрелки (Arrow Type)
Поставщики	Поставка товара, тех. поддержка и т. д.	Input
Потенциальные покупатели	Запросы информации, заказы и т. д.	Input
Программное обеспечение	Программное обеспечение, продаваемое фирмой.	Input
Данные маркетинговых исследований	Данные о программном обеспечении, пользующемся наибольшим спросом.	Input
Инструкции и нормативные документы	Правила продаж, правила оформления документации и т. д.	Control
ОП, бухгалтерия, офисный персонал	Работа с заказами, оформление счетов и т. д.	Mechanism
Маржа (прибыль)	Прибыль от сделок.	Output

С помощью кнопки **T** в поле диаграммы вносится текст: точка зрения и цель (рис. 3.6).

Таким образом, можно сказать, что контекстная диаграмма (рис. 3.7) построена (уровень A0). Это самый высокий уровень абстракции для данной задачи, выражающий точку зрения любого внешнего субъекта на организацию. Большинству внешних субъектов важен результат, которого можно достичь, обратившись в эту организацию, а работа организации является её внутренним устройством. То, как организация проявляет себя во внешней среде, для выражения точки зрения менеджера недостаточно. Менеджер должен видеть организацию в контексте процессов, протекающих в ней, а не внешнее явление организации.

3.1.2. Детализация контекстной диаграммы

Обычно IDEF0-модели несут в себе сложную и подробную информацию. Для того, чтобы упростить их и сделать более наглядными, в соответствующем стандарте приняты соответствующие ограничения сложности:

- количество функциональных блоков на диаграмме должно быть от трех до шести. Верхний предел (шесть) заставляет разработчика использовать иерархии при описании сложных предметов, а нижний предел (три) гарантирует, что на соответствующей диаграмме достаточно деталей, чтобы оправдать ее создание;
- количество подходящих к одному функциональному блоку (выходящих из одного функционального блока) не должно превышать четырех.

Рекомендации по рисованию диаграмм

В IDEF0 существует соглашения по рисованию диаграмм, которые призваны облегчить чтение и экспертизу модели. Некоторые из этих правил VPwin поддерживает автоматически, выполнение других следует обеспечить вручную:

- прямоугольники работ должны располагаться по диагонали с левого верхнего в правый нижний угол;
- следует максимально увеличивать расстояние между входящими или выходящими стрелками на одной грани работы;
- следует максимально увеличивать расстояние между работами, поворотами и пересечениями стрелок;

- если две стрелки проходят параллельно (начинаются из одной и той же грани и заканчиваются на одной и той же грани другой работы), то по возможности следует их объединить и назвать единым термином;

- обратные связи по входу рисуются "нижней" петлей, обратная связь по управлению - "верхней";

- циклические обратные связи следует рисовать только в случае крайней необходимости, когда подчеркивают значение повторно используемого объекта. Принято изображать такие связи на диаграмме декомпозиции;

- следует минимизировать число пересечений, петель и поворотов стрелок;

- если нужно изобразить связь по входу, необходимо избегать "нависания" работ друг над другом. В этом случае BPwin изображает связи по входу в виде петли, что затрудняет чтение диаграмм.

При декомпозиции контекстной диаграммы выделены четыре основные работы (активности), перечисленные в табл. 3.3.

Таблица 3.3

Работы диаграммы декомпозиции A0.

Имя работы (Activity Name)	Определение (Definition)
Выявление возможности продаж (A1)	Информирование заказчика, консультации, семинары
Получение ПО от поставщика (A2)	Поставка товаров на склад
Сделка (A3)	Заклучение сделки, оплата заказа, отгрузка товара
Послепродажное отслеживание (A4)	Дополнительный сервис, повторная продажа

Продажа осуществляется путём заключения сделки. Это центральная активность АЗ «Сделка». Сделку можно охарактеризовать как непосредственный процесс продажи, документальное оформление отношений. Сделка - сложный процесс в деятельности организации. Она охватывает работу как с потенциальными (клиент и организация сотрудничают в первый раз), так и с постоянными клиентами. В качестве входных данных для этой функции служат покупатели (одна из возможных сущностей предметной области), а точнее их идентификационные атрибуты и данные для контакта, программное обеспечение (тоже одна из возможных сущностей). Выход – проданное ПО (сущность) и маржа (разница между суммой выручки от продажи и суммой затрат на приобретение ПО у поставщика или определённый процент от суммы продажи, который организация оставляет себе).

Поскольку покупка ПО некоторой организацией – не что иное, как попытка внедрить информационную систему, то любая организация должна относиться к этому серьёзно, так как это связано со значительными финансовыми затратами и определяет успешность в конкуренции.

Учитывая данную потребность своих потенциальных покупателей, моделируемая организация, предваряя процесс продаж, проводит целый комплекс мер, связанных с выявлением действительных потребностей клиентов в отношении ПО, оценкой степени уверенности и определённости в том, что им нужно. Одну из главных ролей в этом процессе играют маркетинговые данные, анализируя которые, организация может точнее определить, к какому сегменту (группа потребителей, характеризующаяся более менее однородными потребностями) относится конкретный клиент, что позволит выделить круг ПО, наилучшим образом подходящий для него. Весь этот комплекс мер называется «Выявление возможности продаж». Конечно, это не выгодно моделируемой организации, но только в краткосрочном периоде (идёт снижение показателей продаж за текущий отрезок времени), но эти меры относятся к элементам дополнительного сервиса, которые создают хорошую репутацию организации и способствуют её укреплению, что приводит к выигрышу в долгосрочной перспективе.

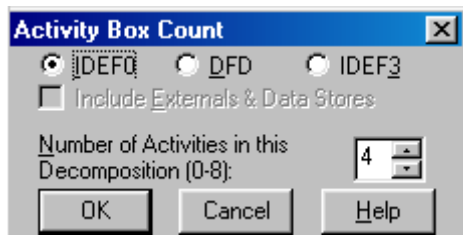


Рис. 3.8. Диалог Activity Box Count

Входными данными для функции А1 «Выявление возможности продаж» являются потенциальные покупатели, маркетинговые данные, программное обеспечение. Выходными данными являются данные для контакта (фактически это внесение клиента в список потенциальных клиентов) и рекомендуемое ПО

для данного потенциального клиента. Исполнителями функции являются отдел маркетинга и отдел продаж (кандидаты на ролевые группы). Исполнение функции осуществляется в соответствии с инструкциями и нормативными документами.

Функция А3 «Сделка» требует, чтобы после оплаты со стороны клиента, ему был передан экземпляр ПО на физическом носителе. У моделируемой организации имеется так называемый склад ПО (возможная сущность предметной области), где хранятся экземпляры ПО. Таким образом, выявляется ещё одно требование – перед заключением сделки необходимо, чтобы ПО было получено от поставщика. Это новая функция А2 «Получение ПО от поставщика». Её входными данными являются поставщики и ПО (их атрибуты), а выходными – полученное ПО (принятое на склад и зарегистрированное в базе данных). Выполнение данной функции обеспечивает отдел продаж и офисный персонал (возможные ролевые группы).

После завершения функции «Сделка» сотрудничество моделируемой организации и клиента, который приобрёл статус действительного, не прекращается. Начинается выполнение функции А4 «Послепродажное отслеживание».

Сущность этой функции состоит в оказании дополнительного сервиса, который важен как для клиента, так и для моделируемой организации. Кратко эту функцию можно охарактеризовать как процесс преобразования входных данных о заказчике в данные о повторных продажах. Повторные продажи – это обновление пакета клиента или приобретение нового пакета ПО. Процесс повторных продаж может инициироваться как самим клиентом, так и моделируемой организацией.

В частности, инициатива осуществления обновления, как правило, исходит от организации, в то время, как инициатива приобретения нового пакета исходит от клиента.

После определения всех активностей диаграммы декомпозиции можно приступить непосредственно к построению диаграммы. Для создания диаграммы декомпозиции необходимо перейти на нижний уровень, щелкнув кнопку ▼ в палитре инструментов или, находясь во вкладке *Activity* навигатора *Model Explorer*, выбрать пункт меню *Decompose*. Откроется диалог *Activity Box Count* (рис. 3.8), в котором нужно установить число работ на диаграмме нижнего уровня 4 и нажать *OK*. Автоматически будет создана диаграмма декомпозиции. Теперь нужно щелкнуть правой кнопкой мыши по работе, в контекстном меню выбрать *Name* (рис. 3.9) и внести имя работы.

Данную операцию необходимо повторить для всех четырех работ. Затем нужно внести определение, статус и источник для каждой работы согласно табл. 3.3.

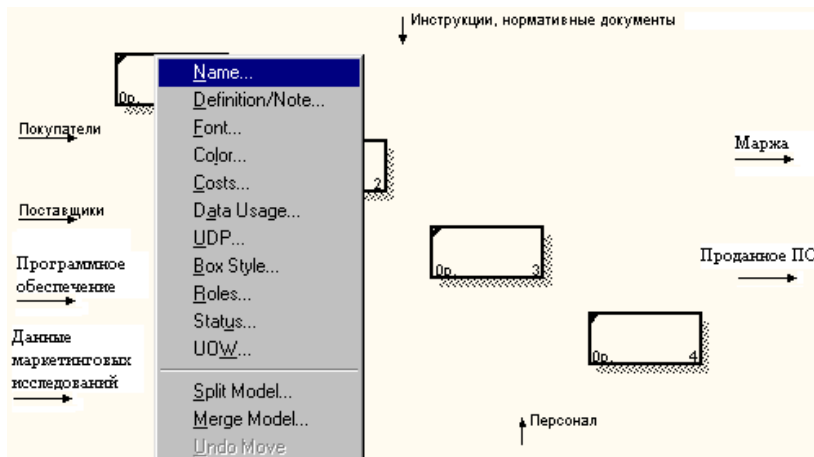



Рис. 3.9. Создание диаграммы декомпозиции

Для изменения свойств работ после их внесения в диаграмму можно воспользоваться словарем работ *Computer Associates BPwin – Activity Dictionary* (рис. 3.10). Для вызова словаря необходимо выполнить: меню *Dictionary*

=> *Activity*. Если описать имя и свойства работы в словаре, то работу можно будет внести в диаграмму позже с помощью кнопки □ в палитре инструментов. Невозможно удалить работу из словаря, если она используется на какой-либо диаграмме. При удалении работы из диаграммы из словаря она не удаляется. Для удаления всех имен работ, не используемых в модели, нужно щелкнуть по кнопке 🗑️ (*Purge*).

Иногда возникает необходимость удалить диаграмму декомпозиции. Для удаления всей диаграммы нужно воспользоваться пунктами меню *Diagram* => *Diagram Manager* (рис. 3.10).

При декомпозиции работы входящие в нее и исходящие из нее стрелки автоматически появляются на диаграмме декомпозиции, но при этом не касаются работ. Их необходимо связать с работами (несвязанные стрелки воспринимаются как синтаксическая ошибка). Для связывания граничных стрелок нужно перейти в режим рисования стрелок (кнопка  на палитре инструментов) и продлить стрелку до нужной работы.

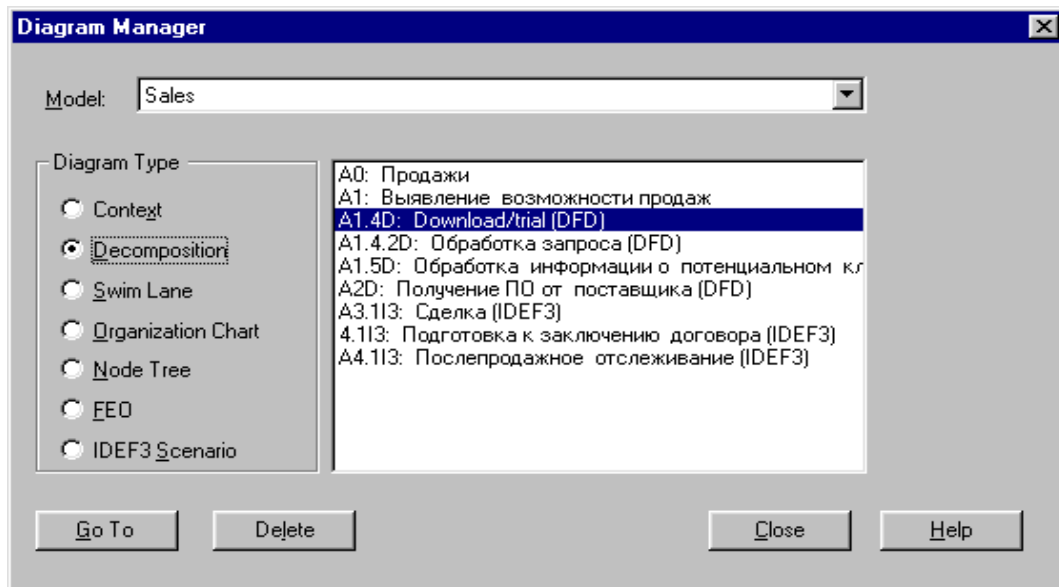


Рис. 3.10. Менеджер диаграмм


Стрелки, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются точно теми же самыми, что и дуги, входящие в диаграмму нижнего уровня и выходящие из нее, потому что блок и диаграмма представляют одну и ту же часть системы.


Для переименования стрелок нужно двойным щелчком на стрелке открыть окно *Arrow Properties* и на вкладке *Name* изменить имя стрелки (*Arrow Name*).

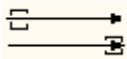
Альтернативный метод внесения имен и свойств стрелок – использование словаря стрелок (*Computer Associates BPwin – [Arrow Dictionary]*). Для вызова словаря необходимо выполнить: меню *Dictionary => Arrow*.

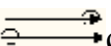
Если внести имя и свойства стрелки в словарь, то её можно будет внести в диаграмму позже. Невозможно удалить стрелку из словаря, если она используется на какой-либо диаграмме. При удалении стрелки из диаграммы из словаря она не удаляется. Имя и описание такой стрелки может быть использовано в дальнейшем.

Для добавления стрелки в словарь нужно перейти в конец списка и щелкнуть правой кнопкой по последней строке. Возникает новая строка, в которой нужно внести имя и свойства стрелки.

После связывания граничных стрелок, нужно создать внутренние стрелки и присвоить им имена. Имена стрелок нужно располагать так, чтобы их удобно было читать. Если необходимо, можно установить Squiggle (из контекстного меню, вызываемого при щелчке правой кнопкой по стрелке или кнопка  панели инструментов).

 Squiggle используется для того, чтобы было понятно, к какой стрелке относится данное имя.

Вновь внесенные граничные стрелки на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках  и автоматически не появляются на диаграмме верхнего уровня. Для перетаскивания их на диаграмму верхнего уровня нужно щелкнуть правой кнопкой мыши по квадратным скобкам и в контекстном меню выбрать пункт *Arrow Tunnel*. Откроется диалог *Border Arrow Editor* (рис. 3.11).

При выборе опции *Resolve it to border arrow* стрелка мигрирует на диаграмму верхнего уровня. Если же выбрать опцию *Change it to resolved rounded tunnel*, то стрелка будет затоннелирована и не попадет на другую диаграмму. Тоннельная стрелка изображается  с круглыми скобками на конце.

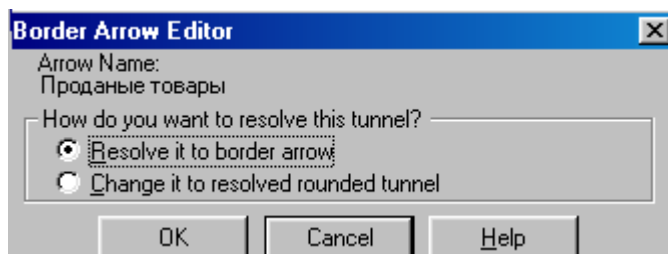


Рис. 3.11 Диалог *Border Arrow Editor*

Таким образом, первый этап детализации контекстной диаграммы завершён. Фактически, данная модель является одним из уровней детализации общей модели деятельности компании. Диаграмма имеет вид, представленный на рисунке 3.12. К данному моменту можно судить о значительном наполнении словаря предметной области, а именно раздела активностей, раздела сущностей.

Прежде чем закупать ПО, необходимо провести маркетинговое исследование: выявить круг потенциальных покупателей, определить их потребности. Поэтому функции А3 «Сделка» предшествует функция А1 «Выявление возможности продаж».

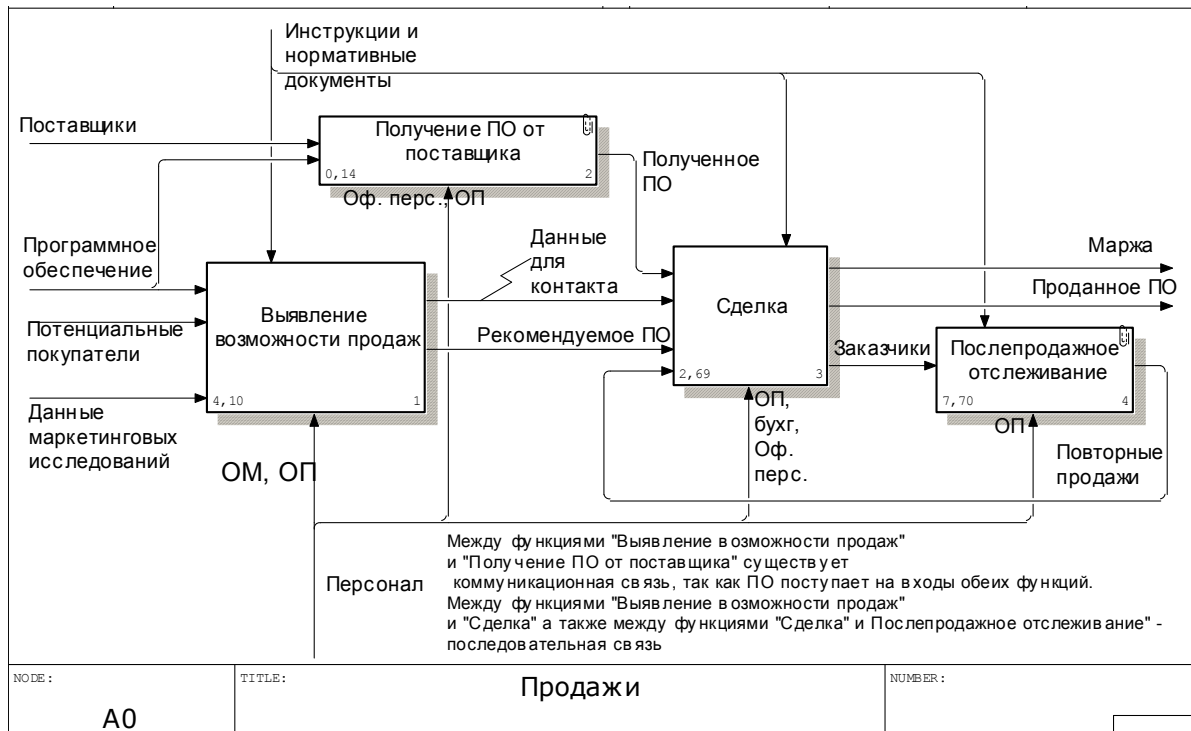


Рис. 3.12. Создание диаграммы декомпозиции A0

Возможность продаж определяется, исходя из сведений о клиенте. Представитель организации-клиента предоставляет сведения, которые касаются основных направлений деятельности предприятия, масштабов этой деятельности (размер организации и её подразделений), существующей системы обработки информации (технические и программные средства, находящиеся в распоряжении организации). Представитель организации-клиента так же высказывает предположения и предпочтения в отношении ПО. Это все составляет так называемое первоначальное знание о продукте. Задача моделируемой организации на этом этапе состоит в выявлении этого первоначального знания – функции A1.1 «Первоначальное знание о продукте». Поэтому входными данными здесь являются потенциальные покупатели. Выходные данные – это три группы потенциальных покупателей. Каждая группа формируется в соответствии с уровнем имеющихся представлений о системе ПО и выводами экспертов моделируемой организации о потребностях потенциального клиента.

Первая группа направляется на бесплатный семинар. Это те, кто не может выразить определённые мнения и запросы по поводу необходимого программного обеспечения и кто впервые обратился в моделируемую организацию.

Вторая группа направляются на консультацию с техническим специалистом. Это те заказчики, которые формулируют относительно определённые требования.

Третья группа снабжается демо-версиями ПО. Это те организации, которые имеют в своём штате опытных информационных менеджеров, разрабатывающих стратегии развития ИС. При наличии соответствующих средств им нет необходимости посещать моделируемую организацию, а достаточно загрузить демо-версии с сайта моделируемой организации.

Для обслуживания каждой из групп необходимо выполнить соответствующие функции:

- А1.2 «Бесплатный семинар»;
- А1.3 «Download/trial» - загрузка пробной версии;
- А1.4 «Консультация с техническим специалистом».

Входы всех этих функций представляют собой соответствующие запросы, в которых передаются первоначальные сведения об организации-клиенте. На выходах каждая их функций предоставляет уточнённые данные о клиентах и их потребностях.

Все полученные данные нуждаются в дальнейшей обработке и анализе. Эту работу выполняет функция А1.5 «Обработка информации о потенциальном клиенте». Её входы - это выходы предыдущих трёх функций А1.2 «Бесплатный семинар», А1.4 «Консультация с техническим специалистом» и А1.3 «Download/trial». Выходы функции А1.5 – данные для контакта и рекомендуемое ПО (рекомендации окончательные и определённые, используемые для выбора конкретного ПО для совершения сделки).

Для декомпозиции работы А1 “Выявление возможности продаж” можно воспользоваться данными из табл. 3.4 и табл. 3.5.

Таблица 3.4

Работы диаграммы декомпозиции A1.

Имя работы (Activity Name)	Определение (Definition)
Первоначальное знание о продукте (A1.1)	Информирование потенциальных покупателей о качестве и достоинствах товара
Бесплатные семинары (A1.2)	Ознакомление всех желающих с общими характеристиками товаров
Download/trial (A1.3)	Снабжение демо-версиями
Консультации с техническими специалистами (A1.4)	Предоставление возможности заинтересованным лицам подробнее познакомиться с товаром и его техническими характеристиками
«Обработка информации о потенциальном клиенте» (A1.5)	Получение данных о покупателе, его координат и другой информации

Таблица 3.5

Стрелки диаграммы декомпозиции A1.

Имя стрелки (Arrow Name)	Источник стрелки	Тип источника стрелки	Назначение стрелки	Тип стрелки
Потенциальные покупатели	Граница диаграммы	Input	Первоначальное знание о продукте	Input
Программное обеспечение	Граница диаграммы	Input	Download/trial	Input
Данные маркетинговых исследований	Граница диаграммы	Input	Консультации с техническими специалистами	Input
Данные маркетинговых исследований	Граница диаграммы	Input	Бесплатные семинары	Input
Инструкции, нормативные документы	Граница диаграммы	Control	Первоначальное знание о продукте	Control
Инструкции, нормативные документы	Граница диаграммы	Control	Консультации с техническими специалистами	Control
Инструкции, нормативные документы	Граница диаграммы	Control	Бесплатные семинары	Control
Инструкции, нормативные документы	Граница диаграммы	Control	Download/trial	Control

Окончание табл. 3.5

Имя стрелки (Arrow Name)	Источник стрелки	Тип источника стрелки	Назначение стрелки	Тип стрелки
Инструкции, нормативные документы	Граница диаграммы	Control	Обработка информации о потенциальном клиенте	Control
Запрос на консультацию	Первоначальное знание о продукте	Output	Консультации с техническими специалистами	Input
Заявки на участие в семинаре	Первоначальное знание о продукте	Output	Бесплатные семинары	Input
Запросы на демо- версии	Первоначальное знание о продукте	Output	Download/trial	Input
Информация от Downloads	Download/trial	Output	Обработка информации о потенц. клиенте	Input
Информация с консультаций	Консультации с техническими специалистами	Output	Обработка информации о потенц. клиенте	Input
Информация с семинаров	Бесплатные семинары	Output	Обработка информации о потенц. клиенте	Input
Отдел маркетинга	Персонал	Mechanism	Первоначальное знание о продукте	Mechanism
Учебный центр	Персонал	Mechanism	Бесплатные семинары	Mechanism
Технические специалисты	Персонал	Mechanism	Консультации с техническими специалистами	Mechanism
Отдел продаж	Персонал	Mechanism	Обработка информации о потенц. клиенте	Mechanism
Данные для контакта	Обработка информации о потенциальном клиенте	Output	Граница диаграммы	Output
Рекомендуемое ПО	Обработка информации о потенциальном клиенте	Output	Граница диаграммы	Output

Результат выполнения всех действий показан на рис. 3.13.

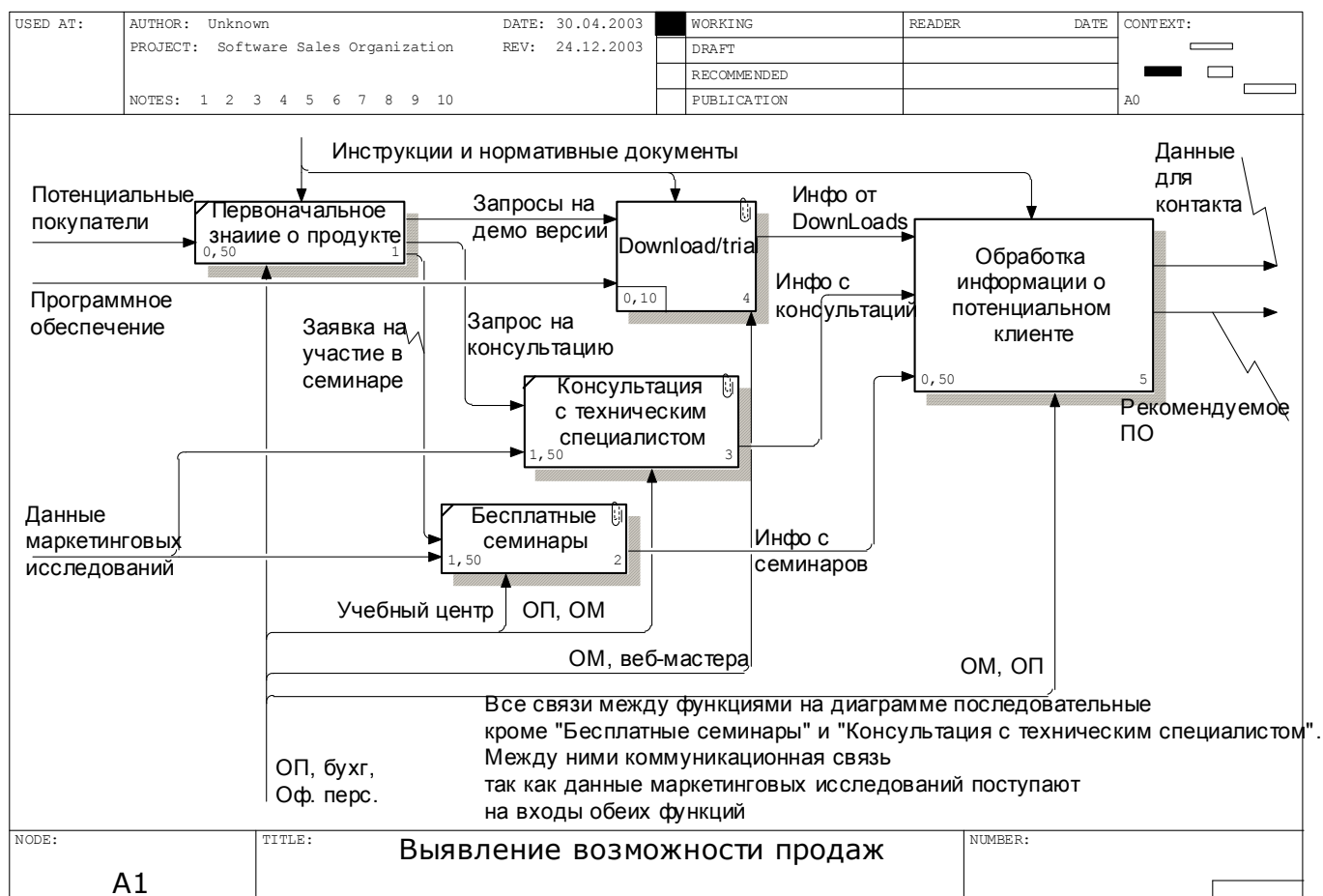


Рис. 3.13 – Результат построения диаграммы декомпозиции A1

3.1.3. Создание диаграммы декомпозиции в нотации DFD

Часто бывает необходимо не столько показать взаимосвязь между функциональными блоками, сколько отобразить движение информации внутри компании. В особенности, в специальном отображении нуждаются процессы, связанные с получением и обработкой внешней информации. Декомпозицию функции A1.5 «Обработка информации о потенциальном клиенте» целесообразно провести в нотации DFD, так как происходит регистрация или поиск клиента в базе данных. При оформлении заказа важно проверить, существует ли такой клиент в базе данных и, если нет, – внести его в базу данных и оформить заказ.

Примечание: В процессе декомпозиции, согласно правилам DFD, необходимо преобразовать граничные стрелки во внутренние, начинающиеся и заканчивающиеся на внешних ссылках.


Оформление заказа начинается со звонка или прихода клиента. В процессе оформления заказа база данных клиентов может просматриваться и редактироваться. Первоначально клиент может пожелать подробнее узнать о товаре и подать заявку на участие в семинаре или технической консультации.

Для декомпозиции функции A1.5 «Обработка информации о потенциальном клиенте» в диалоге *Activity Box Count* выбирается нотация DFD и количество работ, равное 3.

В новую диаграмму DFD “ Обработка информации о потенциальном клиенте ” вносятся имена работ:


- «Обработка заявок на участие в семинаре»;
- «Обработка заявок на участие в консультации»;
- «Внесение заказа на товар».

Хранилище данных позволяет описать данные, которые необходимо сохранить в памяти прежде, чем использовать в работах.

Используя кнопку  на палитре инструментов, необходимо внести хранилища данных:

- база данных “Клиенты”;
- база данных “Участники семинаров”;
- база данных “Участники консультаций”.

Граничные стрелки с диаграммы удаляются и добавляются внешние ссылки, которые являются источником или приемником данных извне модели.

При помощи кнопки  на палитре инструментов вносится внешняя ссылка «Звонок или явка клиента».

В отличие от стрелок IDEF0, которые представляют собой жесткие взаимосвязи, стрелки DFD показывают, как объекты (включая данные) двигаются от одной работы к другой.

Необходимо обратить внимание, что стрелки “Информация об участниках” и “Данные для контакта” двунаправленные. Для того чтобы сделать стрелку двунаправленной, нужно щелкнуть правой кнопкой мыши по стрелке и выбрать в контекстном меню пункт *Style*. Далее на вкладке *Style* выбрать опцию *Bidirectional*.

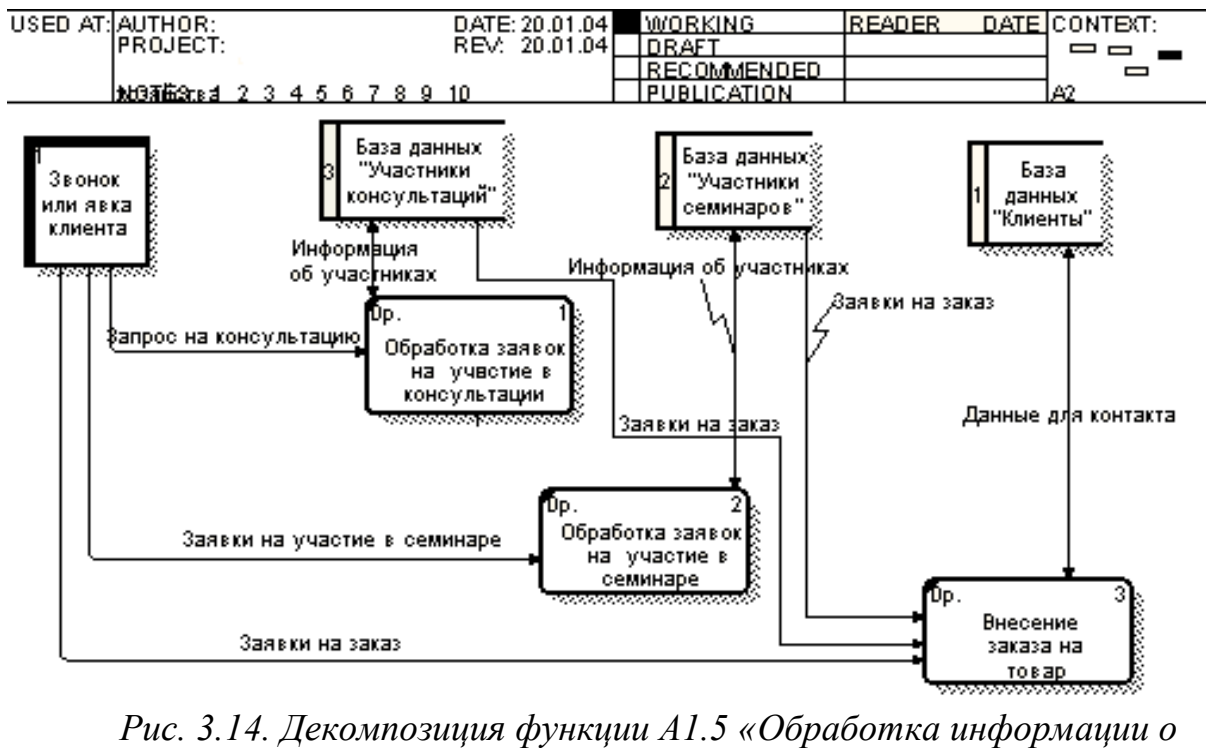



Рис. 3.14. Декомпозиция функции A1.5 «Обработка информации о потенциальном клиенте»

Таким образом, были рассмотрены ряд процессов и функций, которые обеспечивают основную деятельность моделируемой организации. Это один из множества вариантов построения функциональной модели предприятия. Для построения предшествующих диаграмм были использованы нотация IDEF0 и дополняющая ее нотация DFD (рис. 3.14). Эти нотации представляют модельную систему как сеть связанных между собой работ.

3.1.4. Создание диаграммы декомпозиции в нотации IDEF3.

Для полноты модели, а также для реализации дополнительных смысловых возможностей, связанных с элементами логики и организации ветвления, целесообразно использовать нотацию IDEF3. Диаграммы IDEF3 могут быть использованы в моделировании бизнес-процессов для анализа завершенности процедур обработки информации. С их помощью можно описывать сценарии действий сотрудников организации. Каждый сценарий сопровождается описанием процесса и может быть использован для документирования каждой функции.

Активность ”Сделка” будет декомпозирована в нотации IDEF3. Число работ в диалоге *Activity Box Count* – 7, нотация – *IDEF3*. При необходимости количество работ на диаграмме IDEF3 можно увеличить (кнопка ).

Каждая работа в IDEF3 описывает какой-либо сценарий бизнес-процесса и может являться составляющей другой работы. Единицы работы – Unit of Work (UOW) являются центральными компонентами модели, которые требуют более подробного описания, чем работы в IDEF0.

Для внесения свойств работы нужно щелкнуть по ней правой кнопкой мыши, выбрать в контекстном меню *Name* и внести имя работы (“Открытие сделки”). Затем на вкладке *Definition* внести определение “Обсуждаются все действия, которые могут иметь место в процессе сделки”. Каждая UOW должна иметь ассоциированный документ, который включает текстовое описание компонентов работы (табл. 3.6). Эта информация заносится во вкладку *UOW* диалога *Activity Properties*.

Таблица 3.6

Свойства UOW.

<i>Objects</i> (объекты, связанные с работой)	Действия: заключение договора о сделке, расторжение договора о сделке, оформление кредита, оформление рассрочки
<i>Facts</i> (факты, связанные с работой)	Продажа товаров может производиться как оптом, так и в розницу
<i>Constrains</i> (ограничения, накладываемые на работу)	Оформление кредита и рассрочки требует предоставления дополнительных документов

Функция АЗ «Сделка» реализуется следующими активностями:

- подготовка к заключению сделки;
- выписка счета / договора;
- проверка наличия товара на складе;
- получение платежа;
- получение товара на складе;
- выдача товара покупателю;
- проверка и закрытие сделки.

Окончание одной работы может служить сигналом к началу нескольких работ, или же одна работа для своего запуска может ожидать окончания нескольких работ. Перекрестки используются для отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы.


Для внесения перекрёстка служит кнопка  (добавить в диаграмму перекресток - *Junction*) на панели инструментов. В диалоге *Junction Type Editor* необходимо указать тип перекрестка. Смысл каждого типа перекрёстка приведен в табл. 3.7.

Таблица 3.7

Типы перекрестков.

Обозначение	Наименование	Смысл в случае слияния стрелок (Fan-in Junction)	Смысл в случае разветвления стрелок (Fan-out Junction)
	Асинхронное “И” (Asynchronous AND)	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Синхронное “И” (Synchronous AND)	Все предшествующие процессы должны быть завершены одновременно	Все следующие процессы запускаются одновременно
	Асинхронное “ИЛИ” (Asynchronous OR)	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
	Синхронное “ИЛИ” (Synchronous OR)	Один или несколько предшествующих процессов должны быть завершены одновременно	Один или несколько следующих процессов запускаются одновременно
	Исключающее “ИЛИ” XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

На диаграммах (рис. 3.15 и рис. 3.16) присутствуют так называемые узлы или точки ветвления, которые позволяют выделять различные пути выполнения процессов в зависимости от тех или иных условий. Точки ветвления бывают сводящими и разводящими. На диаграмме (рис. 3.15) присутствует две разводящие («асинхронное И» и «асинхронное ИЛИ») и две сводящие («асинхронное ИЛИ» и «асинхронное И») точки ветвления. На диаграмме (рис. 3.16) – две разводящие («синхронное ИЛИ») точки ветвления.

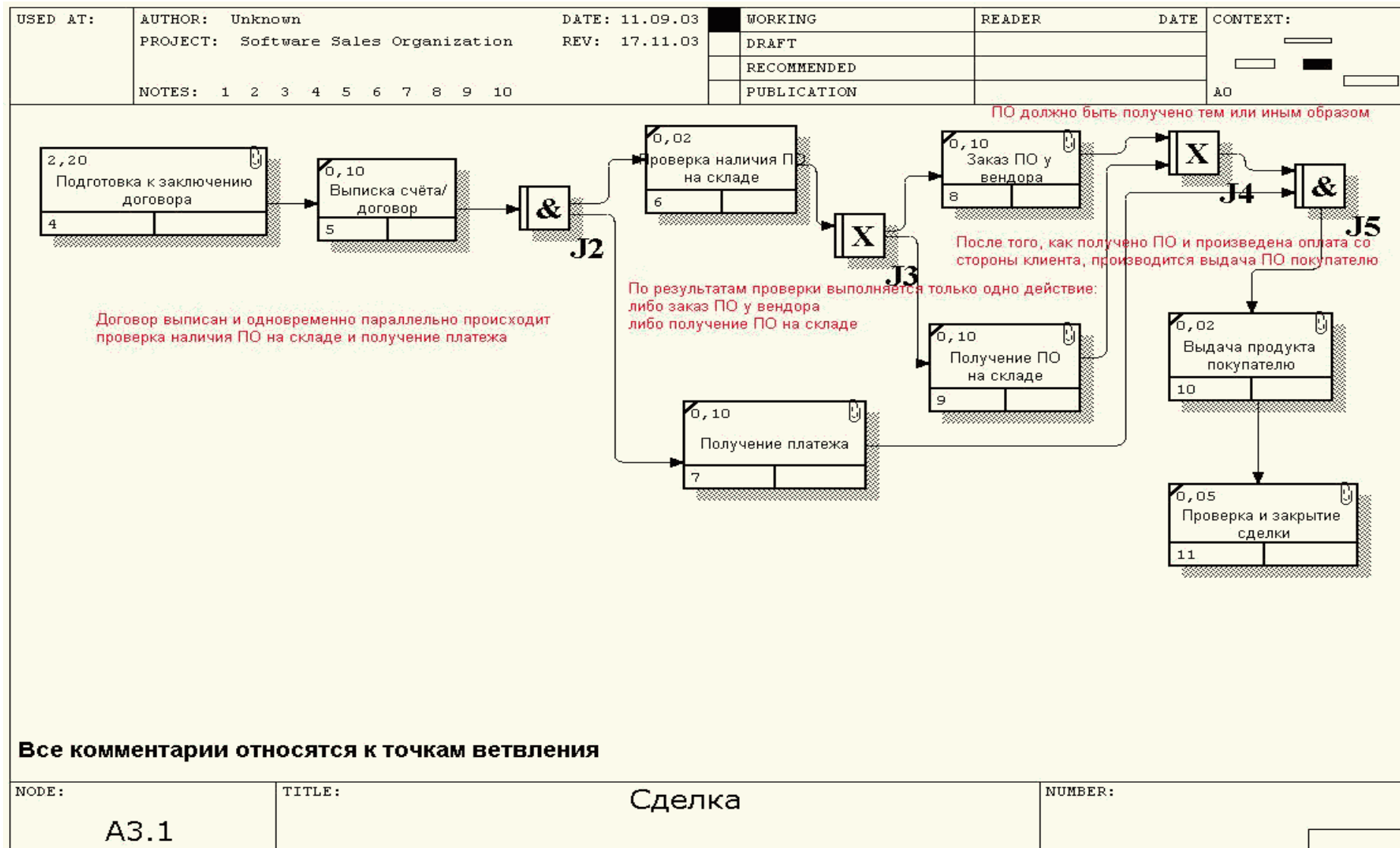


Рис. 3.15 - Декомпозиция функции АЗ «Сделка»

USED AT:	AUTHOR: Unknown	DATE: 26.09.02	WORKING	READER	DATE	CONTEXT:
	PROJECT: Software Sales Organization	REV: 20.11.03	DRAFT			<div style="border: 1px solid black; width: 20px; height: 10px;"></div>
			RECOMMENDED			<div style="border: 1px solid black; width: 20px; height: 10px;"></div>
			PUBLICATION			<div style="border: 1px solid black; width: 20px; height: 10px;"></div>
NOTES:	1 2 3 4 5 6 7 8 9 10					AO


```

graph TD
    Start([0,10]) --> J7((J7))
    J7 --> A1[0,50]
    J7 --> A2[3,00]
    A1 --> End1([0,50])
    A2 --> J8((J8))
    J8 --> A3[3,00]
    J8 --> A4[0,00]
    A3 --> End2([0,10])
    A4 --> End2
  
```

The flowchart illustrates the post-sales tracking process. It begins with a start node (0,10) labeled "Приём заявки от клиента". This leads to a decision point J7. From J7, the process can follow two paths: one leading to activity A1 (0,50) "Рассмотрение заявки" and another leading to activity A2 (3,00) "Непосредственная работа группы отслеживания". Activity A1 leads to end node 0,50. Activity A2 leads to decision point J8. From J8, the process can follow two paths: one leading to activity A3 (3,00) "Обновление пакета клиента" and another leading to activity A4 (0,00) "Ожидание очередного обновления/дополнения". Both A3 and A4 lead back to the start node (0,10).

Рис. 3.16 – Декомпозиция функции А4 «Послепродажное отслеживание»

3.2. Определение типов связей между функциями в модели

Типы связей для декомпозиции функции A0 «Сделка» представлены в табл. 3.8.

Таблица 3.8

Типы связей между функциями в декомпозиции функции A0 «Сделка»

Наименование функции	Тип связи
A1 «Выявление возможности продаж»	Коммуникационная (используются одни и те же входные данные)
A2 «Получение ПО от поставщика»	
A1 «Выявление возможности продаж»	Последовательная (выход от A2 является входом для A3)
A3 «Сделка»	
A3 «Сделка»	Последовательная
A4 «Послепродажное отслеживание»	

Таблица 3.9

Типы связей между функциями в декомпозиции функции A1 «Выявление возможности продаж»

Наименование функции	Тип связи
A1.2 «Бесплатные семинары»	Коммуникационная
A1.4 «Консультации с техническим специалистом»	
A1.1 «Первоначальное знание о продукте»	Последовательная
A1.3 «Download/trial»	
A1.3 «Download/trial»	Последовательная
A1.5 «Обработка информации о потенциальном клиенте»	

Во всех остальных диаграммах связи последовательные.

Необходимо подчеркнуть, что определение типов связей между функциями значимо для оценки качества модели. Следует избегать связности между функциями, имеющих слабо выраженную зависимость, таких, как случайная или логическая. Необходимо стремиться к функциональной связи между функциями, так как она может быть выражена через формулу, которая позволит производить необходимые вычисления и точно определять выходные параметры при данных входных. Но на практике при моделировании социально-экономических систем сложно выразить связи между элементами системы функционально. Альтернативой может служить последовательная связь, так как моделируются причинно-следственные зависимости между функциями. В создаваемой модели преобладают именно последовательные связи между функциями.

4. Построение ролевой диаграммы

Следующий шаг в процессе бизнес-моделирования – создание ролевой модели на основе модели структурной. Суть этого этапа состоит в том, чтобы для выделенных бизнес-процессов определить исполнителей, а также характер и круг их взаимодействия с другими сотрудниками и подразделениями, вид источников дополнительной информации. Ролевая модель является необходимым дополнением к модели процессной, так же, как структурная модель, она обязательно должна дополнять и дополняться структурной моделью. Фактически ролевая модель представляет собой структурную модель в динамике, то есть система взаимоотношений между сотрудниками по ходу осуществления бизнес-процесса.

Организационная структура оказывает существенное влияние на определение бизнес-процессов и их выполнение. Без достаточного понимания организационной структуры, ролей, отношений и ответственностей часто невозможно смоделировать бизнес-процессы.

Построение организационной диаграммы в BPWin осуществляется в автоматическом режиме. Но для этого нужно указать соответствующие данные, а точнее - их определить. К таким данным относятся ролевые группы, роли и ресурсы. Ролевая группа – набор ролей, выполняющих ту или иную часть бизнес-процессов. Роль – конкретная обязанность или должность в рамках ролевой группы. Ресурсы – конкретная личность, выполняющая данную обязанность (в некоторых случаях это могут быть и технические средства).

Ролевые группы можно рассматривать на разных уровнях. На высоком уровне под ролевой группой можно понимать целую организацию. На более низком уровне ролевой группой обычно считается подразделение в организации. Роли тоже можно рассматривать на разных уровнях. Так, на высоком уровне ролью может быть подразделение в рамках организации, а на более низком уровне – конкретная должность в рамках подразделения или организации.

В словаре ролевых групп и ролей указываются выявленные в процессе анализа предметной области ролевые группы и роли, причем роль может входить в несколько групп.

Словарь ролевых групп вызывается из меню *Dictionary/Role Group*, словарь ролей – из меню *Dictionary/Role*. Эти словари (рис. 4.1 и рис. 4.2) заполняются данными в соответствии с табл. 4.1 и табл. 4.2

Computer Associates BPwin - [Role Group Dictionary]			
Dictionary Edit View Help			
Name	Definition	Bitmap	Importance
Web группа	Персонал, ответственный за разработку Web-сайта, организации, его обслуживание, публикацию данных о ПО в наличии		Medium
Бухгалтерия	Обслуживание ФХД		Low
Организация по продаже ПО	Целая организация, которая моделируется		High
Отдел маркетинга	Взаимодействие с исполнительным менеджментом по вопросам выработки рыночных целей, стратегии, планов, их реализации, изучение рынка, отслеживание		Medium
Отдел продаж	Работа с покупателями, детальное выяснение потребностей		Low
Офисный персонал	Оформление документов, приём заказов, высылка уведомлений...		Low
Управление и менеджмент	Директор, главный менеджер. Определяют долгосрочный план развития (стратегию), разрабатывают среднесрочную программу. Наличие прямых связей		High
Учебный центр	Проведение семинаров, обучающих курсов, консультаций с продавцами....		Low
			Low

Рис. 4.1 – Словарь ролевых групп

Computer Associates BPwin - [Role Dictionary]				
Dictionary Edit View Help				
Name	Definition	Role Group	Bitmap	Importance
Web группа	Сотрудники, ответственные за разработку и поддержку	Организация по продаже ПО		Low
Администратор Web сайта	Администрирование работы сайта, своевременная пч	Web группа		Medium
Бухгалтер	Сотрудники, обеспечивающие ФХД	Бухгалтерия		Low
Бухгалтерия	Обработка документов, отражающих финансово-хозяй	Организация по продаже ПО		Low
Главный бухгалтер	Обработка документов, отражающих финансово-хозяй	Бухгалтерия		Low
Директор	Глава организации, официальный представитель во в	Организация по продаже ПО		High
Маркетолог	Деятельность отдела маркетинга	Организация по продаже ПО		Medium
Менеджер по продажам	По поручению директора. Планирует, организует, мот	Организация по продаже ПО		High
Организация по продаже ПО		Организация по продаже ПО		Low
Отдел маркетинга	Анализ маркетинговых данных	Организация по продаже ПО		Low
Отдел продаж	Сотрудники, ответственные за проведение процесса	Организация по продаже ПО		Low
Офисный работник	Офисная работа	Организация по продаже ПО		Low
Продавец	Обслуживание покупателей, рекомендации, выяснени	Организация по продаже ПО		Low
Разработчик Web сайта	Участие в разработке Web сайта, обрабатывает ста	Web группа		Medium
Учебный центр	Сотрудники, проводящие бесплатные семинары и общ	Организация по продаже ПО		Low
				Low

Рис. 4.2 – Словарь ролей

Таблица 4.1

Ролевые группы

Name	Definition	Importance
Web группа	Персонал, ответственный за разработку Web-сайта, его обслуживание, публикацию данных о ПО в наличии	Medium
Бухгалтерия	Обслуживание финансово-хозяйственной деятельности	Low
Организация по продаже ПО	Организация, которая моделируется	High
Управление и менеджмент	Директор, главный менеджер (по продажам). Определяют долгосрочный план развития (стратегию), разрабатывают среднесрочную программу. Наличие прямых связей с отделом маркетинга	High

Окончание табл. 4.1

Name	Definition	Importance
Отдел маркетинга	Взаимодействие с управлением и менеджментом по вопросам выработки рыночных целей, стратегии, планов, их реализации, изучение рынка, отслеживание тенденций, действий конкурентов, динамики спроса. Реализация краткосрочных планов	Medium
Отдел продаж	Работа с покупателями, детальное выяснение потребностей	Low
Отдел логистики	Работа с поставщиками и покупателями. Выяснение потребностей, заказ и приобретение необходимого ПО	Medium
Офисный персонал	Оформление документов, приём заказов, высылка уведомлений...	Low
Учебный центр	Проведение семинаров, обучающих курсов, консультаций с продавцами....	Low

Таблица 4.2

Роли

Name	Definition	Role Group	Importance
Web группа	Персонал, ответственный за разработку Web-сайта, организации, его обслуживание, публикацию данных о ПО в наличии	Организация по продаже ПО	Low
Администратор Web сайта	Администрирование работы сайта, своевременная публикация данных, отслеживает потребности посетителей сайта.	Web группа Организация по продаже ПО	Medium
Бухгалтер	Финансово-хозяйственный работник	Бухгалтерия	Low
Бухгалтерия	Сотрудники, обеспечивающие ФХД	Организация по продаже ПО	Low
Директор	Глава организации, официальный представитель во внешней среде. Заключает договоры с поставщиками ПО	Организация по продаже ПО. Управление и менеджмент	High
Маркетолог	Сотрудник отдела маркетинга	Организация по продаже ПО Отдел маркетинга	Medium

Окончание табл. 4.2

Name	Definition	Importance	Name
Менеджер по продажам	Сотрудник менеджмента. Планирует, организует, мотивирует, контролирует	Организация по продаже ПО Управление и менеджмент	High
Организация по продаже ПО	Моделируемая организация	Организация по продаже ПО	High
Отдел маркетинга	Анализ маркетинговых данных	Организация по продаже ПО	Medium
Отдел продаж	Проведение процесса продаж, включая демонстрацию	Организация по продаже ПО	Medium
Офисный работник	Офисная работа	Организация по продаже ПО	Low
Продавец	Обслуживание покупателей, рекомендации, выяснение потребностей. Специализируется на ПО	Организация по продаже ПО	Medium
Разработчик Web сайта	Участвует в разработке Web сайта, обрабатывает статистику по сайту	Web группа Организация по продаже ПО	Medium
Учебный центр	Сотрудники, проводящие бесплатные семинары и обучение персонала организации	Организация по продаже ПО	Low

Для создания диаграммы Swim Lane следует выбрать меню Diagram/Add Swim Lane diagram. В появившемся окне нужно внести название и имя автора диаграммы, выбрать диаграмму IDEF3, на основе которой будет построена ролевая диаграмма, и группу ролей, из которой выбираются роли, связанные с диаграммой (рис. 4.3). В следующем окне выбираются роли, на основе которых будет создана диаграмма. Количество полос на диаграмме будет соответствовать указанному в колонке Display Swin Line (рис. 4.3).

На диаграмме Swim Lane все объекты располагаются произвольно. Необходимо вручную перетащить объекты на полосы в соответствии с ролями (рис. 4.4).

Swim Lane Diagram Wizard - Step 1 of 2

You must base a Swim Lane diagram on either a Role Group or Single-select Text List UDP that you have created.

On what would you like to base the Swim Lane diagram?

☒ Role Group
 ☐ Text List UDP

Сделка

You can create a Swim Lane diagram that copies an existing Process Flow (IDEF3) diagram, or you can create an empty Swim Lane diagram with no diagram objects.

Do you want to select a source Process Flow diagram?

☒ Yes
 ☐ No

A3.113: Сделка (IDEF3)

Diagram Name:

Swim Lane Diagram Wizard - Step 2 of 2

Swim Lane diagram Name:

Сделка

Select Swim Lanes to display in diagram:

Display Swim Lane	Display Bitmap	Swim Lanes
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Бухгалтерия
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Отдел логистики
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Отдел продаж

Рис. 4.3. Два шага построения ролевой диаграммы

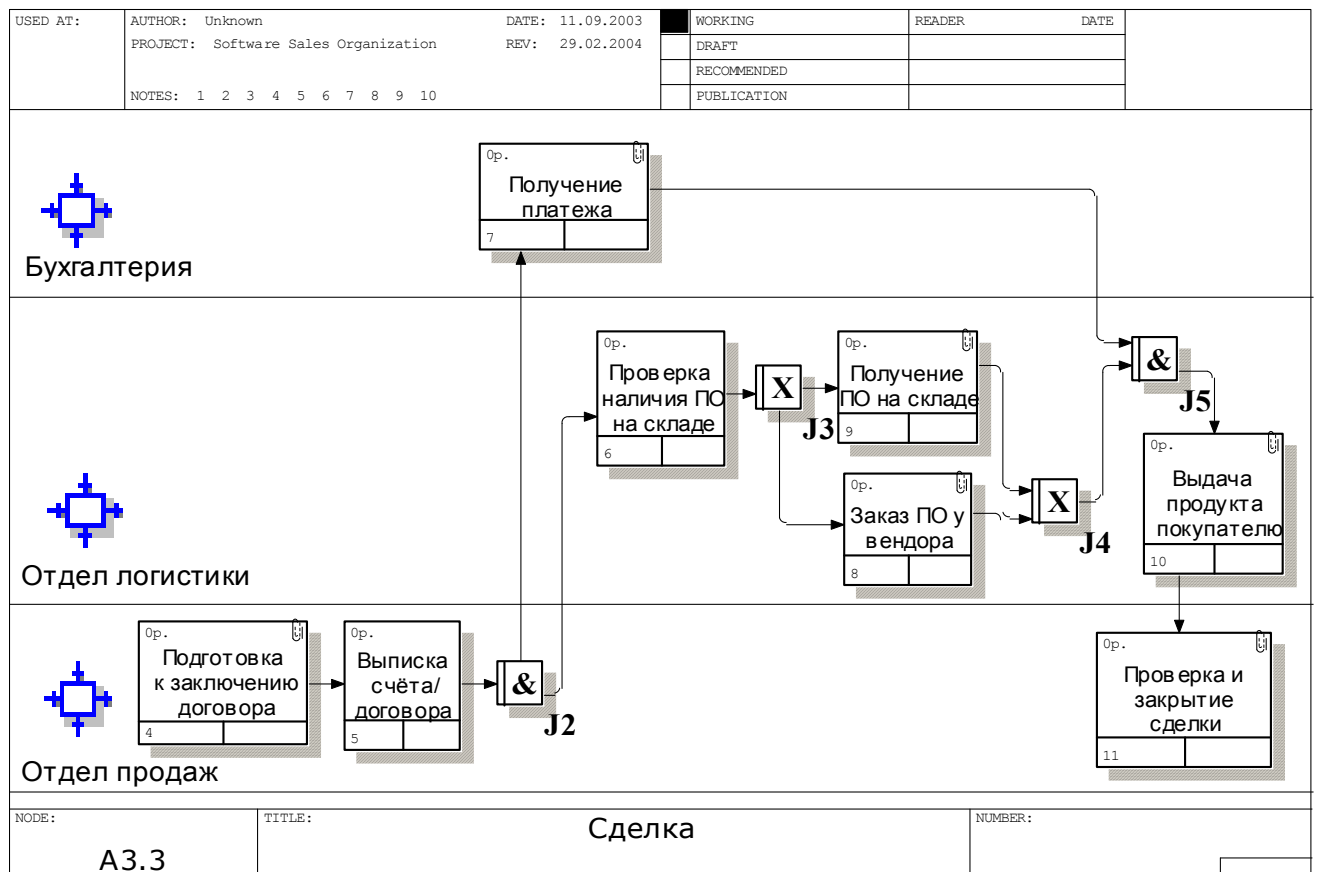


Рис. 4.4. Ролевая диаграмма

5. Функционально-стоимостной анализ процессов (АВС)

Методика АВС предназначена для обнаружения и анализа затрат, связанных с осуществлением функций. АВС-анализ допускается проводить либо в контексте стоимостных (денежных) затрат, либо в контексте затрат времени. Первый подход позволяет оценить общую сумму затрат, связанных с деятельностью организации. Второй подход позволяет оценить общую длительность всех функций, составляющих контекстную функцию.

Основные направления использования АВС-модели для реорганизации бизнес-процессов - это повышение производительности, снижение стоимости, трудоемкости, времени и повышение качества.

Можно выделить три этапа повышения производительности.

1. Анализ функций для определения возможностей повышения эффективности их выполнения.
2. Выявление причины непроизводительных расходов и пути их устранения.
3. Отслеживание и ускорение нужных изменений с помощью измерения основных параметров производительности.

С помощью АВС-метода можно реорганизовать деятельность предприятия, чтобы достигнуть снижения стоимости, трудоемкости и времени. Для этого необходимо сделать следующее:

- сократить время, необходимое для выполнения функций;
- устранить ненужные функции;
- сформировать ранжированный перечень функций по стоимости, трудоемкости или времени;
- выбрать функции с низкой стоимостью, трудоемкостью и временем;
- организовать совместное использование всех возможных функций;
- перераспределить ресурсы, высвободившиеся в результате усовершенствований.

Повышение качества бизнес-процессов осуществляется за счет проведения сравнительной оценки и выбора рациональных (по стоимостному или временному критерию) технологий выполнения операций или процедур.

В основе управления, основанного на функциях, лежат несколько аналитических методов, использующих ABC-информацию. Это - стратегический анализ, стоимостной анализ, временной анализ, анализ трудоемкости, определение целевой стоимости и исчисление стоимости, исходя из жизненного цикла продукта или услуги.

Одним из направлений использования принципов, средств и методов ABC является планирование бюджета, основанное на функциях. Планирование бюджета использует ABC-модель для определения объема работ и потребности в ресурсах. Можно выделить два пути использования:

- выбор приоритетных направлений деятельности, увязанных со стратегическими целями;
- разработка реалистичного бюджета.



Рис. 5.1. Концептуальная схема ABC-метода.

ABC-информация позволяет принимать осознанные и целенаправленные решения о распределении ресурсов, опирающиеся на понимание взаимосвязей функций и стоимостных объектов, стоимостных факторов и объема работ.

При построении функционально-стоимостных моделей существует методологическая и технологическая взаимосвязь между IDEF0- и ABC-моделями.

Связанность методов IDEF0 и ABC заключается в том, что оба метода рассматривают предприятие как множество последовательно выполняемых функций, а стрелки входов, выходов, управления и механизмов IDEF0-модели соответствуют стоимостным объектам и ресурсам ABC-модели. На рис. 5.1 представлена концептуальная модель ABC-метода, из которой четко видно, что Ресурсы (Затраты) в ABC-модели – это входные стрелки, стрелки управления и механизмов в IDEF0-модели (рис. 5.2), Продукты (Стоимостные объекты) ABC-модели – это выходные стрелки IDEF0-модели, а Действия ABC-метода – это Функции в IDEF0-модели.

На более низком уровне, а именно уровне функционального блока, связь IDEF0- и ABC-моделей базируется на трех принципах.

1. Функция характеризуется числом, которое представляет собой стоимость или время выполнения этой функции.

2. Стоимость или время функции, которая не имеет декомпозиции,

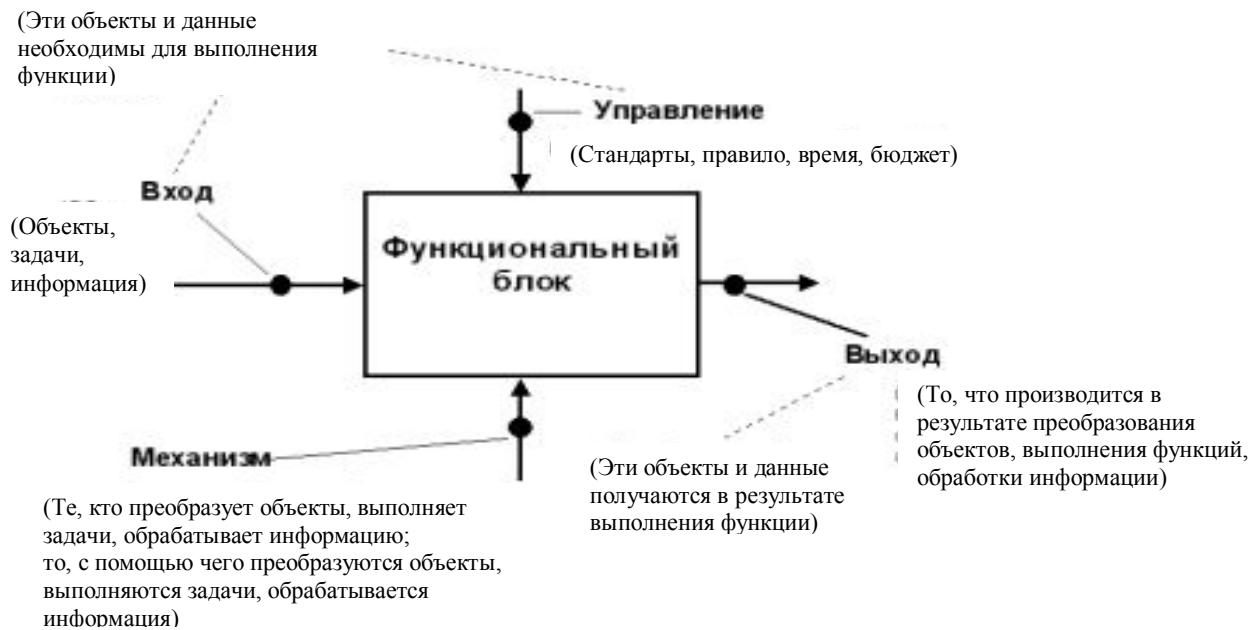


Рис. 5. 2 Функциональный блок со стрелками

определяется разработчиком системы.

Стоимость или время функции, которая имеет декомпозицию, определяется, как сумма стоимостей (времен) всех подфункций на данном уровне декомпозиции.

BPWin предоставляет возможность определять одновременно и денежные и временные характеристики, но отображение на диаграммах в данный момент времени возможно только одной из них.

ABC-метод включает следующие основные понятия:

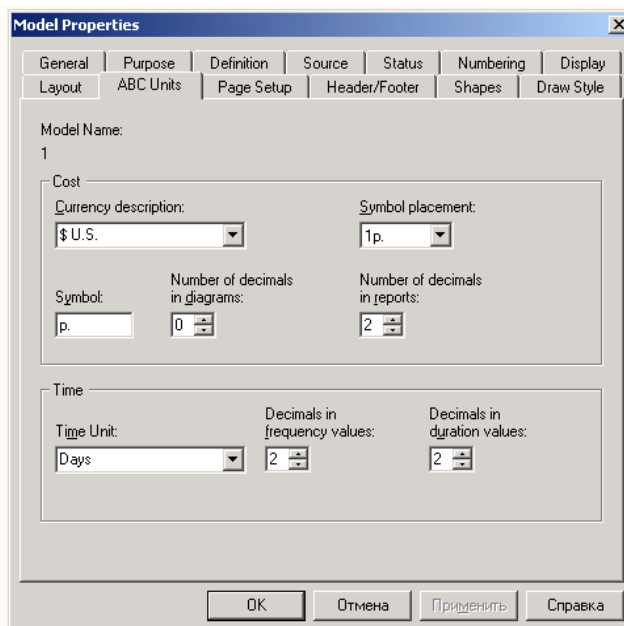


Рис. 5. 3. Выбор параметров стоимости и времени

- *Объект затрат* – причина, по которой работа выполняется: обычно основной выход работы, стоимость работ есть суммарная стоимость объектов затрат.
- *Движитель затрат* – характеристики входов и управлений работы, которые влияют на то, как выполняется и как долго длится работа.
- *Центры затрат*, которые можно трактовать как статьи расхода.

Для проведения ABC- анализа необходимо:

1. Вызвать диалог *Model Properties* (меню *Model* => *Model Properties*), на вкладке *ABC Units* которого нужно установить единицы измерения стоимости и времени – рубли и дни. Если в списке выбора отсутствует необходимая валюта (например, рубль), то ее можно добавить (вписать). Здесь же можно выбрать режим отображения частоты или продолжительности работы переключением соответствующих радиокнопок в группе *ABC Units* (рис. 5.3).

2. Выполнить

Dictionary =>

Cost Center.

Откроется *Cost*

Center Dictionary

(словарь учетного

отдела), в котором

следует внести

название и

определение

центров затрат, приведённых в табл. 5.1.

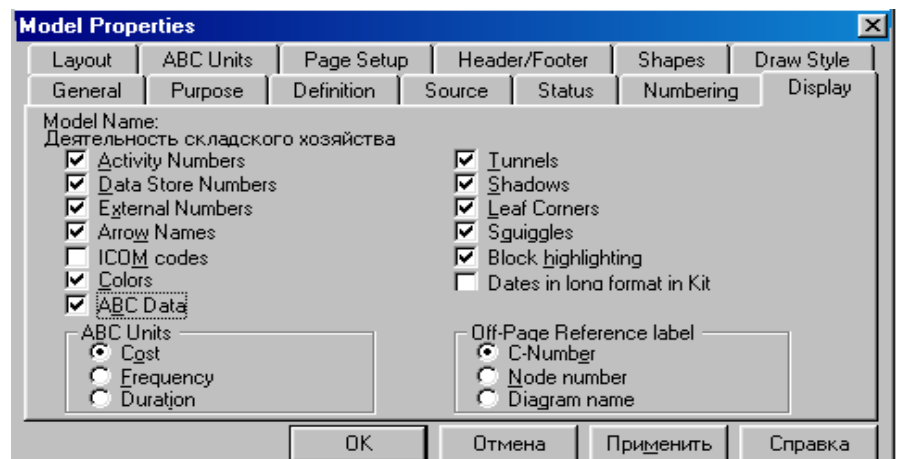


Рис.. 5.4. Вкладка *Display* диалога *Model Properties*

Для отображения стоимости каждой работы в нижнем левом углу прямоугольника в диалоге *Model Properties*, на вкладке *Display* следует включить опцию *ABC Data* (рис. 5.4). Для назначения стоимости работе следует щелкнуть по ней правой кнопкой мыши и выбрать в контекстном меню *Costs*. Откроется вкладка *Costs* диалога *Activity Properties* (рис. 5.5), в которой следует внести стоимость работ.

Activity Properties

UDP Values | UOW | Source | Roles | Box Style

Name | Definition | Status | Font | Color | Costs

Activity Name: Консультации с техническими специалистами

Cost Center	\$ U.S.
Обобщение специалистов	16 000,00
Реклама	0,00
Управление	26 000,00
Учебный центр	20 000,00

This Activity has NO Decomposition.

☐ Override decompositions

☐ Compute from decompositions

Frequency: 10,00

Duration: 1,00 Days

Duration x Frequency: 10,00 Days

Total cost: 62 000,00

Total cost x Frequency: 620 000,00

Cost Center Editor...

OK | Отмена | Применить | Справка

Рис. 5.5. Вкладка *Costs* диалога *Activity Properties*

Для работ на диаграмме A1 “Выявление возможности продаж” (рис. 3.12) вносятся параметры ABC согласно табл. 5.1.

Таблица 5.1

Центры затрат ABC

Центр затрат	Определение
Обучение специалистов	Затраты на обучение, командировки.
Реклама	Затраты на исследование рынка, рекламу, содержание отдела маркетинга.
Учебный центр	Затраты на содержание и оплату технических и других специалистов.
Управление	Затраты на управление, связанные с организацией проведения бесплатных семинаров, консультаций с техническими специалистами и соблюдением инструкций.

Таблица 5.2

Стоимость и продолжительность работ на диаграмме
 “Выявление возможности продаж”

Имя работы (Activity Name)	Центр затрат (Cost Center)	Сумма центра затрат (Cost Center Cost), руб.	Продолжи- тельность (Duration), день	Частота (Frequency)
Download/trial	Ведение сайта с демо-версиями	10000	0,1	10
Первоначальное знание о продукте	Обучение специалистов	15000	0,5	1
	Реклама	30000		
Консультации с техническими специалистами	Обучение специалистов	16000	1,5	10
	Учебный центр	26000		
	Управление	20000		
Бесплатные семинары	Обучение специалистов	14000	1,5	5
	Учебный центр	23000		
	Управление	17000		
Обработка информации о потенциальном клиенте	Обучение специалистов	20000	0,5	4
	Учебный центр	60000		
	Управление	32000		

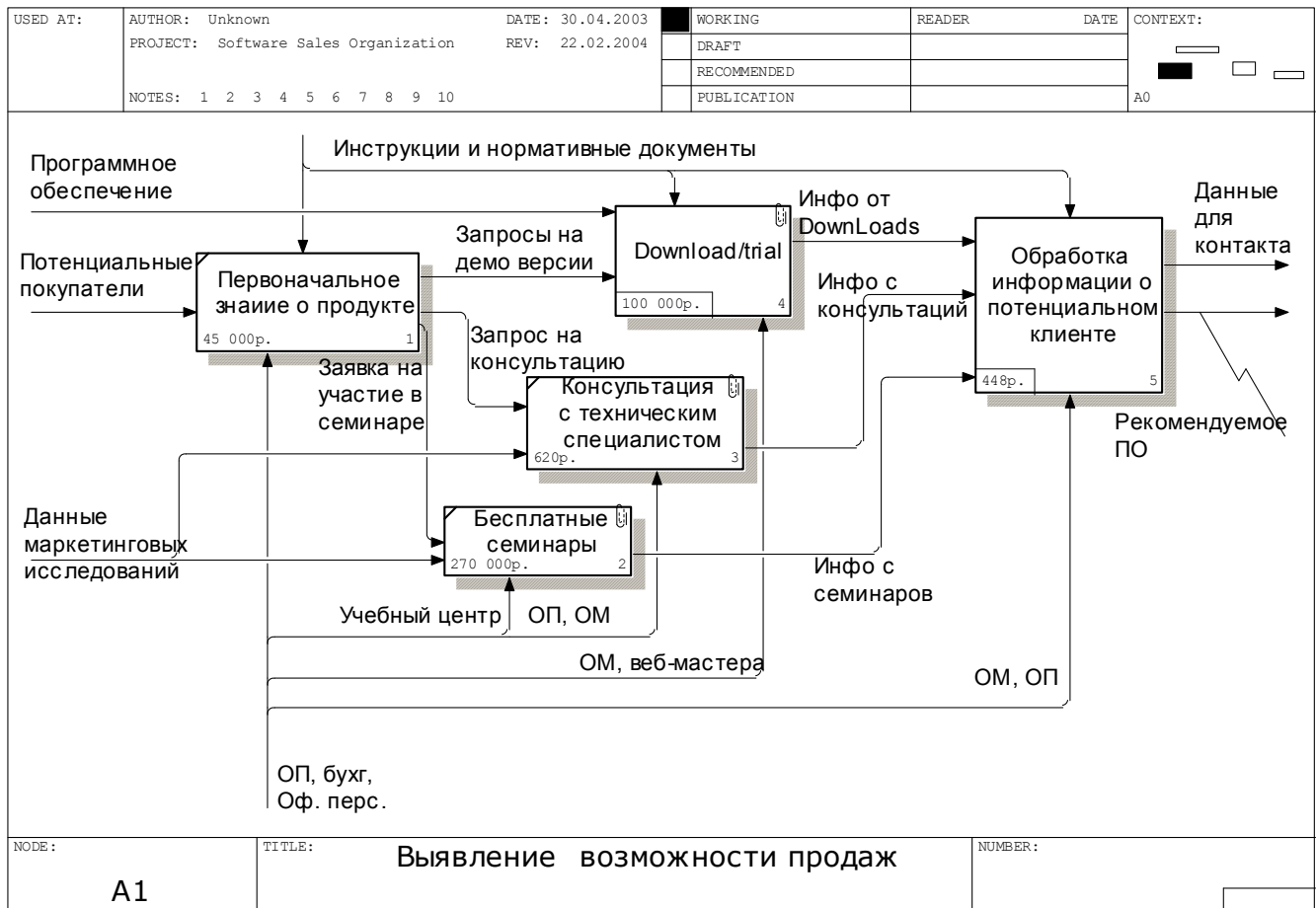


Рис. 5.6. Диаграмма с внесенными затратами

После внесения необходимых значений параметров временные характеристики отображены на рис. 3.12; денежные характеристики – на диаграмме рис. 5.6.

Общие затраты по работе рассчитываются как сумма по всем центрам затрат (рис. 5.7).

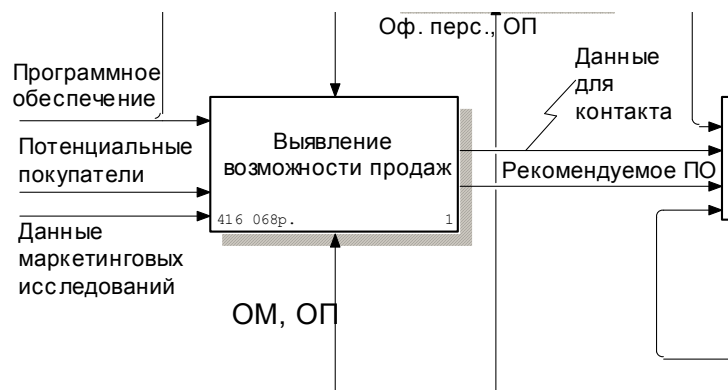


Рис. 5.7. Общая стоимость затрат

6. Моделирование данных

Моделирование данных – это процесс описания информационных структур и фиксации правил бизнес-процессов с целью определения информационных потребностей системы. Проектирование структур данных для поддержания предметной области является второй частью процесса разработки ИС.

Для моделирования данных используется инструментальное средство ERWin, которое было создано для поддержания методологии IDEF1X. Эта методология предусматривает применение различных уровней при моделировании данных. Уровни моделей варьируются от очень широкого и недетализированного взгляда на основные сущности предметной области до уровня точности, который требуется для представления базы данных в виде, понятном целевой СУБД. На самом нижнем уровне, который характеризуется высокой точностью, модель данных является технологически зависимой. На самом высоком уровне, который характеризуется низкой точностью, модель может представлять информацию, которая будет отсутствовать в будущей информационной системе.

Существуют два типа моделей высокого уровня: модели сущность-связь (ERD) и модели, основанные на ключах (Key-Based). ERD определяют основные сущности и их отношения. Модели, основанные на ключах, устанавливают некоторое множество требований к информации и начинают раскрывать детали.

6.1. Моделирование в ERwin

ERwin создает визуальное представление для решаемой задачи. Это представление может использоваться для детального анализа, уточнения и распространения документации, необходимой в цикле разработки.

Процесс построения модели состоит из следующих шагов:

1. Определение сущностей.
2. Определение зависимостей между сущностями.
3. Задание первичных и альтернативных ключей.
4. Определение атрибутов сущности.
5. Приведение модели к требуемому уровню нормальной формы.
6. Переход к физическому описанию модели.
7. Генерация базы данных.

6.1.1. Физическая и логическая модели данных

ERwin имеет два уровня представления модели: логический и физический.

Логическая модель данных описывает факты и объекты, подлежащие регистрации в будущей базе данных. Основными компонентами такой модели являются сущности, их атрибуты и связи между ними. Логический уровень модели может быть построен на основе другой модели. Например, на основе модели процессов.

Физическое проектирование осуществляется на основе логической модели. Результатом этого процесса является *физическая модель данных*, содержащая полную информацию, необходимую для генерации всех объектов в базе данных.

Разделение модели данных на логический и физический уровень позволяет решить следующие задачи.

1. Документирование модели. На физическом уровне объекты базы данных могут называться так, как того требуют ограничения СУБД. На логическом уровне можно этим объектам дать синонимы – имена, более понятные неспециалистам. Такое соответствие позволяет лучше задокументировать модель и даёт возможность обсуждать структуру данных с экспертами предметной области.
2. Масштабирование. ERwin позволяет создавать модели трех типов:
 - модель, имеющую только логический уровень;
 - модель, имеющую только физический уровень;
 - модель, имеющую как логический, так и физический уровень.

Создание модели, как правило, начинается с создания модели логического уровня.

В модели, имеющей оба уровня, ERwin автоматически создаёт соответствующую физическую модель. Это означает, что каждому объекту логического уровня соответствует объект физического, например сущности соответствует таблица.

Модель, имеющая только логический уровень, может быть синхронизирована с несколькими моделями, имеющими только физический уровень. Для создания новой модели запускается ERwin, в открывшемся диалоговом окне *Computer Associates ERwin* выбирается *Create a new model*, а в диалоге *Create Model* – тип новой модели – Logical/Physical.

6.1.2. Интерфейс ERwin. Уровни отображения модели

Для создания моделей ERwin использует три международно признанные системы обозначения (нотации):

1. *Integration DEFinition for Information Modeling (IDEF1X)* – является федеральным стандартом США, а также стандартом в ряде международных организаций.

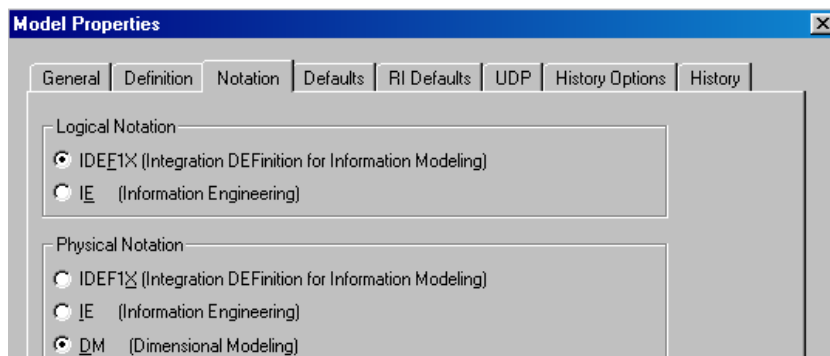


Рис..6.1. Выбор нотации

2. *Information Engineering (IE)* – используется преимущественно в промышленности.
3. *Dimensional Modeling (DM)* – специализированная нотация, предназначенная для разработки хранилищ данных.

На логическом уровне ERwin поддерживает две нотации (IE и IDEF1X), на физическом – три (IE, IDEF1X и DM). Чтобы поменять нотацию, необходимо выбрать: *Model => Model Properties* и вкладку *Notations* (рис. 6.1).

Палитра инструментов изменяется в соответствии с выбранным уровнем модели и нотацией.

Таблица 6.1

Стандартная панель инструментов

Кнопки	Назначение
	Создание, открытие, сохранение и печать модели
	Вызов диалога Report Browser для генерации отчётов. Вызов Report Template Builder – инструмента для публикации отчётов в Internet
	Изменение уровня просмотра модели: уровень сущностей, уровень атрибутов и уровень определений
	Изменение масштаба просмотра модели
	Переключение между областями модели – Subject Area
	Список выбора для переключения между логической и физической моделью

Таблица 6.2

Палитра инструментов на логическом уровне































Вид кнопки IDEF1X	Вид кнопки IE	Назначение
		Указатель мыши
		Создание новой сущности
		Создание новой категории
		Создание идентифицирующей связи
		Создание связи многие ко многим
		Создание неидентифицирующей связи

Таблица 6.3

Палитра инструментов на физическом уровне

Вид кнопки IDEF1X	Вид кнопки IE	Вид кнопки DM	Назначение
			Указатель мыши
			Создание новой таблицы. Для этого нужно щелкнуть левой кнопкой мыши по кнопке и один раз по свободному пространству модели
			Создание нового представления (view). Для этого нужно щёлкнуть левой кнопкой мыши по кнопке и один раз по свободному пространству модели
			Создание идентифицирующей связи
			Создание связи между представлением и временной таблицей
			Создание не идентифицирующей связи

Логическая модель ERwin имеет несколько уровней отображения диаграммы: *уровень сущностей*, *уровень атрибутов*, *уровень определений*. Переключиться между этими уровнями можно при помощи иконок на стандартной панели инструментов (табл. 1).

Также имеются *уровень первичных ключей* и *уровень иконок*. Чтобы переключиться на два последних уровня, необходимо щелкнуть по любому месту диаграммы, не занятому объектами модели. В контекстном меню следует выбрать уровень Display Level, а затем необходимый уровень отображения.

При переключении на уровень иконок показывается большая иконка. Для отображения малой иконки следует выбрать в контекстном меню пункт Entity Display => Entity Icon.







Малая иконка будет показана слева от имени сущности на всех уровнях отображения модели.

Установка цвета и шрифта

Установить шрифт и цвет объектов в ERwin можно при помощи панели инструментов выбора шрифтов и цветов (Font & Color) (табл. 6.4).

Таблица 6.4

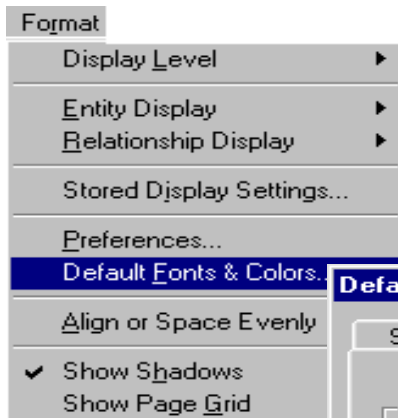
Панель инструментов выбора шрифтов и цветов.

Кнопки	Назначение
	Выбор наименования шрифта
	Выбор размера шрифта
	Выбор стиля шрифта
	Выбор цвета символов
	Выбор цвета заливки
	Выбор цвета линий

Для редактирования цвета и шрифта конкретного объекта следует, щелкнув правой кнопкой мыши по сущности или связи и выбрав из всплывающего меню пункт Object Font & Color, вызвать диалог Font & Colors, в котором можно выбрать шрифт и установить его стиль, цвет, установить цвет заливки и линий.

Для того чтобы изменить шрифт у всех объектов модели или у какой – либо отдельной категории, необходимо в контекстном меню выбрать пункт меню Format => Default Fonts&Colors (или RClick => Default Fonts&Colors. Откроется диалог Default Fonts & Colors (рис. 6.2.) Каждая его вкладка позволяет редактировать шрифт и цвет для определённой категории объектов:

- General – все объекты модели – цвет фона диаграммы;
- Entities (Tables) – наименования сущностей и таблиц; заливка сущностей и таблиц; линии, которыми прорисовываются сущности и таблицы;



обозначение

мощности;

- Subtypes – иерархия категорий;

• Drawing Object Text – текстовые блоки;

- Drawing Object Color – текстовые блоки.

Флажок у надписи

All Objects изменяет шрифт, как у вновь создаваемых, так и созданных уже объектов.

- Attributes (Columns) – атрибуты и колонки, включая отдельные настройки для атрибутов и колонок внешних ключей;
- Relationships – связи, включая имя и

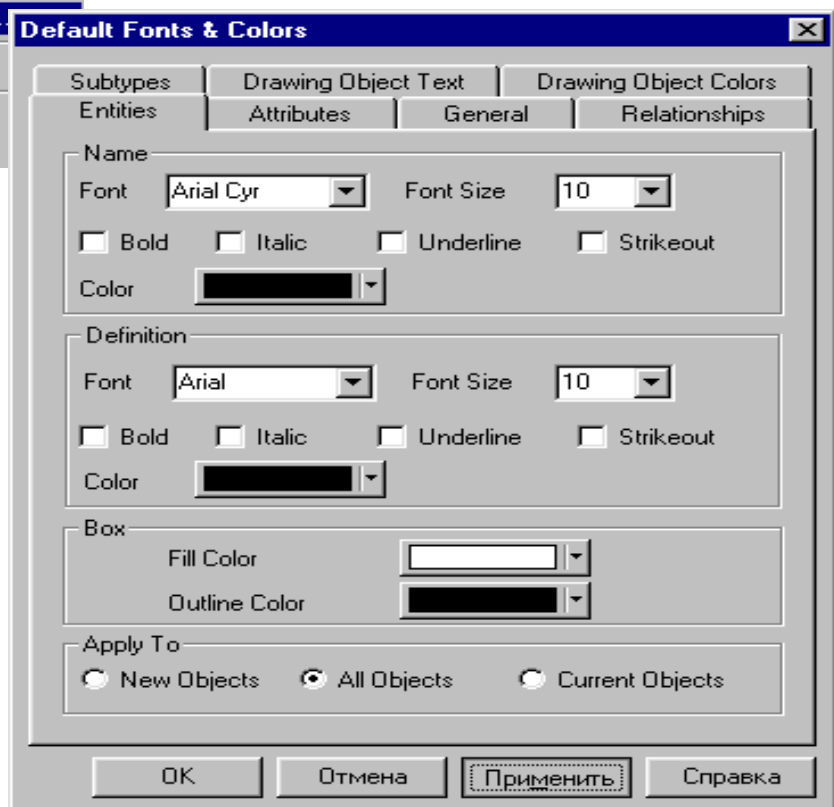


Рис. 6.2. Диалог Default Fonts & Colors

Подуровни логического уровня модели данных


Существует три подуровня логического уровня модели данных.

1. Диаграмма сущность – связь (Entity Relationship Diagram – ERD) – включает сущности и взаимосвязи, отражающие основные бизнес–правила предметной области. Данная диаграмма может включать связи “многие ко многим” и не включать описание ключей. Используется ERD, как правило, для презентаций и обсуждения структуры данных с экспертами предметной области.
2. Модель данных, основанная на ключах (Key Based model – KB) является более подробным представлением данных. Включает описание всех сущностей и первичных ключей. KB предназначена для представления структуры данных и ключей, которые соответствуют предметной области.
3. Полная атрибутивная модель (Fully Attributed model – FA) является наиболее детальным представлением структуры данных: представляет данные в третьей нормальной форме и включает все сущности, атрибуты и связи.

Сущности

Построение логической модели данных является первым этапом в моделировании данных. Для этого необходимо идентифицировать основные сущности предметной области.

Чтобы поместить сущности на диаграмму, необходимо перейти на

уровень логической модели (кнопка на панели инструментов), затем щелкнуть левой кнопкой мыши по кнопке **Entity**  на панели инструментов и один раз по свободному пространству модели.

Чтобы описать сущность, нужно поместить на нее курсор мыши и выполнить RClick => Entity Properties или выбрать Model => Entities на панели инструментов. Откроется диалоговое окно Entities. В него необходимо внести имя, описание сущности (рис. 6.3).

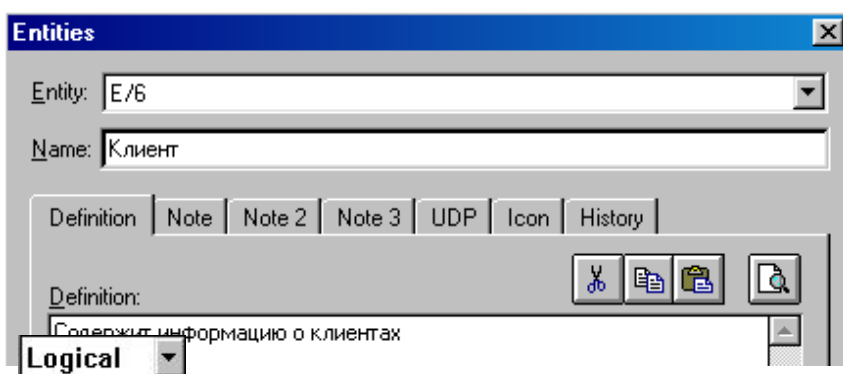



Рис. 6.3 - Вкладка Definition диалога Entities


Диалог Entities содержит следующие вкладки:

- для ввода определения сущности используется вкладка Definition;
- вкладки Note, Note2, Note3, UDP (User Defined Properties – свойства, определенные пользователем) служат для внесения дополнительных комментариев и определений сущности.

ERwin автоматически сохраняет историю всех изменений, связанных с объектами (сущностями, атрибутами и так далее). Вкладка History отображает список изменений. К каждому изменению в окне Comment можно дать комментарий.

Во вкладке Icon каждой сущности можно поставить в соответствие изображение, отображаемое в режиме просмотра модели на уровне иконок. Для связывания изображения с сущностью нужно щелкнуть по кнопке  в появившемся диалоге Icons щелкнуть по кнопке Import и выбрать соответствующий файл формата .bmp. После выбора иконки она отображается во вкладке Icon диалога Entities.

Атрибуты

Чтобы описать атрибуты, необходимо щелкнуть по кнопке  на панели инструментов, затем поместить курсор на сущность и выполнить: RClick => Attributes. В открывшемся диалоге Attributes (рис. 6.4), нужно щелкнуть по кнопке New.

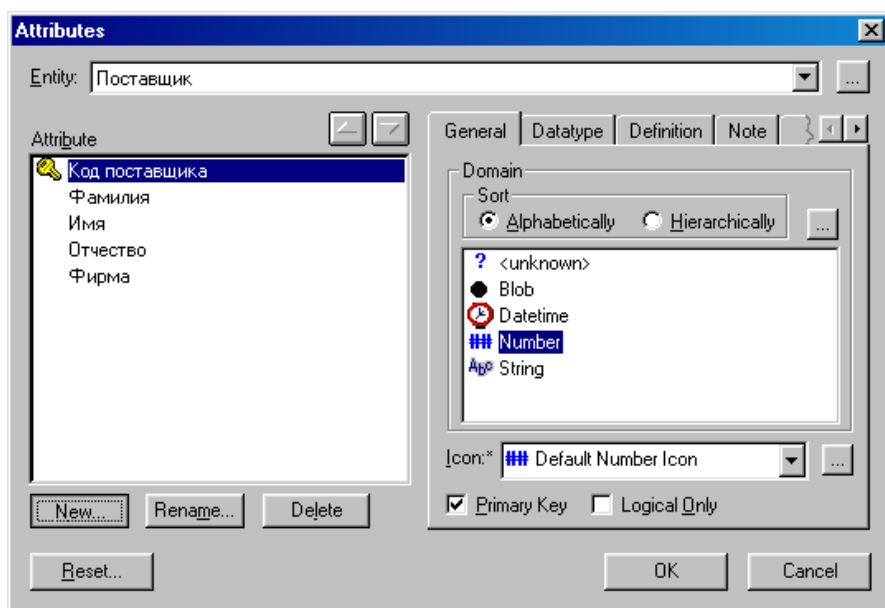


Рис. 6.4. Диалог Attributes

Откроется диалог New Attribute (рис. 6.5). В нем можно указать имя атрибута, имя соответствующей ему в физической модели колонки и домен. Атрибуты должны именоваться в единственном числе и иметь четкое смысловое значение.

Согласно синтаксису IDEF1X, имя атрибута должно быть уникальным в рамках модели. Следовательно, новый атрибут, если его имя совпадает с именем уже существующего, должен быть переименован.

Для атрибутов первичного ключа во вкладке General диалога Attributes (рис. 6.4) необходимо сделать пометку в окне выбора Primary Key. В данной вкладке при помощи списка выбора Icon можно связать иконку с атрибутом.

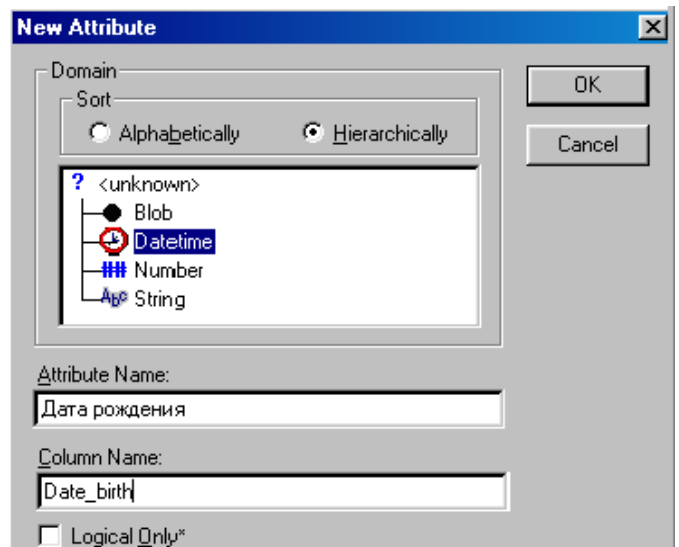


Рис. 6.5. Диалог New Attribute

Для отображения иконки атрибута следует щелкнуть правой кнопкой мыши по любому свободному месту диаграммы, выбрать в контекстном меню пункт Entity Display и в каскадном меню включить опцию Attribute Icon.

На рисунке 6.6 показано отображение сущности на уровне атрибутов с включенной опцией Attribute Icon. Имя сущности находится над прямоугольником, изображающим сущность. Внутри прямоугольника – список атрибутов. Над горизонтальной чертой – атрибуты первичного ключа, ниже – неключевые атрибуты.

- Вкладка Definition позволяет записывать определения отдельных атрибутов.
- Вкладка Note позволяет делать замечания об одном или нескольких атрибутах, которые не вошли в сущность.
- Во вкладке History отображается история создания и изменения свойств атрибута.
- Вкладка UPD служит для задания значений свойств, определяемых пользователем. Предварительно эти свойства должны быть внесены в диалог User Defined Properties как свойства атрибутов.
- Вкладка Key Group позволяет включить атрибут в состав первичного, альтернативного или инвертированного ключа.

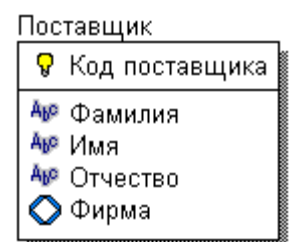


Рис. 6.6.

Также можно задать режимы именования атрибутов. Для этого необходимо выполнить: *Tools => Names => Model Naming Options*. Откроется диалог *Model Naming Options* (рис. 6.6), в его вкладке *Duplicate Names* можно задать следующие режимы:

- *Allow duplicate names*
- *позволить*

внесение одинаковых имен (опция по умолчанию);

- *Automatically Rename duplicate names* – переименовывать атрибуты. То есть при попытке внесения уже существующего имени атрибута ERwin переименовывает его. Например, если атрибут «Фирма» уже существует в модели, другой атрибут (в другой сущности) будет назван Фирма/2;

- *Ask* – запрашивать возможные действия каждый раз при внесении одноимённых атрибутов.

ERwin будет показывать на

экране диалог *Edit Unique Name* каждый раз, когда вводится неуникальное имя сущности или атрибута. В этом диалоге можно ввести другое имя или разрешить дублирование;

- *Disallow duplicate names* – запретить внесение одинаковых имен. Если двойное имя обнаружено, то ERwin выдаст сообщение о том, что ввод неуникальных имен запрещен.

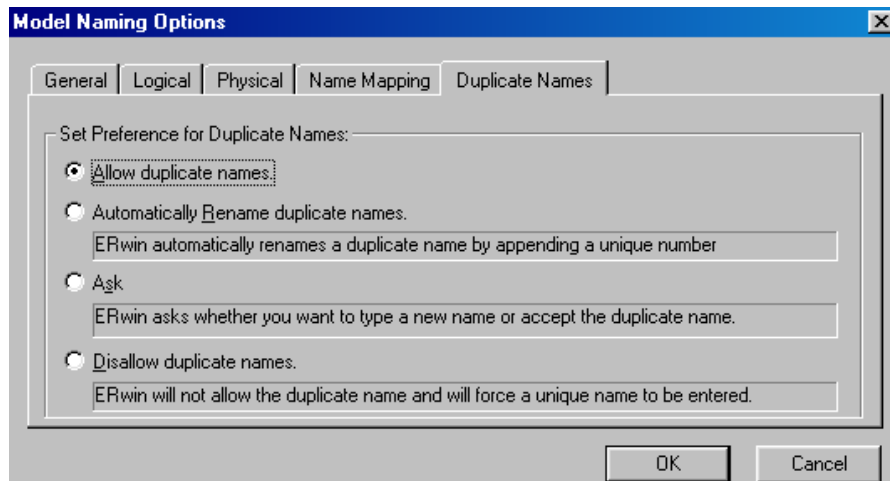


Рисунок 6.7. Диалог *Model Naming Options*

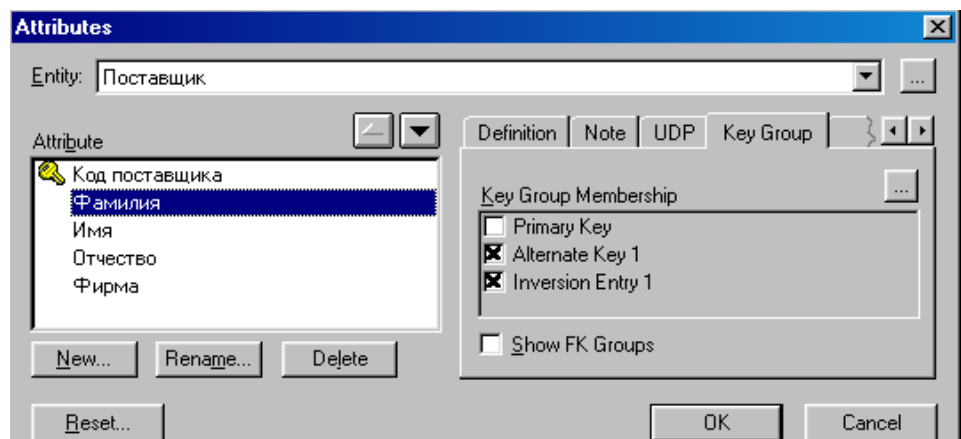


Рисунок 6.8. Вкладка *Key Group* диалога *Attributes*

Определение зависимостей между сущностями

Связь – это понятие логического уровня, которому соответствует внешний ключ на физическом уровне.

Каждая связь должна именоваться глаголом или глагольной фразой (Relationship Verb Phrases). Имя связи выражает некое ограничение или бизнес–правило и облегчает чтение диаграммы, например: Каждый ТОВАР *<поставляется>* ПОСТАВЩИКом.

При определении связи происходит миграция атрибутов первичного ключа родительской сущности в соответствующую область атрибутов дочерней сущности. (По умолчанию атрибуты первичного ключа мигрируют со своими именами, поэтому при неоднократной миграции атрибутов их необходимо переименовать.)

В ERwin на логическом уровне можно установить идентифицирующую связь “один ко многим”, связь ”многие ко многим” и неидентифицирующую связь ”один ко многим ” (табл. 6.2).

Связь называется *идентифицирующей*, если экземпляр дочерней сущности идентифицируется через неё с родительской сущностью. Атрибуты,

составляющие первичный ключ родительской сущности, при этом входят в первичный ключ дочерней сущности (то есть происходит миграция ключей). В дочерней сущности новые атрибуты помечаются как внешний ключ (FK). Дочерняя сущность при идентифицирующей связи всегда является зависимой. Зависимая сущность изображается прямоугольником со скругленными концами, идентифицирующие же связи изображаются сплошной линией.

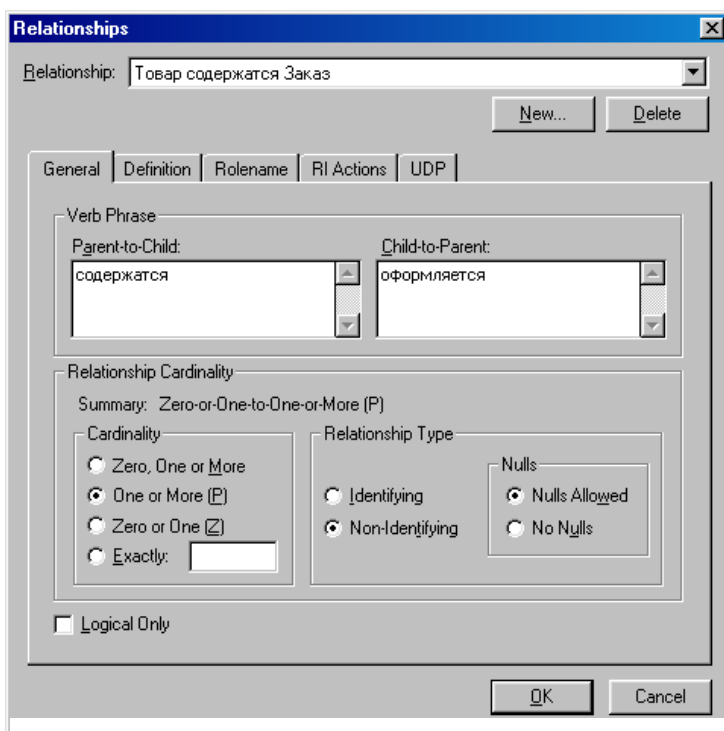


Рисунок 6.9. Вкладка General диалога Relationships

Связь называется *неидентифицирующей*, если экземпляр дочерней сущности идентифицируется иначе, чем через связь с родительской сущностью. При установлении такой связи дочерняя сущность остается независимой. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в состав неключевых атрибутов дочерней сущности. Неидентифицирующие связи служат для связывания независимых сущностей и изображаются пунктирной линией. Линии заканчиваются точкой со стороны дочерней сущности.

Для создания новой связи необходимо:

- щелкнуть левой кнопкой мыши по нужной кнопке связи (идентифицирующая или не идентифицирующая) в палитре инструментов;
- щелкнуть сначала по родительской, а затем по дочерней сущности.

По умолчанию ERwin называет новую связь R/1, R/2, R/3 и так далее. Но можно переименовать связь в любое время. Чтобы отредактировать свойства связи, нужно поместить на нее курсор мыши и выполнить: Rclick => Relationship Properties. Откроется диалог Relationships (рис. 6.9). Во вкладке General данного диалога можно задать мощность, имя и тип связи.

Имя связи (Verb Phrase). Для связи “один ко многим” достаточно указать имя, характеризующее отношение родительской к дочерней сущности (Parent-to-Child). Для связи “многие ко многим” следует указывать имя, как Parent-to-Child, так и Child-to-Parent.

По умолчанию имя связи на диаграмме не отображается. Для отображения имени необходимо выполнить: Format => Relationship Display => Verb Phrase.

Мощность связи (Cardinality) служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней.

Различают четыре типа мощностей.

1. Общий случай, когда одному экземпляру родительской сущности соответствуют ноль, один или много экземпляров дочерней сущности **не помечается** каким – либо символом.

2. Символом **P** помечается случай, когда одному экземпляру родительской сущности соответствуют один или много экземпляров дочерней сущности (исключается нулевое значение).

3. Символом **Z** помечается случай, когда одному экземпляру родительской сущности

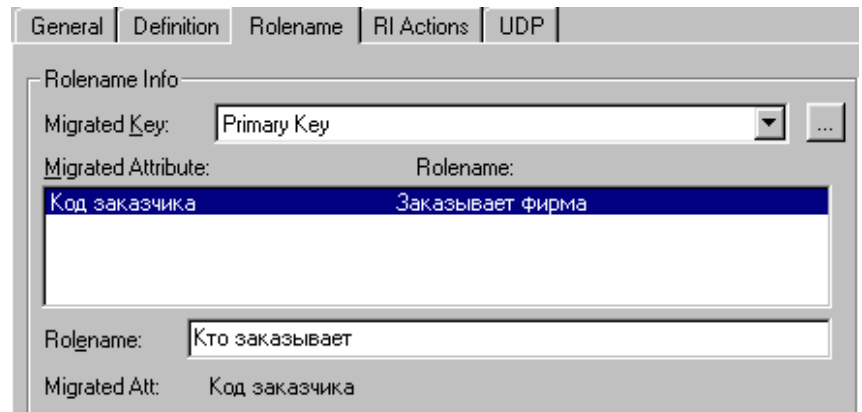


Рисунок 6.10. Вкладка Rolename диалога Relationships

соответствуют ноль или один экземпляр дочерней сущности (исключено нулевое значение).

4. Цифрой помечается случай точного соответствия, когда одному экземпляру родительской сущности соответствует заранее заданное число экземпляров дочерней сущности.

По умолчанию символ, обозначающий мощность связи, не отображается на диаграмме. Для его отображения нужно щелкнуть правой кнопкой мыши по свободному месту диаграммы, не занятому объектами модели, и в контекстном меню выбрать: Relationship Display => Cardinality.

На вкладке Definition диалога Relationships можно дать более полное определение связи. На вкладке Rolename этого же диалога можно задать имя роли и правила ссылочной целостности (рис. 6.10).

Имя роли (функциональное имя) - это синоним атрибута внешнего ключа, который показывает, какую роль играет атрибут в дочерней сущности.

В примере, приведенном на рис. 6.11, в сущности «Заказ» внешний ключ «Код заказчика» имеет функциональное имя «Кто заказывает», которое показывает, какую роль играет этот атрибут в сущности. По умолчанию в списке атрибутов показывается только имя роли. Для отображения полного имени атрибута (как функционального имени, так и имени роли) необходимо щелкнуть правой кнопкой мыши по любому месту диаграммы, не занятому объектами модели, и в открывшемся контекстном меню выбрать Entity Display => Rolename/Attribute.

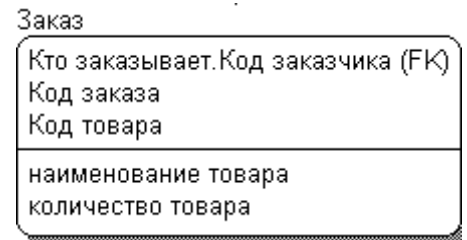


Рис. 6.11. Пример сущности с атрибутами



Рис. 6.12. Случай обязательности имен

Обязательным является применение имен ролей, когда два или более атрибута одной сущности определены по одной и той же области (то есть они имеют одинаковую область значений, но разный смысл).

На рис. 6.12 сущность «Продажа товара» содержит информацию об обращении товара, в котором участвуют два товара – купленный и проданный. Следовательно, сущности «Продажа товара» и «Товар» должны быть связаны дважды и первичный ключ «Код товара» должен дважды мигрировать в сущность «Продажа товара» в качестве внешнего ключа. Необходимо различать эти атрибуты, которые содержат информацию о коде проданного и купленного товара (имеют разный смысл), но ссылаются на одну и ту же сущность «Товар» (имеют общую область значений).

Другим примером обязательности присвоения имен ролей являются *рекурсивные связи*, когда одна и та же сущность является и родительской, и дочерней одновременно. При задании такой связи атрибут должен мигрировать в качестве внешнего ключа в состав неключевых атрибутов той же сущности. Атрибут не может появиться дважды в одной сущности под одним именем, поэтому должен обязательно получить имя роли.

Если атрибут мигрирует в качестве внешнего ключа более, чем на один уровень, то на первом уровне отображается полное имя (имя роли + базовое имя атрибута), на втором и более – только имя роли.

Правила ссылочной целостности

Правила ссылочной целостности (referential integrity – RI) – логические конструкции, которые выражают бизнес–правила использования данных и представляют собой правила вставки, замены и удаления.

ERwin автоматически присваивает каждой связи значение ссылочной

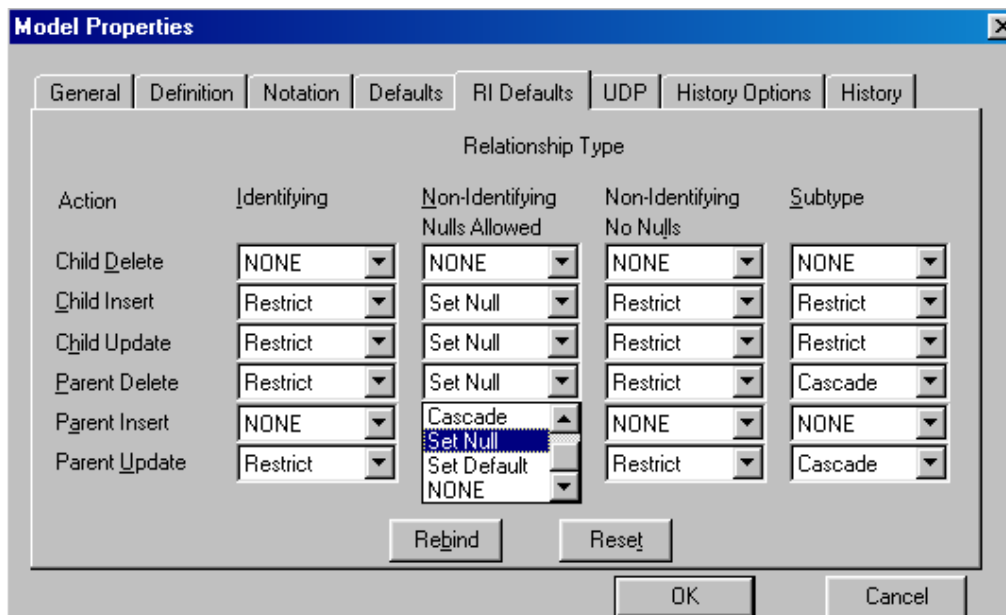


Рис. 6.13. Вкладка RI Defaults диалога Model Properties

целостности, устанавливаемой по умолчанию, прежде чем добавить её в диаграмму.

Режимы RI, присваиваемые ERwin по умолчанию, можно изменить. Для этого нужно выполнить: Model

=> Model Properties. Откроется диалог Model Properties, во вкладке RI Defaults которого можно задать один из режимов RI, представленных в выпадающем списке (рис. 6.13).

Model Explorer

Навигатор модели данных (Model Explorer) – специальная панель для поиска и редактирования объектов модели данных. Располагается слева от окна модели и содержит три вкладки (рис. 6.14):

1. Model – содержит в виде раскрывающегося списка все объекты модели, как физического, так и логического уровня. Model Explorer позволяет редактировать объекты непосредственно в списке. Для этого нужно щелкнуть правой кнопкой мыши по объекту. Откроется контекстное меню, содержащее пункты:

- New – создание объекта;
- Go To – переход к соответствующему объекту на диаграмме;
- Delete – удаление объекта;
- Rename – переименование объекта;
- Properties – редактирование свойств объекта.

2. Subject Areas – служит для отображения и редактирования подмножеств модели. В ней храниться раскрывающийся список подмножеств модели, включающий все объекты подмножеств (хранимые отображения, сущности и др.). В данной вкладке с

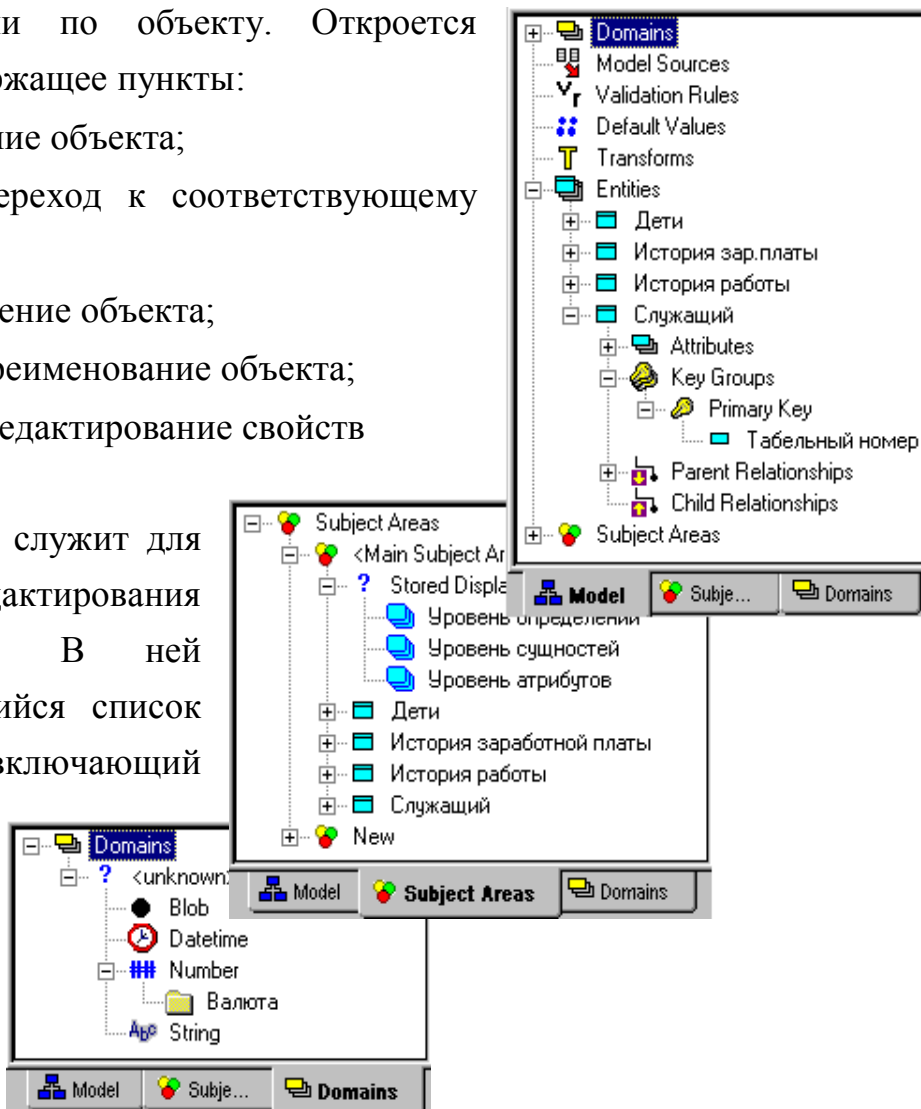


Рис. 6.14. Model Explorer

объектами можно производить такие же манипуляции, как и во вкладке Model.

3. Domains – здесь отображаются домены созданные с модели (то есть домены, представляющие собой абстрактные атрибуты, на основе которых можно создавать обычные атрибуты. При этом создаваемые атрибуты будут иметь все свойства домена-прародителя.

По умолчанию ERwin имеет четыре домена: String, Number, Blob, Datetime.

Чтобы создать домен, нужно перейти во вкладку Domains окна Model Explorer, выбрать родительский домен из списка Domains, щелкнуть по нему правой кнопкой мыши и выбрать пункт Properties. Появляется диалог Domain Dictionary, где следует щелкнуть по кнопке New (рис. 6.15).

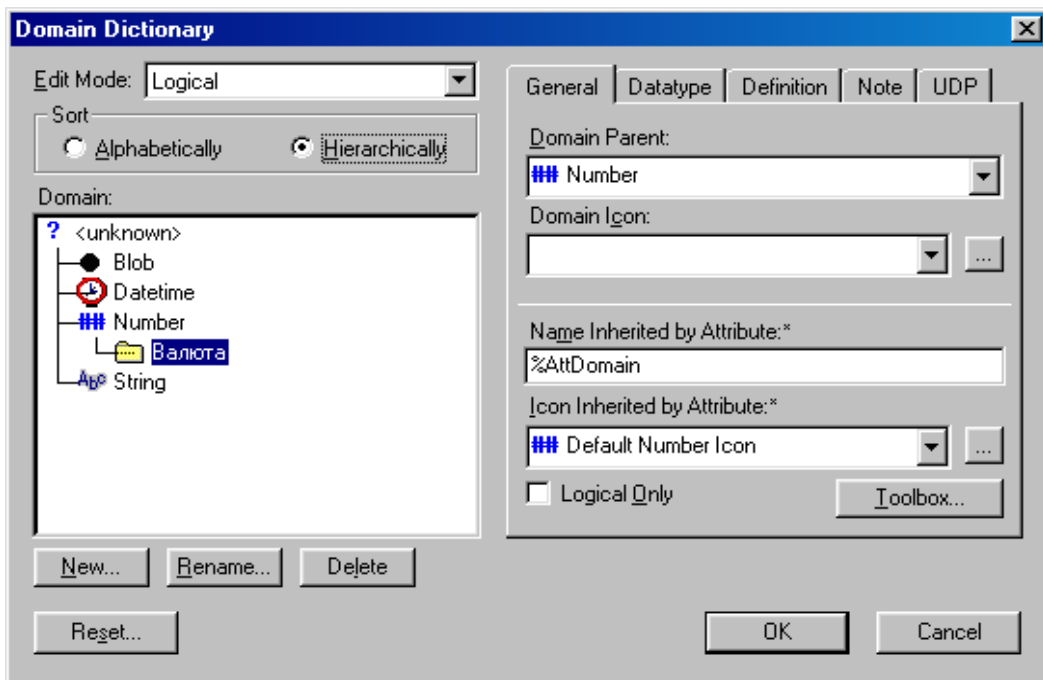


Рис. 6.15. Диалог Domain Dictionary

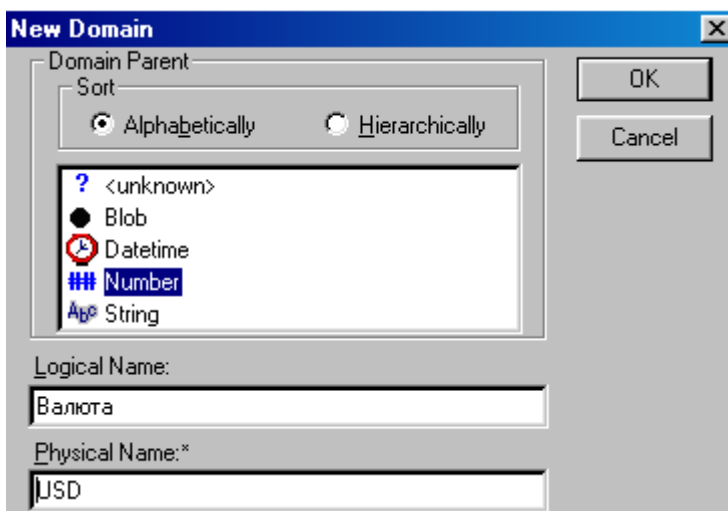


Рис. 6.16. Диалог New Domain

В появившемся диалоге New Domain (рис. 6.16) следует набрать имя домена в поле Logical Name. Можно также указать имя домена на физическом уровне (поле Physical Name), после чего щелкнуть по кнопке ОК. Если не указано имя на физическом уровне, то оно по умолчанию принимает значение имени на логическом уровне. Новый домен наследует все свойства родительского домена.

Создать атрибут на основе домена можно либо так же, как на основе обычного домена, либо просто выбрать созданный домен из списка Domains (вкладка Domains окна Model Explorer) и перетащить его в какую-либо сущность.

6.2. Построение логической модели данных

Построение логической модели данных является первым этапом в моделировании данных. Для этого необходимо идентифицировать основные сущности предметной области. В процессе предыдущего анализа был выявлен ряд сущностей. Среди них ПО (программное обеспечение), Поставщик, Покупатель.

Можно сразу описать эти сущности в виде первой нормальной формы.

Таблица 6.5

ПО (программное обеспечение)

Наименование атрибута	Описание
ИД_ПО	Уникальный код ПО (первичный ключ для связи)
Наименование	Наименование ПО, которое продается в фирме
Описание	Описание ПО (фирма-изготовитель, версия и т.д.)
ИД_типа	Код типа ПО (системное, СУБД, редакторы и т.д.)

Таблица 6.2

Поставщик

Наименование атрибута	Описание
ИДПоставщика	Уникальный код поставщика (первичный ключ)
Наименование	Название фирмы–поставщика
Информация для контакта	Данные для контакта (телефон, факс и т.д.)
Описание	Описание фирмы–поставщика
Контактер	Сотрудник фирмы–поставщика ПО, обслуживающий данную фирму

Таблица 6.3

Покупатель

Наименование атрибута	Описание
ИД_клиента	Уникальный идентификатор покупателя (первичный ключ)
Статус клиента	Код состояния покупателя (потенциальный или действительный)
Тип_клиента	Код типа покупателя (фирма, частное лицо и т.д.)

Определено, что покупатель может не иметь купленного ПО (когда он является потенциальным) либо иметь от одного и более купленного ПО. С другой стороны, каждое конкретное ПО может быть куплено одним или более покупателем или быть не куплено вообще. Если напрямую связать эти сущности, приняв в качестве главного сущность ПО, то произойдёт миграция первичного ключа сущности ПО в сущность Покупатель, для которой это будет внешний ключ.

В соответствии с вышеуказанным требованием, будет наблюдаться избыточность данных, связанная с тем, что при условии покупки одним покупателем более одного экземпляра ПО для каждого такого экземпляра в сущности Покупатель будет содержаться отдельная строка, в которой неоднократно повторяются данные одного и того же покупателя. Это признак того, что отношение Покупатель находится во второй нормальной форме и у него наблюдается транзитивная зависимость неключевых атрибутов от первичного ключа. Оно требует нормализации и приведения к третьей нормальной форме. Для этого необходимо вынести атрибут SoftWareID в отдельное отношение и затем дополнить новое отношение атрибутом ClientID. Это новое отношение будет выполнять служебную роль в базе данных. Главное его назначение состоит в обеспечении связи между ПО и Покупателем. Связанные отношения представлены на рис. 6.17. Промежуточное отношение здесь называется Клиент_ПО.



Рис. 6.17. Отношения ПО и Покупатель в третьей нормальной форме

Аналогично проводятся рассуждения, касающиеся связи ПО и Поставщика. Необходимость применения промежуточного отношения обосновывается тем, что если Поставщик не является производителем ПО, то каждое ПО может поставляться более, чем одним Поставщиком. Связь между ПО и Поставщиком представлена на рис. 6.18.

Отношение Покупатель на рис. 6.17 необходимо уточнить. Моделируемая организация в основном работает с организациями, но нельзя исключать возможность работы с физическими лицами. Для уточнения отношения «Покупатель» применяется связь «супертип-подтип». Супертипом является отношение ClientList, которое содержит общие атрибуты для подтипов.

Подтипами будут отношения «Организация» и «Частное лицо» (рис. 6.31).

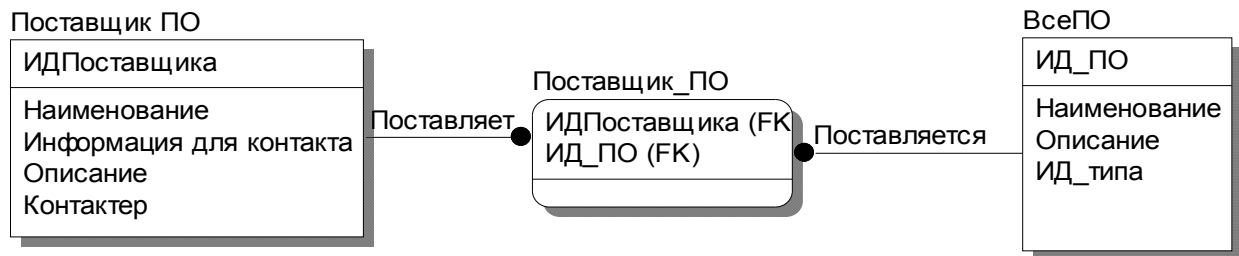


Рисунок 6.18 – Отношения Поставщик и ПО в третьей нормальной форме

Связь на рис. 6.19 является так называемой полной категорией. Это значит, что существует только два типа клиента, которые и представлены на рисунке. Наряду с этим, может также существовать неполная категория, которая допускает появление (выявление) новых типов в процессе функционирования системы. Новые типы добавляются к базе динамически с использованием соответствующих предложений языка SQL (CREATE TABLE).



Рис. 6.19. Связь «супертип-подтип»

Вариант логической модели данных представлен на рис. 6.20. В соответствии с методологией IDEF1X построенная логическая модель данных одновременно является и ФА-моделью, так как на представленных рисунках сущности содержат относительно полный набор атрибутов. Этот набор атрибутов на данный момент ещё не является

фиксированным. В процессе дальнейшего моделирования допускается выявление новых особенностей предметной области, которые найдут своё отражение в наборе атрибутов сущностей.

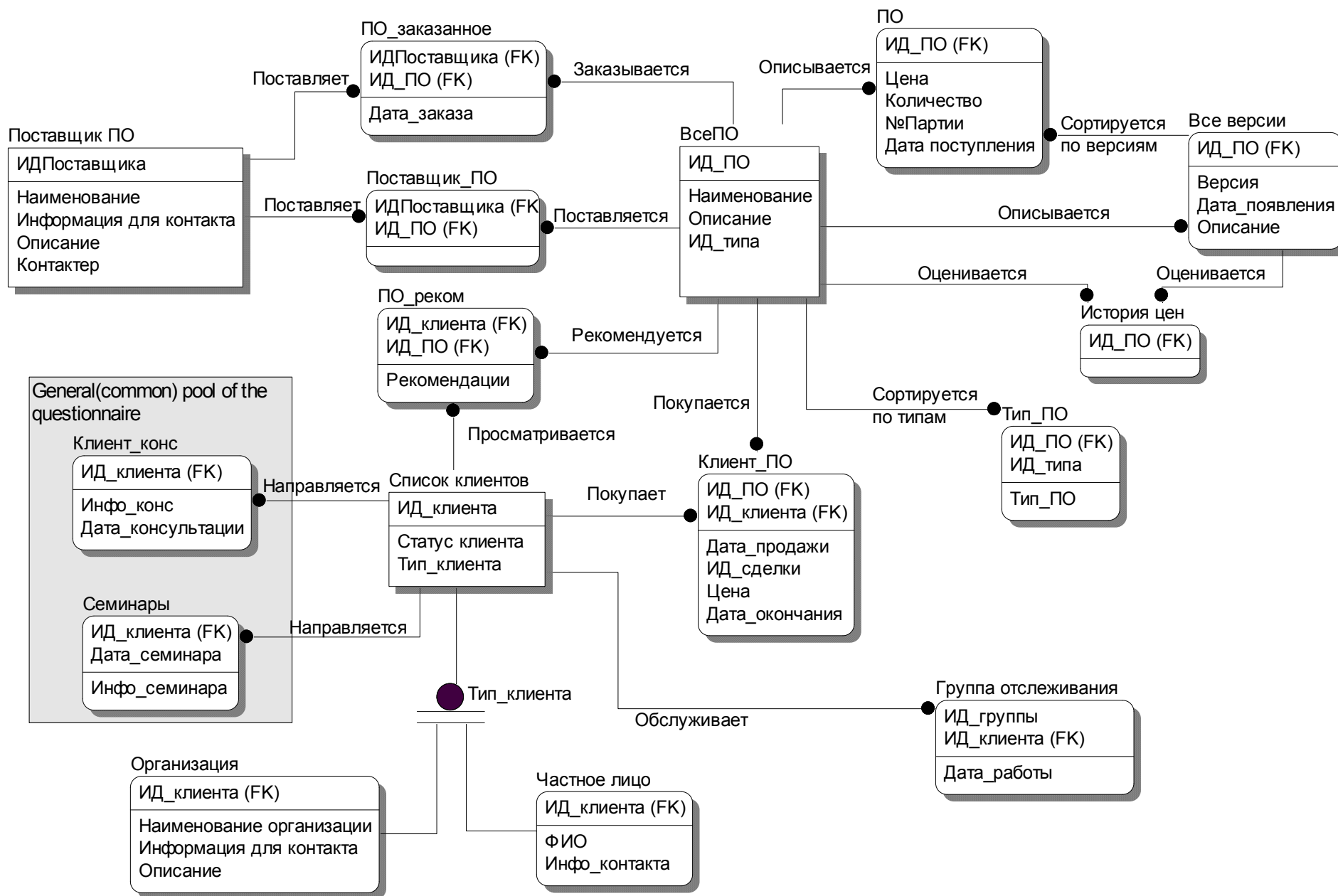



Рис. 6.20. Логическая модель данных

6.3. Построение физической модели данных

Назначение физической модели данных заключается в определении необходимой спецификации для создания адекватной физической модели. Кроме того, физическая модель обеспечивает некий контекст для определения и заполнения словаря элементов данных, которые формируют базу данных и содействует команде разработчиков в разработке программы, которая будет взаимодействовать с базой.

Выбор сервера

Физический уровень модели зависит от выбранного сервера. Для выбора сервера необходимо перейти на физический уровень модели (выбрать *Physical* в окне на панели инструментов  или *Model => Physical Model*). Затем

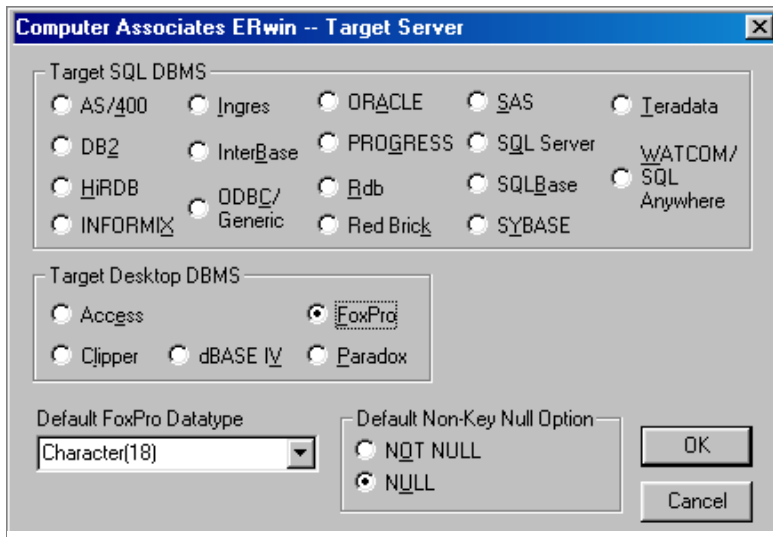



Рис. 6.21. Диалог Target Server

выполнить: меню *Database => Choose Database* или нажать кнопку  на панели инструментов. В открывшемся диалоге *Target Server* (рис. 6.21) нужно щёлкнуть по соответствующей кнопке рядом с именем СУБД.

В раскрывающемся списке *Default Datatype*, который автоматически заполняется типами данных,

поддерживаемыми выбранным сервером, можно выбрать тип данных.

Группа колонок *Default Non-Key Null Option* позволяет разрешить или запретить значения *NULL* для неключевых колонок. При смене СУБД ERwin предлагает автоматически преобразовать тип

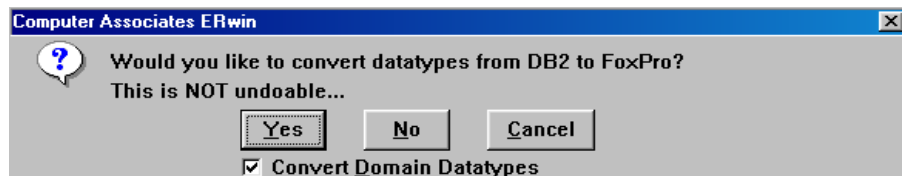


Рис. 6.22. Запрос при преобразовании типов данных

данных каждой колонки на доступный для новой СУБД (рис. 6.22). Правила соответствия типов могут быть заданы предварительно

Таблицы

ERwin автоматически создает имена колонок и таблиц на основе имен соответствующих сущностей и атрибутов, учитывая максимальную длину и другие синтаксические ограничения, накладываемые СУБД. При генерации имени таблицы или колонки по умолчанию все пробелы автоматически преобразуются в символы подчеркивания, а длина имени обрезается до максимальной длины, допустимой для выбранной СУБД. Все изменения, сделанные в редакторах *Tables* или *Columns*, не отражаются на именах сущностей и атрибутов, поскольку информация на логическом и физическом уровнях ERwin хранится отдельно.

Редактор *Tables* позволяет задать свойства таблицы. Для вызова редактора нужно щелкнуть правой кнопкой мыши по таблице и выбрать во всплывающем меню пункт *Tables properties* или выполнить: *Model => Tables*.

Переключиться на другую таблицу можно при помощи раскрывающегося списка выбора в верхней части диалога. Окно *Name* служит для задания имени таблицы. Окно *Owner* позволяет внести имя владельца, отличного от имени пользователя, производящего генерацию схемы базы данных. Окно выбора *Physical Only* служит для создания объектов только на физическом уровне. Если выбрана опция *Generate*, при генерации схемы базы данных будет выполняться команда *CREATE TABLE*.

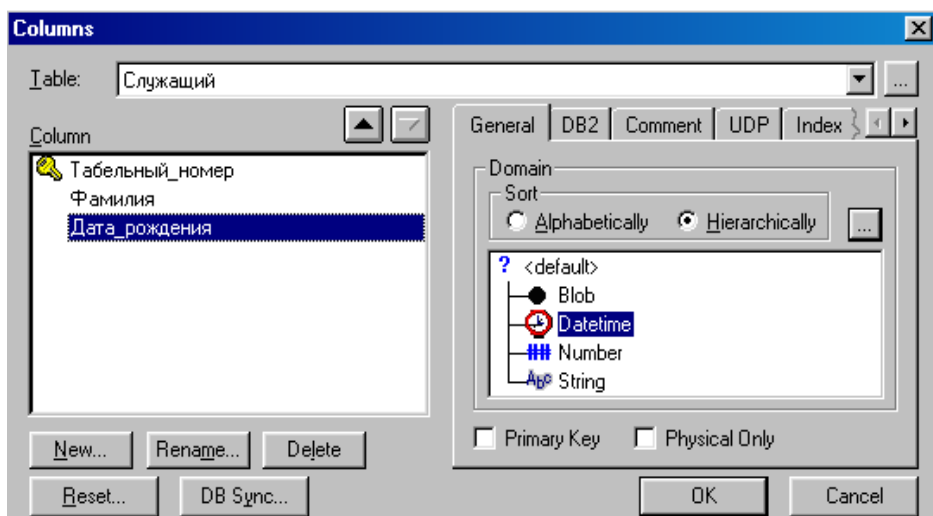
Диалог *Tables* содержит следующие закладки:

- *Dimensional* Доступна только на уровне моделирования хранилищ данных (Dimensional Modeling);
- *Comment* – внесение комментария к таблице;
- *Volumetric* – служит для оценки размера базы данных;
- *Physical Props* – позволяет задать параметры таблицы;
- *Partitions* – служит для задания значений разделения. Доступна только для Oracle 8.x и 9.x;
- *UDP* – задание свойств, определяемых пользователем;
- *Validation* – задание правил валидации (то есть список допустимых значений и/или правила проверки допустимых значений);
- *History* – содержит историю создания и редактирования таблицы;
- *Synonym* – задание синонимов таблицы (если сервер таковые поддерживает).

Колонки

Для задания свойств колонок, отличных от значения по умолчанию, служит редактор *Columns* (рис. 6.23). Для вызова редактора нужно щелкнуть правой кнопкой мыши по таблице и выбрать во всплывающем меню пункт *Columns*.

По умолчанию ERwin присваивает режимы нулевых значений всем неключевым колонкам, исходя из значений по умолчанию,



Рису 6.23. Диалог *Columns*

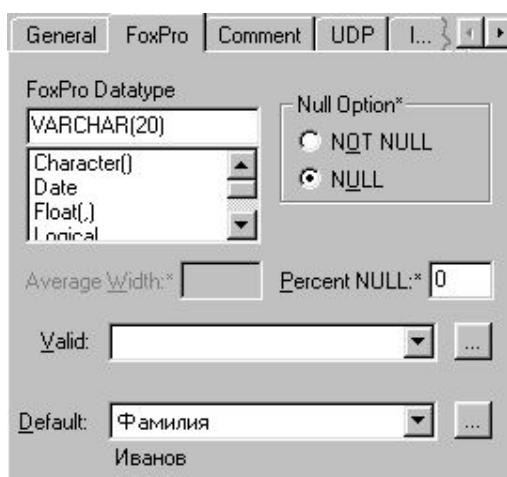



Рис. 6.24. Зкладка *FoxPro* диалога *Columns*

устанавливаемых в редакторе *Target Server*. Для колонок первичного ключа и альтернативных ключей устанавливается режим *NOT NULL*.

В правой части диалога *Columns* используются закладки:

- *General* – позволяет поставить в соответствие колонке определённый домен, создать колонку только на физическом уровне (*Physical Only*) и включить её в состав ключа.


- Зкладка, соответствующая выбранной

СУБД (рис. 6.24). Имя закладки устанавливается автоматически соответствующим выбранной СУБД. Позволяет задать тип данных, опцию *NULL*, значение по умолчанию. Значение по умолчанию должно быть описано и именовано предварительно в диалоге *Default/ Initial Values* (вызывается при нажатии на клавишу );

- *Comment* – служит для внесения комментариев к каждой колонке;
- *UDP* – задание свойств, определяемых пользователем;
- *Data Source* – доступна только при моделировании хранилищ данных;

- *Index* – служит для включения колонки в состав индексов;
- *History* – содержит историю создания и изменения свойств атрибутов.

В левой части диалога содержится упорядоченный список колонок таблицы.

Кнопки  предназначены для перемещения колонки в таблице на позицию вверх и вниз. Кнопки *New*, *Rename* и *Delete* служат для создания, переименования или удаления колонки соответственно. При помощи кнопки *Reset* можно переустановить свойства колонки, заданные вручную, на значения по умолчанию.

ERWin обеспечивает быстрый переход от логической модели к физической путём выбора физического представления модели. Каждый элемент логической модели имеет соответствующий ему элемент в физической модели. Поэтому каждая сущность становится отношением (таблицей), атрибуты становятся столбцами, ключи становятся индексами, связи типа «многие-ко-многим» преобразуются в связи типа «один-ко-многим» путём введения промежуточной таблицы, подтипы и их супертипы подвергаются денормализации, бизнес-правила реализуются в виде триггера или хранимой процедуры, логические типы данных уточняются приведением их к физическим типам данных целевой СУБД и так далее.

Необходимо отметить, что логическая модель данных проектировалась уже с учётом некоторых особенностей реализации. В частности, избегались связи «многие-ко-многим» введением промежуточных сущностей. Это позволяет снять неопределённости реализации, но отвлекает от ключевых моментов модели, так как загромождает диаграмму, что становится актуальным, когда модель разрастётся.

В завершении физического моделирования производится уточнение типов данных для столбцов таблиц. В частности, устанавливается длина строковых полей, указывается диапазон и точность для числовых полей, значения по умолчанию. Диаграмма представлена рис. 6.25.

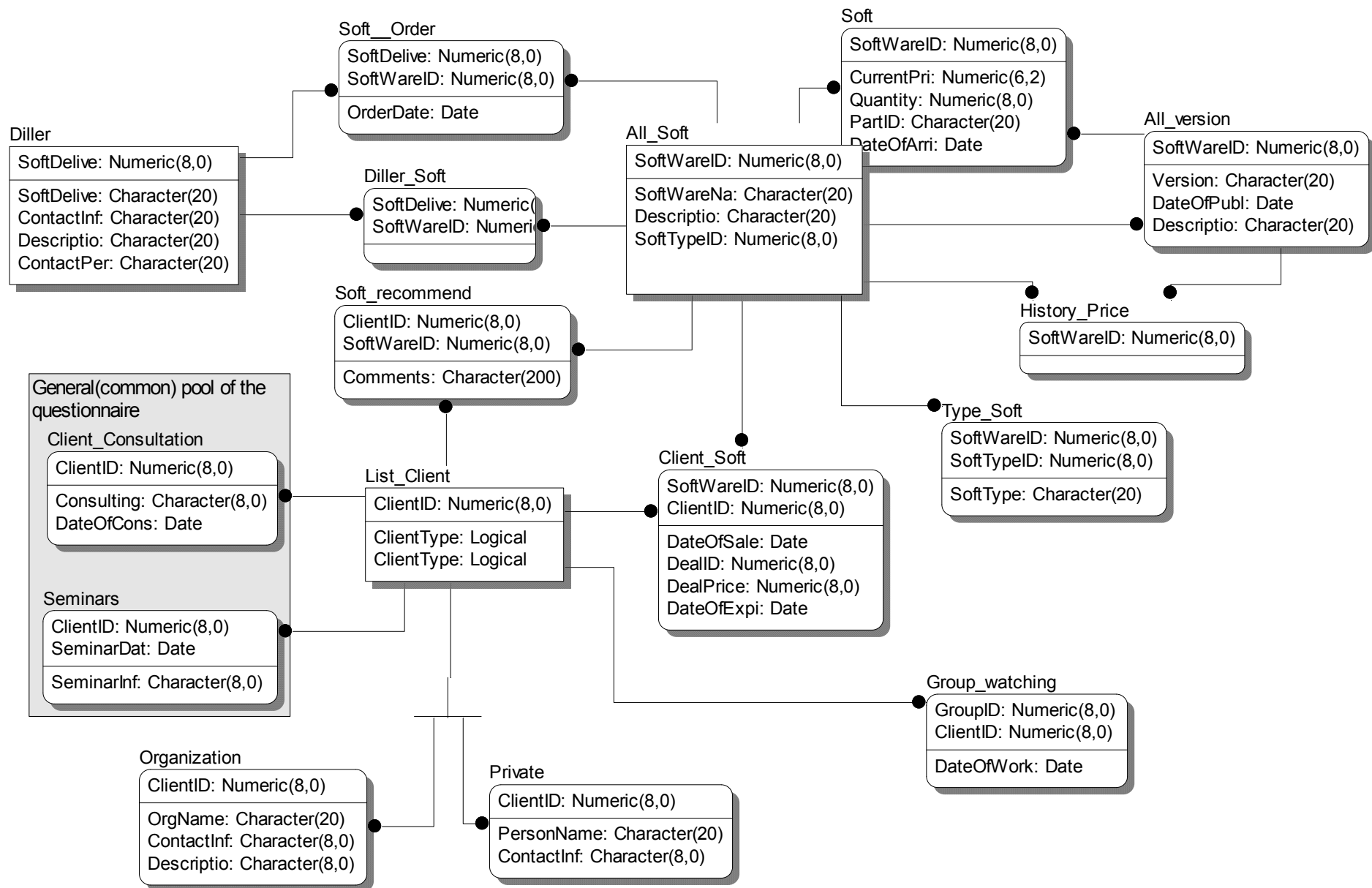



Рис. 6.25. Физическая модель данных

6.4. Прямое и обратное проектирование

Процесс генерации схемы базы данных из модели данных называется *прямым проектированием (Forward Engineering)*. Процесс генерации модели из схемы базы данных называется *обратным проектированием (Reverse Engineering)*.

Для генерации системного каталога базы данных нужно выбрать пункт меню *Tools => Forward Engineering/Schema Generation* или нажать кнопку  на панели инструментов.

Откроется диалог *Schema Generation* (рис. 6.26), который содержит три закладки:

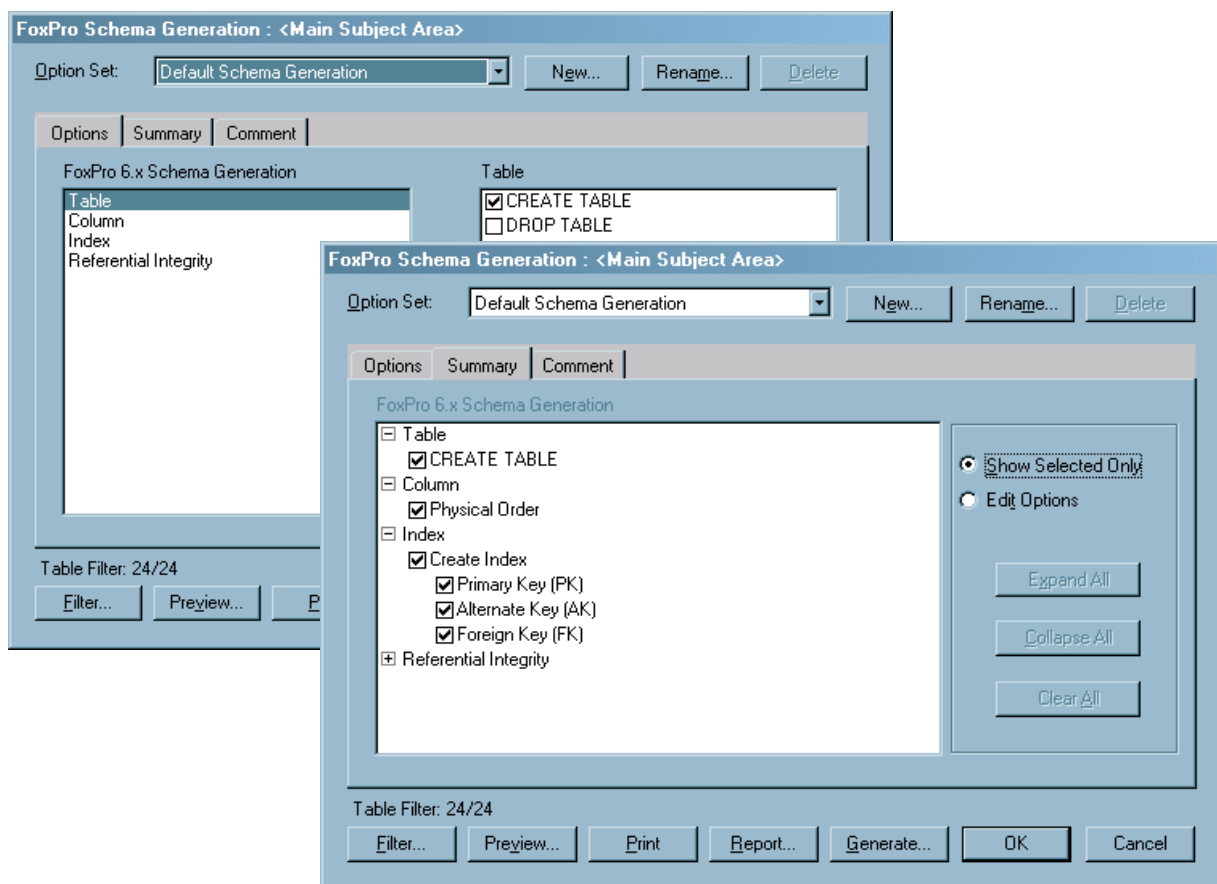


Рис. 6.26. Диалог *Schema Generation*

- *Options* – служит для задания опций генерации объектов базы данных. Для задания опций генерации какого-либо объекта нужно выбрать объект в левом списке закладки, после чего включить соответствующую опцию в правом списке;
- *Summary* – здесь отображаются все опции, заданные во вкладке *Options*. Список опций в *Summary* можно редактировать так же, как и в *Options*;
- *Comment* – позволяет внести комментарий для каждого набора опций.

Кнопка *Preview* диалога *Schema Generation* вызывает диалоговое окно *Schema Generation Preview* (рис. 6.27), в котором отображается SQL-скрипт, создаваемый ERwin для генерации системного каталога СУБД. Нажатие на кнопку *Generate* приведет к запуску процесса генерации схемы.

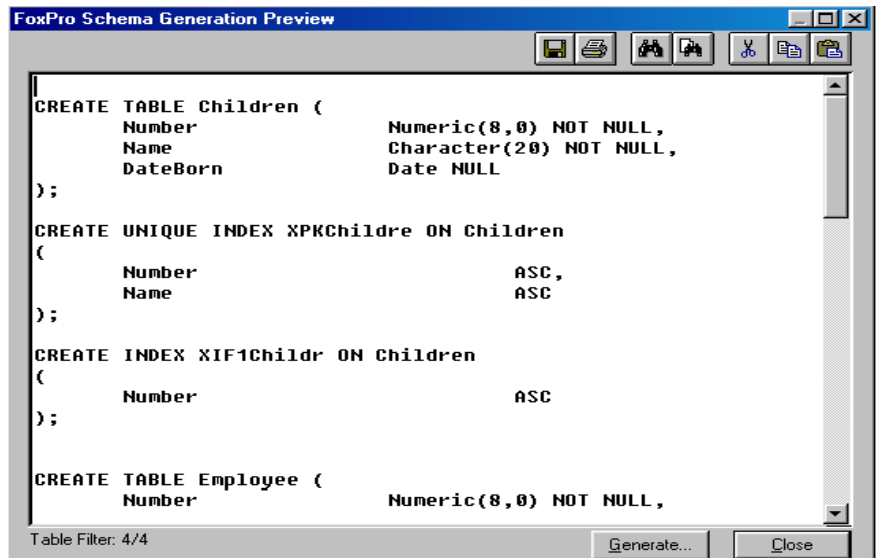


Рис. 6.27. Диалог *Schema Generation Preview*

Кнопка *Print* диалога *Schema Generation* предназначена для вывода на печать создаваемого ERwin SQL-скрипта.

Кнопка *Report* сохраняет тот же скрипт в ERS- или SQL-текстовом файле.

Кнопка *Generate* запускает процесс генерации схемы. Возникает диалог связи с базой данных, устанавливается сеанс связи с сервером базы данных, и начинается выполнение SQL-скрипта. При этом возникает диалог *Generate Database Schema*.

По умолчанию в данном диалоге включена опция *Stop If Failure*. Это означает, что при первой же ошибке выполнение скрипта прекращается (рис. 6.28). Кнопка *Abort* прерывает выполнение. Можно продолжить выполнение, щелкнув по кнопке *Continue*, или после

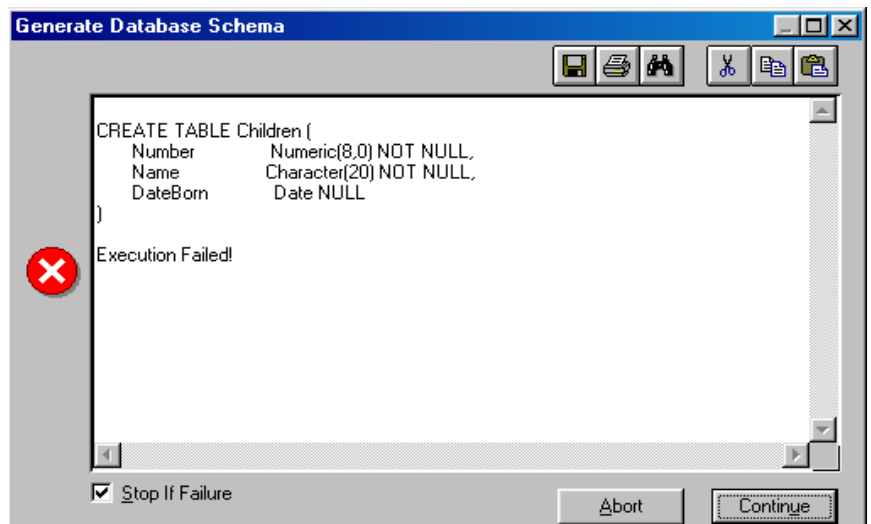


Рис. 6.28. Диалог *Generate Database Schema*

исправления ошибок повторить генерацию (рис. 6.29). При выключенной опции *Stop If Failure* скрипт будет выполняться, несмотря на встречающиеся ошибки.

Для выполнения обратного проектирования нужно выбрать пункт меню:

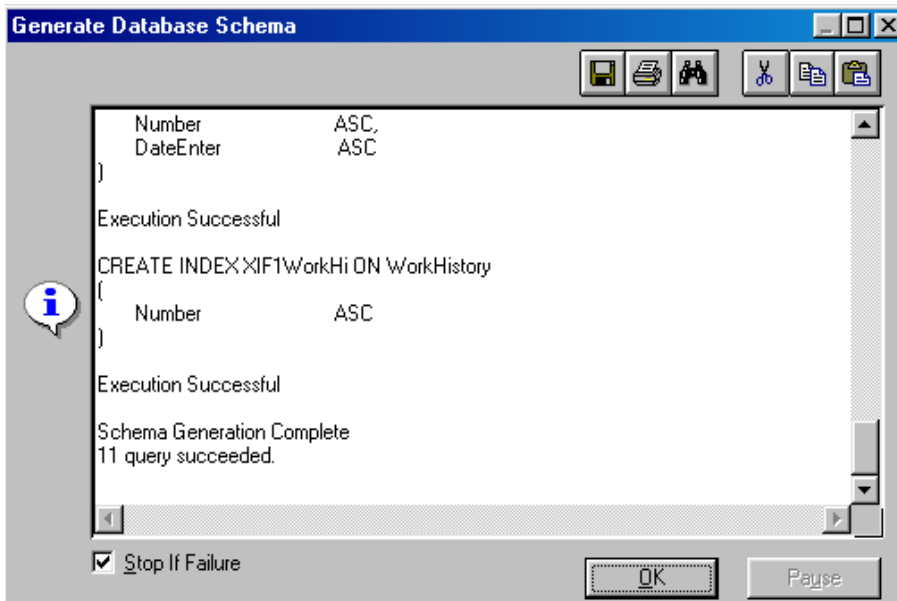



Рисунок 6.29 – Диалог *Generate Database Schema* (генерация завершена)

Tools => *Reverse Engineering* или нажать кнопку  на панели инструментов.

Откроется диалог *Reverse Engineering – Select Template*. (Рис. 6.30), в котором нужно выбрать шаблон диаграммы, затем диалог выбора СУБД и, наконец, диалог задания опций обратного

проектирования *Reverse Engineering – Set Options* (Рис. 6.31).

В диалоге *Reverse Engineering – Set Options* можно задать следующие опции:

- Группа *Reverse Engineering From* позволяет задать источник обратного проектирования – базу данных или SQL(DDL)-скрипт. При помощи кнопки *Browse* можно выбрать текстовый файл, содержащий SQL-скрипт.

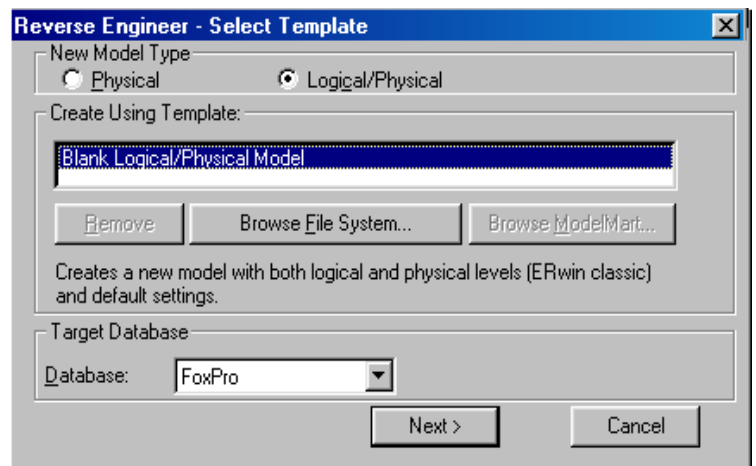


Рисунок 6.30 – Диалог *Reverse Engineering*

- Группа *Items to Reverse Engineering* позволяет задать объекты базы данных, на основе которых будет создана модель.

- Группа *Reverse Engineering* позволяет включить в модель системные объекты и установить фильтр на извлекаемые объекты по их владельцу.

- Группа *Case Conversion* позволяет задать опции конвертации регистра при создании логических и физических имен модели.

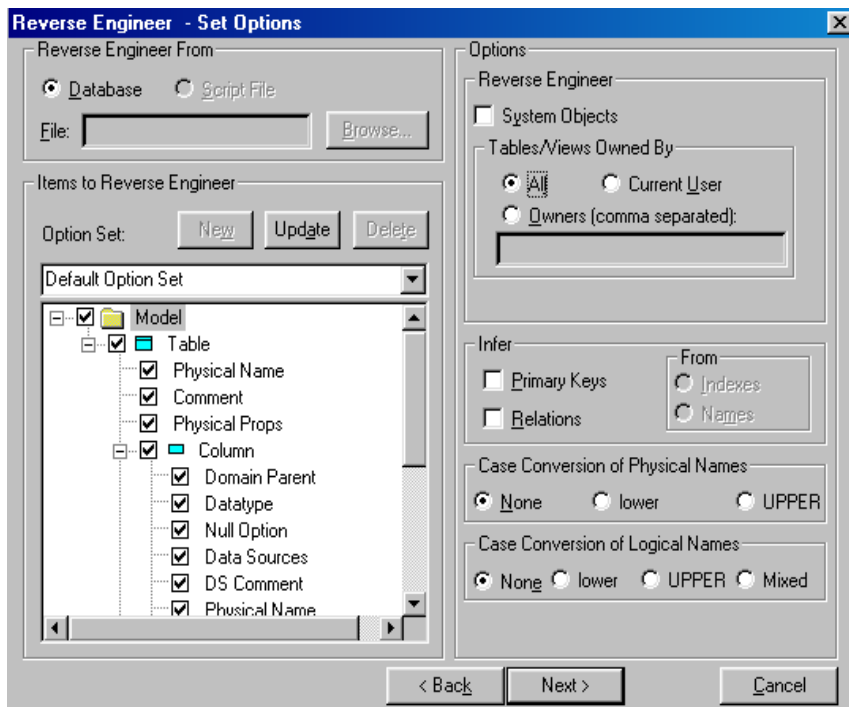


Рис. 6.31. Диалог Reverse Engineering - Set

После задания необходимых опций можно щелкнуть по кнопке *Next*, после чего появляется диалог связи с базой данных, устанавливается сеанс связи с сервером и начинается процесс обратного проектирования, во время которого показывается статус процесса в диалоге *Reverse Engineering-Status*. В результате процесса создается новая модель данных.

7. Связывание функциональной модели и модели данных

Функциональная модель и модель данных могут быть связаны, что приводит к повышению степени интеграции всего процесса разработки информационной системы. Интеграция является ценным механизмом, обеспечивающим постоянство наименований (единую терминологию) и полноту моделей.

Собственно связывание заключается в экспорте данных из ERWin в BPWin с дальнейшим присвоением данных потокам данных и функциям. Это можно осуществить при наличии достаточно хорошо проработанных функциональной модели и модели данных. При этом первоначально каждая из них представляет собой половину необходимого описания системы. Связывание формирует целостную систему. Этот способ может быть использован для организации параллельной работы по моделированию информационной системы.

Однако уже в процессе построения функциональной модели можно выделить ряд элементов данных – сущности и их атрибуты. В этом случае выявленные элементы данных могут быть использованы для построения модели данных «с нуля» путём их экспорта в ERWin. После этого предстоит лишь определить связи между сущностями и провести нормализацию.

7.1. Присвоение импортированных сущностей потокам данных и использование данных активностями

Информация, которая моделируется в виде одной стрелки в модели процессов, может содержаться в нескольких сущностях и атрибутах в модели данных. Кроме того, на диаграмме модели процессов могут присутствовать различные стрелки,

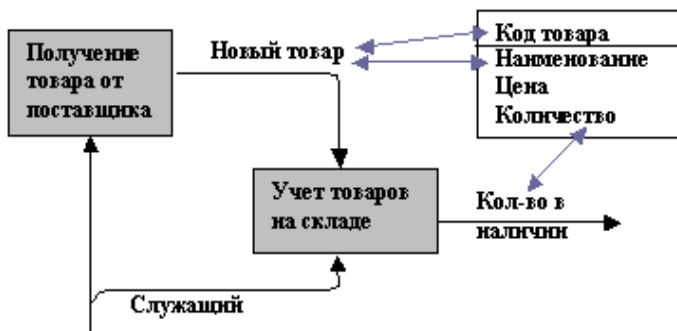


Рис. 7.2. Преобразование стрелки в атрибут

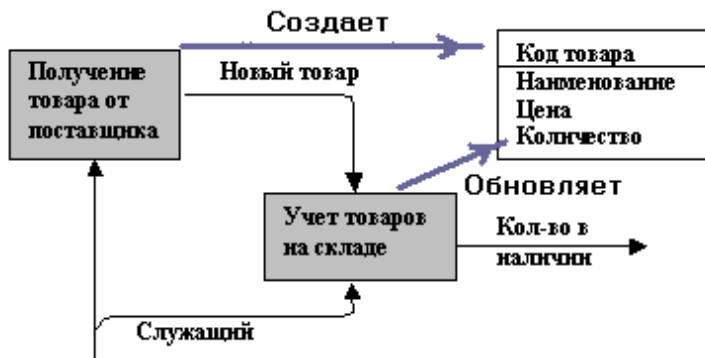


Рис. 7.3. Воздействие работы на сущность

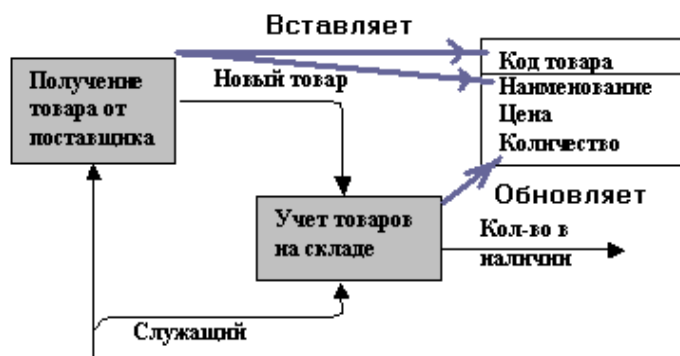


Рис. 7.4. Воздействие работы на атрибуты



Рис. 7.1. Преобразование стрелки в сущность

изображающие одни и те же данные, но на разных этапах обработки (например, необработанная информация – упорядоченная информация – отчетный документ). Информация о таких стрелках находится в одних и те же сущностях. Следовательно, одной и той же стрелке в модели процессов могут соответствовать несколько сущностей в модели данных и наоборот, одной сущности может соответствовать несколько стрелок.

Стрелке в модели процессов может соответствовать отдельная сущность в модели данных.

Так, стрелке “Заказчик” на рис. 7.1 соответствует сущность “Покупатель”.

Информация о стрелке может содержаться только в нескольких атрибутах сущности. Разным атрибутам в одной и той же сущности могут соответствовать разные стрелки.

На рис. 7.2 стрелка “Новый товар ” соответствует атрибутам “Код товара” и ”Наименование”. Стрелка “Количество в наличии ” соответствует атрибуту “Количество”.

Работы в модели процессов могут создавать или изменять данные, которые соответствуют входящим или исходящим стрелкам. Они могут воздействовать как целиком на сущности, создавая или модифицируя экземпляры сущности (рис. 7.3), так и на отдельные атрибуты сущности (рис. 7.4).

Многие потоки данных на функциональных диаграммах уже имеют наименования, соответствующие сущностям и их атрибутам. Поэтому после осуществления операции импортирования данных из ERWin в BPWin следует присвоить потокам данных соответствующие сущности (табл. 7.1).

Поток данных указывает на атрибут или набор атрибутов сущности, или на набор сущностей. Следовательно, интерфейс между функцией и потоком данных представляет объекты, используемые или создаваемые данной конкретной функцией.

Таблица 7.1

Соответствие сущностей потокам данных

Name	Arrow	Entity	Attribute	CRUD	IRUN	Arrow Type
Первоначальное знание о продукте	Запросы на демо версии	Список клиентов	ИД_клиента	C U	I U	O
	Потенциальные покупатели	Список клиентов	ИД_клиента	RU	RU	I
			Статус клиента		RU	
Получение ПО от поставщика	Полученное ПО	ПО		C U		
			Цена		I U	O
			Количество		I U	
	Поставщики	Поставщик ПО	ИД_ПО ИДПоставщика	RU	I U RU	I
			Наименование		RU	
			ИД_ПО		RU	

Окончание табл. 7.1

Name	Arrow	Entity	Attribute	CRUD	IRUN	Arrow Type
	ПО	Все версии	Дата появления	RU	RU	I
			Описание		RU	
			ИД_ПО		RU	
			Версия		RU	
		ВсеПО	Описание	RU	RU	
			ИД_типа		RU	
			ИД_ПО Наименование		RU RU	
		ТипПО	Тип ПО	RU	RU	
			ИД_типа		RU	
			ИД_ПО		RU	
		ПО	Цена	RU	RU	
			Количество		RU	
			ИД_ПО		RU	
			Версия		RU	

Данные не могут использоваться функциями произвольно. Только конкретные типы операций (создание, извлечение, обновление или удаление) разрешены для каждого из четырёх типов стрелок, определяющих внешние связи функции. Существуют правила использования для сущностей и атрибутов со стороны функций. Исходя из смысла сущностей и атрибутов, приняты небольшие различия между операциями, совершаемыми над ними (табл. 7.2).

Таблица 7.2

CRUD и IRUN для сущностей и атрибутов

Сущности	Атрибуты	Различия
Create	Insert	Только сущности могут быть созданы; значения атрибутов вставляются в сущность-владельцу
Retrieve	Retrieve	Нет различий
Update	Update	Нет различий
Delete	Nullify	Только экземпляры сущности могут быть удалены. Удаление атрибута заключается в установке значения атрибута в NULL (пустое значение)

Акроним допустимых операций над сущностями – CRUD (Create, Retrieve, Update, Delete). Акроним для допустимых операций над атрибутами – IRUN (Insert, Retrieve, Update, Nullify). Они являются синонимами словосочетания «использование данных».

BPWin предоставляет средства автоматического контроля за соблюдением данных ограничений.

7.2. Задание свойств, определяемых пользователем

Свойства, определяемые пользователем (*User Defined Properties* – UDP), создаются проектировщиком для присвоения функциям и потокам данных особенных характеристик моделируемой организации. Они используются для повышения точности модели и ее завершённости.

Для определения *UDP* служит диалог *User Defined Properties* (рис. 7.5), который вызывается из меню Model => UDP Dictionary. В нем необходимо указать вид объекта, для которого заводится **UDP** (диаграмма в целом, сущность, атрибут и так далее) и тип данных.

ERwin поддерживает для UDP шесть типов данных:

- 1) Date – дата. Используется формат ММ/ДД/YY. Для выбора значения даты можно использовать контекстный календарь;
- 2) Int – целое число;
- 3) Real – действительное число;
- 4) Text – строка;
- 5) List – список. При задании списка в диалоге User Defined Properties значения следует разделять запятой, значение по умолчанию выделяется символом ~;
- 6) Command – команда (выполняемая строка).

Значение свойств, определяемых пользователем, задается на вкладке *UDP* диалога *Entities*.

Список UDP, определённых в модели, представлен в таблице 7.3.

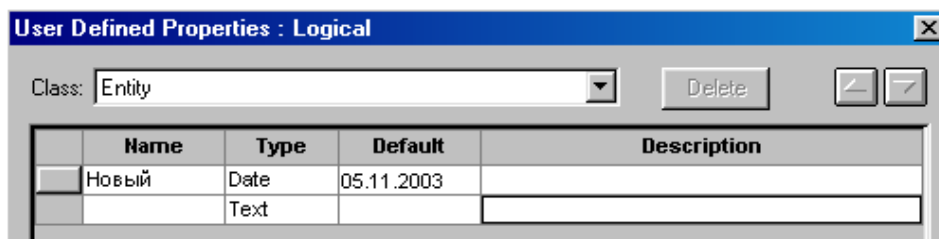


Рисунок 7.5 – Диалог *User Defined Properties*

Таблица 7.3

Свойства, определяемые пользователем – UDP

Name	Definition	UDP Datatype	Value
Знание о продукте	Степень информированности клиента, степень его уверенности и оформленности требований	TextList (Multiple selections)	Полное Только потребности Частичное
Состояние ПО	Имеется в наличии, заказано или не имеется в наличии	TextList (Multiple selections)	Имеется Отсутствует и не заказано Отсутствует, но заказано
Статус клиента	Потенциальный или действительный	TextList (Multiple selections)	Действительный Потенциальный
Статус поставщика	Описывает, заключён ли договор с поставщиком о сотрудничестве	TextList (Multiple selections)	Действительный Не действительный

Библиографический список

1. Маклаков С. Моделирование бизнес-процессов с BPwin. – М.: Диалог-МИФИ, 2003. – 224с.: ил.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 2-е изд./Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 2000. – 560 с., ил.
3. Вендров А. М. Проектирование программного обеспечения экономических информационных систем: Учебник. – М.: Финансы и статистика, 2002. – 352с.: ил.
4. Вендров А.М. Case-технологии. Современные методы и средства проектирования информационных систем. М.: Финансы и статистика, 1998.