

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

по дисциплине

ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**Для студентов
специальности 1-40 01 01
«Программное обеспечение информационных
технологий»**

Минск 2007

Авторы:

– Бахтизин Вячеслав Вениаминович, канд.техн.наук, заведующий кафедрой «Программное обеспечение информационных технологий» Белорусского государственного университета информатики и радиоэлектроники;

– Глухова Лилия Александровна, канд.техн.наук, доцент кафедры «Программное обеспечение информационных технологий» Белорусского государственного университета информатики и радиоэлектроники.

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА № 1. СТРУКТУРНЫЙ АНАЛИЗ И МОДЕЛИРОВАНИЕ В СРЕДЕ CASE-СРЕДСТВА BPWIN5

| | |
|---|----|
| ВВЕДЕНИЕ..... | 6 |
| 1.1.МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПОДДЕРЖИВАЕМЫЕ BPWIN..... | 7 |
| 2.1.1. Общие сведения..... | 7 |
| 2.1.2. Методология функционального моделирования IDEF0..... | 7 |
| 2.1.3. Методология DFD..... | 12 |
| 2.1.4. Методология IDEF3..... | 14 |
| 1.2.ИНСТРУМЕНТАЛЬНАЯ СРЕДА BPWIN 2.5..... | 16 |
| 2.1.5. Интегрированная среда разработки..... | 16 |
| 2.1.6. Описание команд и пунктов главного меню..... | 18 |
| 1.3.СОЗДАНИЕ МОДЕЛЕЙ В ИНСТРУМЕНТАЛЬНОЙ СРЕДЕ BPWIN 2.5...22 | 22 |
| 2.1.7. Создание IDEF0-модели..... | 22 |
| 2.1.8. Особенности построения DFD-диаграмм..... | 37 |
| 2.1.9. Особенности построения IDEF3-диаграмм..... | 38 |
| 1.4.СОЗДАНИЕ ОТЧЁТОВ В BPWIN 2.5..... | 40 |
| 1.5.ПОСЛЕДОВАТЕЛЬНОСТЬ ОСНОВНЫХ ШАГОВ ПРИ СОЗДАНИИ IDEF0-МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ..... | 45 |
| 1.6.КОНТРОЛЬНЫЕ ВОПРОСЫ..... | 46 |
| 1.7.ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНЫХ И ИНДИВИДУАЛЬНЫХ РАБОТ..... | 47 |
| ЛИТЕРАТУРА К ЛАБОРАТОРНОЙ РАБОТЕ № 1..... | 48 |

ЛАБОРАТОРНАЯ РАБОТА № 2. ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ В СРЕДЕ CASE-СРЕДСТВА ERWIN49

| | |
|--|----|
| 2.1.ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ..... | 50 |
| 2.1.1. Реляционные базы данных..... | 50 |
| 2.1.2. Сущности и атрибуты в реляционной модели..... | 51 |
| 2.2.МОДЕЛИРОВАНИЕ В ERWIN..... | 52 |
| 2.2.1. Отображение логического и физического уровня модели данных в ERwin..... | 52 |
| 2.2.2. Компоненты диаграммы ERwin и основные виды представлений диаграммы..... | 53 |
| 2.2.3. Инструменты для создания модели в ERwin..... | 54 |
| 2.2.4. Идентификация сущностей. Сущности (Entities) в ERwin..... | 54 |
| 2.2.5. Связи (relationships) в ERwin..... | 55 |
| 2.2.6. Связи категоризации..... | 56 |
| 2.2.7. Графическое редактирование модели..... | 57 |
| 2.2.8. Альтернативные ключи..... | 57 |

| | |
|--|----|
| 2.2.9. Инвертированные индексы | 57 |
| 2.2.10. Унификация атрибутов..... | 57 |
| 2.2.11. Реализация ссылочной целостности с помощью ERwin | 58 |
| 2.2.12. Хранение информации в модели ERwin..... | 59 |
| 2.3. РАСШИРЕННЫЕ ФУНКЦИИ ERWIN..... | 59 |
| 2.3.1. Обратное проектирование (Reverse engineering) | 59 |
| 2.3.2. Синхронизация с базой данных | 59 |
| 2.3.3. Интерфейсы с СУБД..... | 61 |
| 2.3.4. Поддержка средств 4GL | 62 |
| 2.3.5. Программирование триггеров и процедур | 62 |
| 2.3.6. Правила и начальные значения..... | 63 |
| 2.3.7. Домены | 63 |
| 2.3.8. Генерация отчетов | 65 |
| 2.3.9. Настройка режимов отображения..... | 66 |
| 2.4. ПРЕИМУЩЕСТВА ПРИМЕНЕНИЯ ERWIN..... | 67 |
| 2.5. ОПИСАНИЕ КОМАНД | 68 |
| 2.5.1. Описание команд меню File | 68 |
| 2.5.2. Описание команд меню ModelMart | 69 |
| 2.5.3. Описание команд меню Edit..... | 69 |
| 2.5.4. Описание команд меню Tasks | 69 |
| 2.5.5. Описание команд меню Client | 69 |
| 2.5.6. Описание команд меню Server | 70 |
| 2.5.7. Описание команд меню Option | 71 |
| 2.6. ПРИМЕР РАЗРАБОТКИ МОДЕЛИ В ERWIN..... | 71 |
| 2.7. ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНЫХ И ИНДИВИДУАЛЬНЫХ РАБОТ..... | 79 |
| 2.8. КОНТРОЛЬНЫЕ ВОПРОСЫ | 80 |
| ЛИТЕРАТУРА К ЛАБОРАТОРНОЙ РАБОТЕ № 2..... | 80 |

ЛАБОРАТОРНАЯ РАБОТА № 1.

Структурный анализ и моделирование в среде CASE-средства BPwin

ВВЕДЕНИЕ

Важнейшим этапом современного процесса разработки сложных систем вообще и программного обеспечения в частности является этап системного анализа и моделирования соответствующей предметной области. Данный этап является предпроектным. Его цель заключается в разработке спецификации проекта (технического задания на разработку проекта). От успеха проведения этого этапа зависит успех проекта в целом.

В настоящее время существует ряд методологий, специально предназначенных для упрощения системного анализа и моделирования предметной области. Данные методологии поддерживаются специальными инструментальными средствами автоматизированного анализа, моделирования и разработки сложных систем, получившими название CASE-средств (Computer-Aided Software/System Engineering – компьютерная поддержка проектирования программного обеспечения/систем).

В семействе CASE-средств инструменты для анализа предметной области составляют небольшую часть. Однако именно изучение и моделирование предметной области является наиболее важным этапом при разработке любого приложения, так как позволяет четко и однозначно определить задачи, которые стоят перед разработчиками. Таким образом, использование инструментов анализа и моделирования предметной области должно являться основой начального этапа разработки любой сложной системы, в том числе и программной.

Одним из инструментов системного анализа является CASE-средство верхнего уровня **BPwin (Business Process for Windows)** [4]. Термин Business Process («бизнес-процесс») близок по смыслу к термину «предметная область» и в настоящее время часто заменяет последний. BPwin разработан фирмой LogicWorks. После слияния в 1998г. фирм LogicWorks и PLATINUM technology данное CASE-средство выпускается под логотипом PLATINUM technology [4].

В данном учебном пособии описана одна из последних версий продукта BPwin – версия BPwin 2.5, превосшедшая предыдущие как по функциональности, так и по удобству пользовательского интерфейса.

Основными функциями BPwin являются, во-первых, рисование диаграмм, представляющих собой средства визуального представления отдельных компонентов моделируемой предметной области различных уровней детализации, во-вторых, проверка целостности и согласованности иерархической модели, построенной из диаграмм различных уровней детализации, в-третьих, генерация различного вида отчетов по построенной модели. К достоинствам BPwin следует отнести обеспечение логической четкости в определении и описании элементов диаграмм, проверку целостности связей между диаграммами, локализацию или коррекцию наиболее часто встречающихся ошибок при моделировании [4].

1.1. МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПОДДЕРЖИВАЕМЫЕ ВРWIN

2.1.1. Общие сведения

Цель построения модели некоторого процесса (предметной области) – специфицирование операций и действий, выполняемых в процессе (предметной области), и взаимосвязей между ними. При адекватном построении такая модель обеспечивает полное представление о функционировании исследуемого процесса и обо всех потоках информации и материалов, имеющихся в нем.

ВРwin поддерживает три методологии структурного анализа и моделирования систем - IDEF0, IDEF3 и DFD. В процессе создания модели бизнес-процесса на любой ветви модели можно переключиться на любую из методологий и создать смешанную модель [4].

В IDEF0-модели операция представляет собой процесс преобразования входных материалов или информации в некоторый результат на выходе с использованием ресурсов в виде механизма и при выполнении условий, представленных в виде управления.

Методология DFD включает такие понятия, как *внешняя ссылка* и *хранилище данных*. Это делает её более удобной по сравнению с IDEF0 для моделирования программного обеспечения и систем документооборота.

Методология IDEF3 включает элемент “перекрёсток”, что позволяет описать логику взаимодействия компонентов системы.

Моделирование с использованием всех вышеназванных методологий основано на использовании графических нотаций, основу которых составляют различного вида блоки и соединяющие их дуги.

2.1.2. Методология функционального моделирования IDEF0

Основные понятия IDEF0

При реализации программы интегрированной компьютеризации производства (ICAM), которая финансировалась правительством и военными ведомствами США с середины 70-х годов, были разработаны принципы повышения эффективности производства за счет внедрения компьютерных технологий. В соответствии с проектом ICAM было разработано семейство методологий IDEF (ICAM DEFinition), состоящее из ряда самостоятельных

методологий моделирования различных аспектов функционирования производственной среды или системы [3].

Одной из данных методологий является IDEF0 (Integrated Definition Function Modeling), принятая в настоящее время в качестве федерального стандарта США.

Методология функционального моделирования IDEF0 является подмножеством методологии структурного анализа и проектирования SADT (Structured Analysis And Design Technique), разработанной Дугласом Россом.

При полном SADT-моделировании используются взаимодополняющие модели двух типов [5]:

- *функциональные модели*, выделяющие события (функции, бизнес-процессы) в системе;
- *модели данных*, выделяющие объекты (данные) системы.

В обоих случаях используется один и тот же графический язык блоков и дуг, но блоки и дуги меняются ролями.

Функциональный вариант SADT-методологии в стандартизированной версии правительства США получил название IDEF0.

Методология IDEF0 успешно применяется в самых различных отраслях как эффективное средство анализа, проектирования и представления деловых процессов.

Основной структурной единицей IDEF0-модели является диаграмма, представляющая собой графическое описание модели предметной области или ее части. Главными компонентами IDEF0-диаграммы являются блоки.

Блоки отображают некоторые работы, функции, процессы, задачи, которые происходят или выполняются в течение определённого времени и имеют некоторые результаты. Блоки изображаются в виде прямоугольников. Каждая сторона функционального блока имеет различное назначение (рис.1.1).

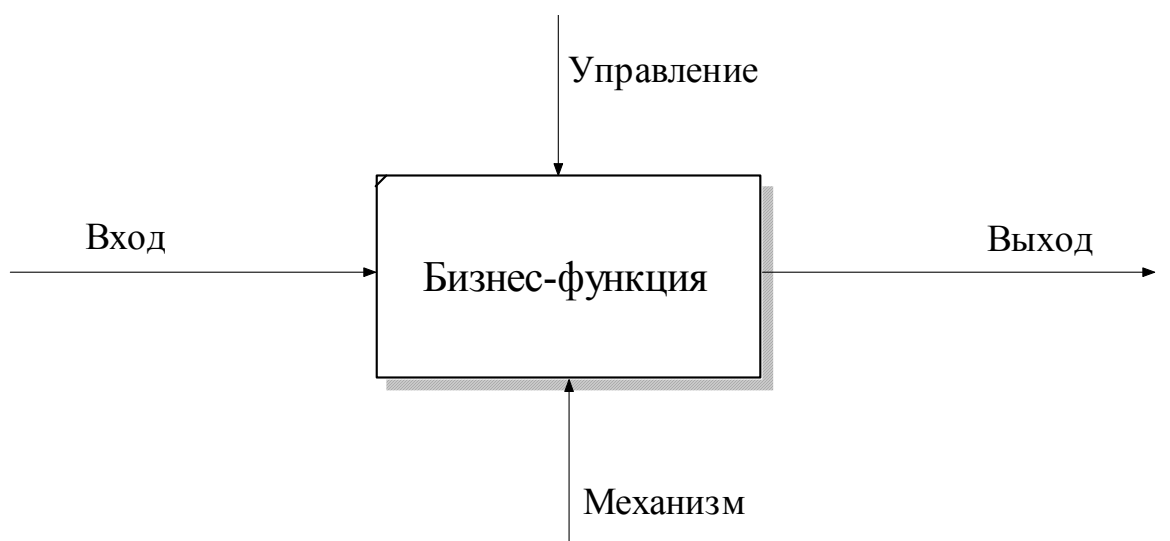


Рис.1.1. Основная конструкция IDEF0-модели

Левая сторона блока предназначена для входа, правая – для выхода, верхняя – для управления, нижняя – для механизмов. Название блока соответствует его функции или выполняемой им работе.

Взаимодействие функций с внешним миром и между собой описывается с помощью дуг (связей), представляемых на диаграммах в виде линий со стрелками (Arrow). В IDEF0 различают *пять типов дуг*.

Вход (Input) – материал или информация, которые используются или преобразуются блоком для получения результата (выхода). Блок может не иметь ни одной входной дуги. Данный вид дуги поступает на левую сторону блока.

Управление (Control) – условия, правила, стратегии, стандарты, которые влияют на выполнение функции. Каждый блок должен иметь *хотя бы одну* дугу управления. Данный вид дуг поступает на верхнюю сторону блока.

Выход (Output) – результат выполнения функции (материал или информация). Каждая функция должна иметь *хотя бы одну* выходную дугу. Данный вид дуг выходит из правой стороны блока.

Механизм (Mechanism) – ресурсы, с помощью которых выполняется работа. Это могут быть, например, денежные средства, персонал предприятия, станки. Данный вид дуг поступает на нижнюю сторону блока.

Вызов (Call) – специальная дуга, указывающая на другую модель предметной области. Данный вид дуги выходит из нижней стороны блока. Дуга вызова не является компонентом собственно методологии SADT. Она является расширением IDEF0-методологии и предназначена для организации коллективной работы над моделью, разделения модели на независимые модели и объединения различных моделей предметной области в одну модель.

Для идентификации граничных дуг используются ICOM-коды (аббревиатура из первых букв типов связей – Input, Control, Output и Mechanism). Граничной дугой называется дуга, выходящая за пределы диаграммы.

Код ICOM содержит префикс, соответствующий типу дуги (I, C, O или M), и порядковый номер, учитывающий положение данной дуги по отношению к родительскому блоку (рис.1.2).

Основополагающими понятиями IDEF0-методологии являются цель моделирования, точка зрения и субъект моделирования.

Точка зрения – это представление о системе с позиции некоторого участника процесса (например, при моделировании некоторого производственного процесса это может быть точка зрения руководителя, технолога, рабочего, экономиста, контролера и т.д.). Точка зрения должна соответствовать цели моделирования.

Цель моделирования определяет степень детализации разрабатываемой модели.

Субъект моделирования определяет границы моделируемой системы и окружающей ее внешней среды.

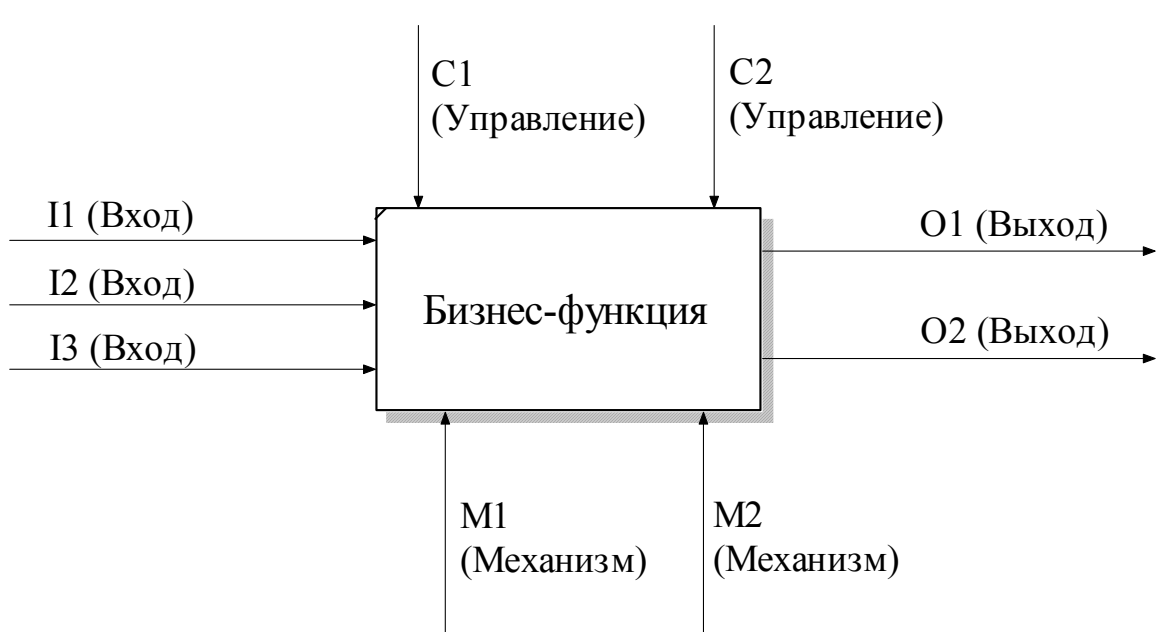


Рис.1.2. Пример ICOM-кодов

IDEF0-модель предполагает наличие чётко сформулированной цели, единственного субъекта моделирования и одной точки зрения.

Принципы моделирования в IDEF0

IDEF0 основана на трех базовых принципах моделирования [4]:

- принципе функциональной декомпозиции;
- принципе ограничения сложности;
- принципе контекста.

Функциональная декомпозиция представляет собой разбиение действий, операций, функций предметной области на более простые действия, операции, функции (на рис.1.3 приведена декомпозиция блока, представленного на рис.1.2). В результате сложная бизнес-функция представляется совокупностью более простых функций, которые в свою очередь также могут быть декомпозированы на более простые функции.

Принцип ограничения сложности обеспечивает понятность и удобочитаемость IDEF0-диаграмм. Он заключается в том, что количество блоков на диаграмме должно быть не менее трех и не более шести (в BPwin допускается от двух до восьми).

Принцип контекстной диаграммы заключается в том, что моделирование предметной области начинается с построения контекстной диаграммы. На этой диаграмме изображается один блок, представляющий собой главную функцию моделируемой системы и определяющий границы системы.

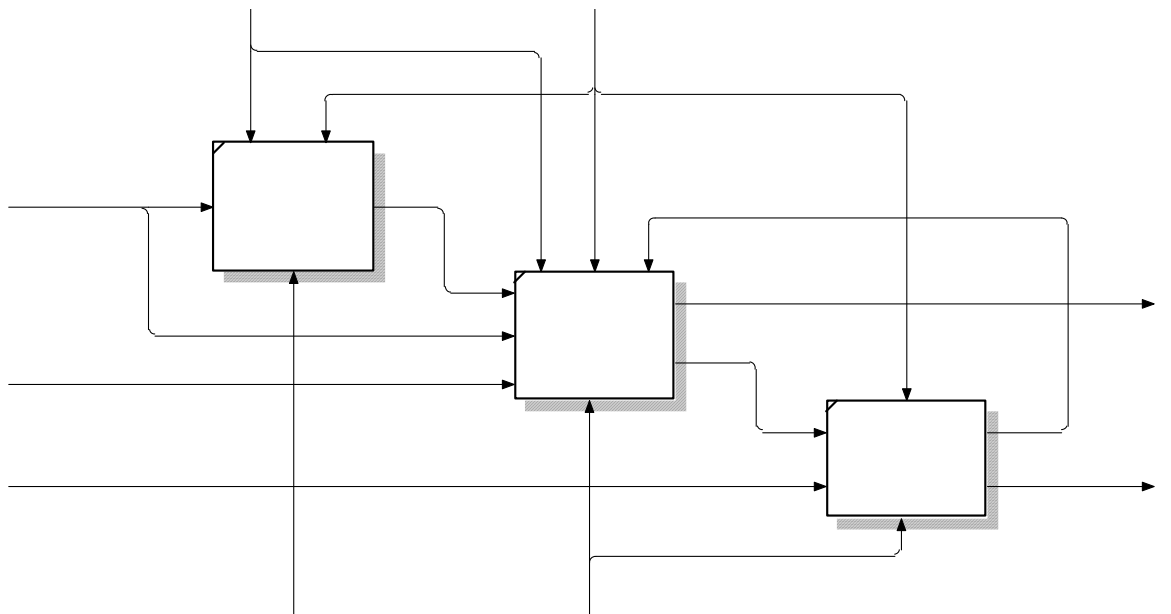


Рис.1.3. Пример декомпозиции

Модель в IDEF0

Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма описывает отдельные компоненты системы и располагается на отдельном листе.

Модель может содержать четыре типа диаграмм [4]:

- контекстную диаграмму;
- диаграммы декомпозиции;
- диаграммы дерева узлов;
- диаграммы только для экспозиции (FEO).

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

После разработки контекстной диаграммы выполняется разбиение ее блока на более мелкие компоненты (*функциональная декомпозиция*). Диаграммы, описывающие каждый компонент и их взаимодействие, называются *диаграммами декомпозиции*.

После декомпозиции контекстной диаграммы проводится декомпозиция каждого большого компонента системы на более мелкие компоненты. Процесс декомпозиции диаграмм повторяется до достижения нужного уровня детализации описания.

Диаграммы, полученные в результате каждого шага декомпозиции, передаются на экспертизу экспертам предметной области. Эксперты оценивают соответствие реальных процессов созданным диаграммам. Найденные несоответствия исправляются автором диаграммы. После прохождения

экспертизы без замечаний выполняется следующий сеанс декомпозиции.

Диаграмма дерева узлов отображает иерархическую взаимосвязь блоков (функций, работ) без описания взаимосвязей между ними. В модели может быть построено произвольное количество диаграмм деревьев узлов, так как их корнем может быть любой блок модели (не обязательно контекстная диаграмма) и они могут быть построены на произвольную глубину.

Диаграммы только для экспозиции (FEO) строятся в основном для справочных целей (например, для иллюстрации отдельных фрагментов модели, для иллюстрации альтернативной точки зрения).

Подробное описание IDEF0-методологии описано в [1]. Процесс создания IDEF0-диаграмм в BPwin приведен в подразд.3.1.

Построение модели AS-IS (как есть) и TO-BE (как будет)

Технология проектирования сложных систем подразумевает, как правило, создание двух видов функциональных моделей предметной области:

- модели существующей организации процесса *AS-IS (как есть)*;
- модели новой организации процесса *TO-BE (как будет)*.

Модель AS-IS позволяет определить неэффективные места существующего на момент моделирования процесса, оценить, насколько глубоким изменениям необходимо подвергнуть существующую структуру организации системы. Признаками неэффективности существующего процесса могут быть, например, бесполезные работы (в работах отсутствует выход), неуправляемые работы (в работах отсутствует управление) и дублирующиеся работы, отсутствие обратных связей по управлению (на проведение процесса не оказывает влияния его результат), входу (материалы или информация используются нерационально) [4].

С учетом анализа найденных в модели AS-IS недостатков создаются модели TO-BE. Модели TO-BE используются для оценки более эффективных способов выполнения процесса в системе. На основе модели TO-BE, отражающей оптимальный способ выполнения процесса, строится прототип, а затем окончательный вариант системы.

2.1.3. Методология DFD

Для представления механизмов передачи и обработки информации в моделируемой системе используются диаграммы потоков данных (DataFlow Diagram - DFD) [2].

Диаграммы DFD наиболее удобно применять для наглядного изображения потоков информации на этапе системного анализа при проектировании программного обеспечения вообще и информационных систем в частности.

Часто диаграммы DFD используют в качестве дополнения к функциональной модели, выполненной в IDEF0, для отражения различных

аспектов предметной области.

В методологии DFD используется четыре графических элемента. В Bpwin для построения диаграмм потоков данных используется нотация Гейна-Сарсона [4].

На рис.1.4 приведен пример диаграммы DFD, построенной с помощью Bpwin.

Основными компонентами диаграмм DFD являются работы (процессы), внешние сущности, хранилища данных и потоки данных.

Работы (процессы). В DFD работы обозначают функции или процессы системы, которые обрабатывают и изменяют информацию (преобразуют входы в выходы). Работы на диаграммах DFD изображаются в виде прямоугольных блоков с закругленными углами (на рис.1.4 блоки «Тестирование программного обеспечения» и «Коррекция программного обеспечения»). Работы не поддерживают связи управления и механизмов как в IDEF0-методологии. В отличие от последней все стороны блоков равнозначны и не имеют конкретного назначения.

Внешние сущности (внешние ссылки). Внешние сущности изображают входы в систему и/или выходы из системы и указывают на место, организацию или человека, которые участвуют в процессе обмена информацией с системой, но располагаются за рамками диаграммы. Одна и та же внешняя сущность может быть использована многократно на одной или нескольких DFD-диаграммах. Внешние сущности изображаются прямоугольниками с тенью (на рис.1.4 блоки «Группа тестирования» и «Группа программистов»).

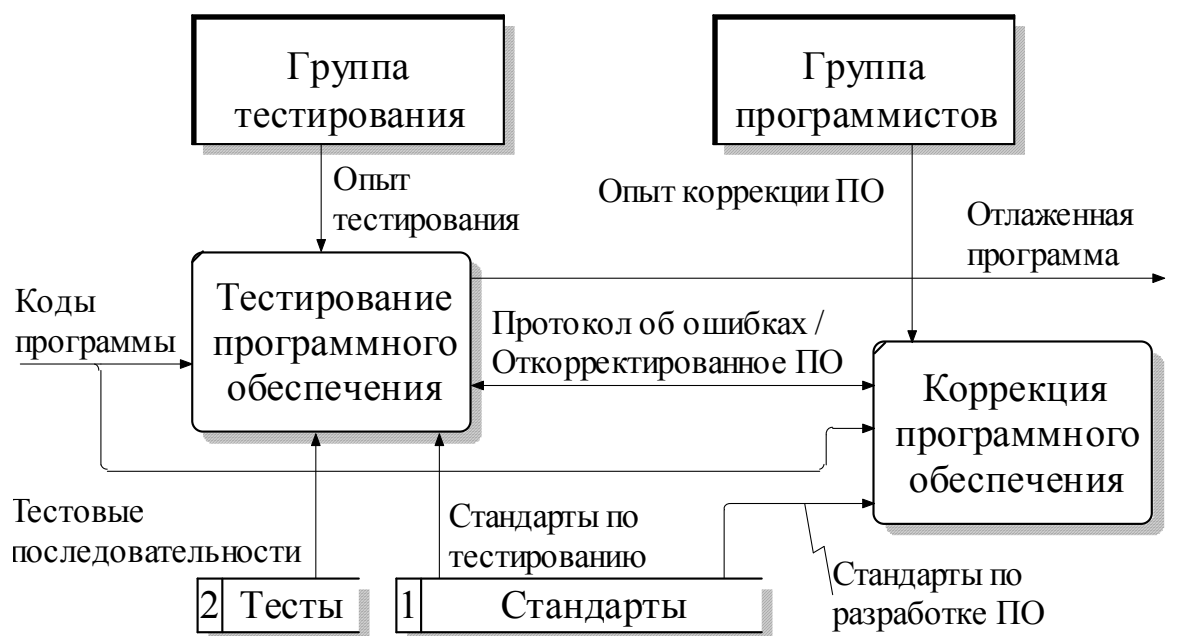


Рис.1.4. Пример диаграммы DFD

Потоки данных. Поток данных определяет качественный характер информации, передаваемой от источника к приемнику. Потоки данных на

диаграммах DFD изображаются линиями со стрелкой на одном из ее концов или на обоих концах. Стрелка показывает направление информационного потока в системе.

Хранилища данных. Хранилища данных представляют собой собственно данные, к которым осуществляется доступ. Эти данные могут быть созданы или изменены работами. В отличие от потоков данных, описывающих информацию в движении, хранилища данных изображают информацию в покое. Хранилища данных на диаграммах DFD изображаются прямоугольными блоками с двумя полями (на рис.1.4 блоки «Стандарты» и «Тесты»). В левом поле указывается номер или идентификатор хранилища. На одной диаграмме может присутствовать несколько копий одного и того же хранилища данных.

Диаграммы потоков данных дают четкое представление о том, какие данные используются и какие функции выполняются существующей информационной системой или моделируемым программным обеспечением. При этом может оказаться, что существующие потоки информации реализованы ненадежно и нуждаются в реорганизации.

Особенности создания диаграмм DFD в BPwin описаны в подразд.3.2.

2.1.4. Методология IDEF3

Наличие в диаграммах DFD элементов для описания источников, приемников и хранилищ данных позволяет описать процесс движения информации в системе.

Для описания логики взаимодействия информационных потоков модель системы дополняют диаграммами методологии IDEF3. Диаграммы данного вида называются диаграммами потоков работ (WorkFlow Diagram). Методология моделирования IDEF3 позволяет графически описать течение процессов во времени и отношения процессов и объектов, являющихся частями этих процессов [2].

В методологии IDEF3 существует два типа моделей [4]:

- модель, отражающая процессы в их логической последовательности и позволяющая увидеть функционирование системы;
- модель, отражающая “сеть переходных состояний объекта” и позволяющая увидеть последовательность состояний, в которых может оказаться объект при прохождении через определенный процесс.

С помощью диаграмм IDEF3 можно анализировать динамику событий из реальной жизни, например, какие действия должны выполнять различные сотрудники университета во время вступительных экзаменов или во время учебной воздушной тревоги и т.п.

Пример диаграммы IDEF3, построенной в BPWin, приведен на рис.1.5.

IDEF3-модель может содержать следующие элементы.

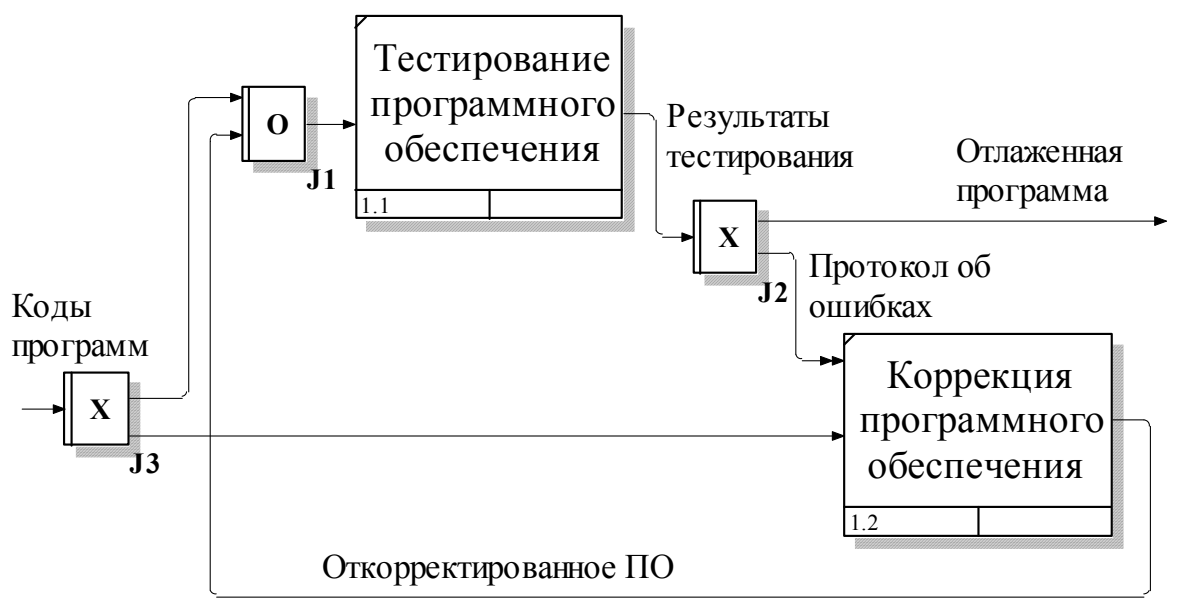


Рис.1.5. Пример диаграммы IDEF3

Единицы работы (Unit of Work - UOW) - основной компонент диаграммы IDEF3, близкий по смыслу к функциональному блоку IDEF0. Единицы работы представляются в виде прямоугольника, вид которого приведен на рис.1.6.

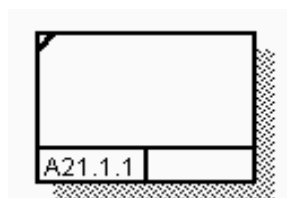


Рис.1.6. Единица работы в IDEF3

Единицы работы называются также работами (Activity). Имя работы записывается в центральном поле прямоугольника.

В левом нижнем углу прямоугольника записывается идентификатор работы. Он присваивается при создании модели и не изменяется в процессе моделирования. Идентификатор работы обозначается иерархическим номером. В его состав в наиболее полном варианте входит символ A (Activity), номер родительской работы, номер декомпозиции, номер работы в пределах данной декомпозиции (например, на рис.1.6 номер работы A21.1.1 соответствует номеру 21 родительской работы, номеру 1 декомпозиции, номеру 1 работы на данной декомпозиции).

Связи (Arrow Links) – изображаются линиями со стрелками и показывают взаимоотношения работ. Все связи являются однонаправленными и

могут быть направлены в любую сторону. Предпочтительнее направление связей слева направо или сверху вниз.

В IDEF3 различают три типа связей:

- *связь предшествования (Precedence)* – связывает единицы работ и обозначает, что, прежде чем начнется работа-приемник, должна завершиться работа-источник; изображается сплошной линией со стрелкой;
- *связь отношения (Relational Link)* - показывает связь между двумя единицами работ (UOW) или между единицей работы и объектом ссылки; обозначается пунктирной линией;
- *поток объектов (Object Flow)* – показывает участие некоторого объекта в двух или более единицах работ (например, объект производится в ходе выполнения одной работы и используется в другой работе); обозначается сплошной линией с двумя стрелками (на рис.1.5 связь «Протокол об ошибках»).

Перекрестки (Junctions) – используются, чтобы показать разветвления и альтернативные пути развития процесса, которые могут возникнуть во время его выполнения. На диаграммах IDEF3 связи могут сливаться и разветвляться только через перекрёстки.

Различают два типа перекрестков:

- *перекресток слияния (Fan-in Junction)* – узел, собирающий несколько связей в одну; указывает на необходимость условия завершения работ (источников связей) для продолжения процесса (на рис.1.5 обозначен прямоугольником с буквой **O**);
- *перекресток разветвления (Fan-out Junction)* – узел, в котором единственная входящая в него связь разветвляется; показывает, что работы, следующие за перекрестком, выполняются параллельно или альтернативно (на рис.1.5 обозначены прямоугольником с буквой **X**).

Объекты ссылок (Referents) - служат для отображения некоторых идей или концепций без использования специальных графических элементов, таких как стрелки, перекрестки или работы.

Особенности создания IDEF3-диаграмм в BРwin описаны в подразд.3.3.

1.2. ИНСТРУМЕНТАЛЬНАЯ СРЕДА ВРWIN 2.5

2.1.5. Интегрированная среда разработки

Врwin имеет достаточно простой интерфейс пользователя, позволяющий создавать сложные модели при минимальных усилиях.

При запуске Врwin по умолчанию появляются главное меню, основная панель инструментов, палитра инструментов и навигатор модели Model

Explorer [4]. На рис.1.7 представлена интегрированная среда разработки модели BPwin2.5.

Вид палитры инструментов (BPwin Toolbox на рис.1.7) зависит от выбранной методологии (IDEF0, IDEF3 или DFD). Компоненты палитры, появляющиеся при выборе конкретной нотации, описываются ниже по ходу изложения материала.

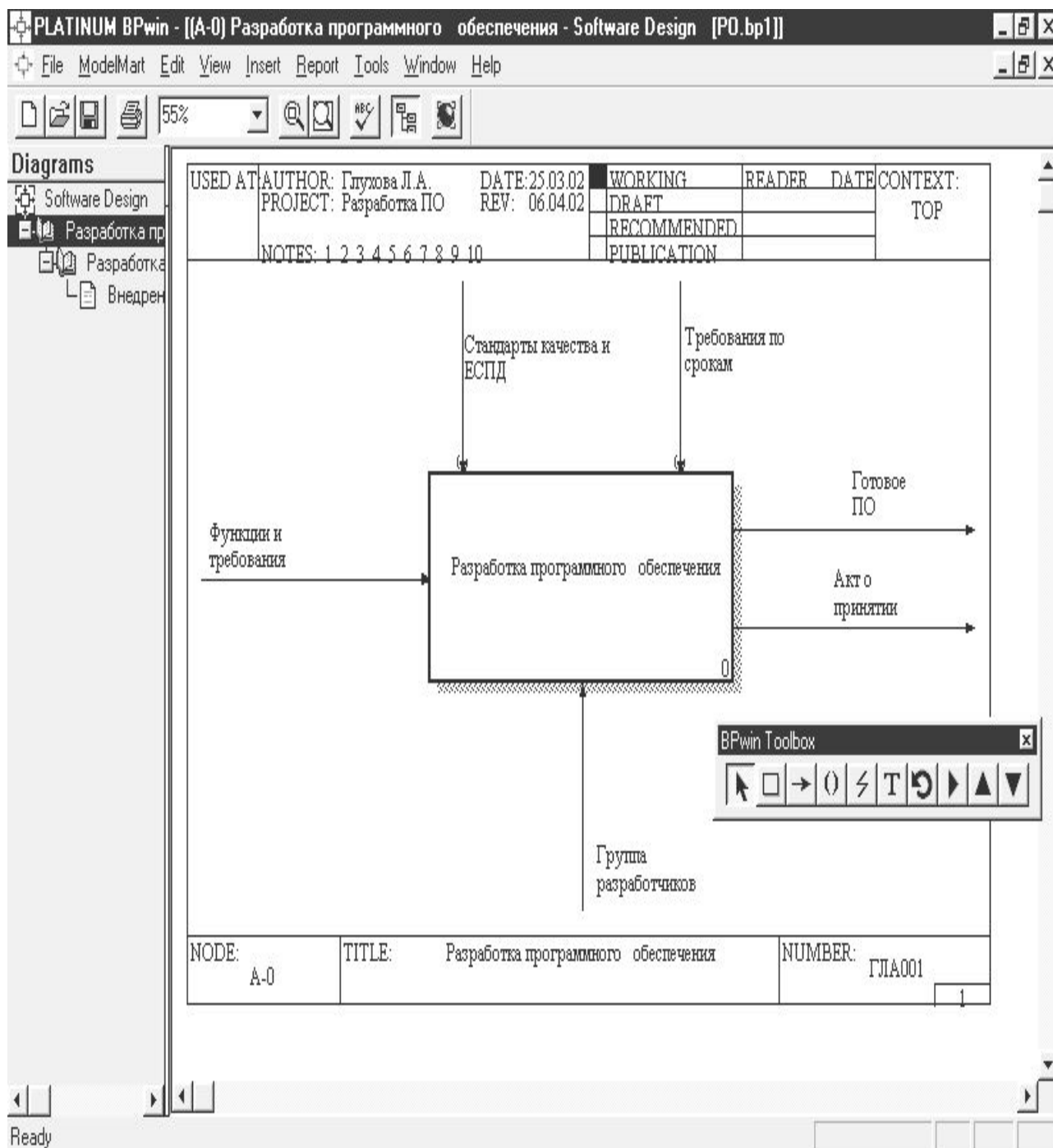


Рис.1.7. Интегрированная среда разработки модели BPwin2.5

Большинство элементов управления на панели инструментов (рис.1.8) имеют стандартные вид и назначение, принятые в современных прикладных программах, и не требуют дополнительного описания. Это (на рис.1.8 слева направо) кнопки «Создать новую модель», «Открыть модель», «Сохранить модель», «Напечатать модель», элемент «Выбор масштаба», две кнопки «Масштабирование», кнопка «Проверка правописания».

Назначение и вид специфических кнопок приведены в табл.1.

Действия, выполняемые с помощью элементов управления, могут быть выполнены также при выборе соответствующих пунктов главного меню.

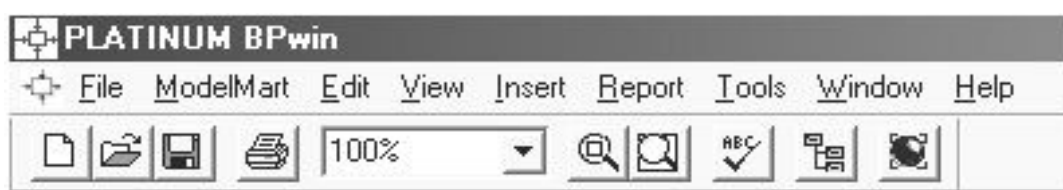




Рис.1.8. Главное меню и основная панель инструментов

Таблица 1

Специфические кнопки основной панели инструментов

| Кнопка | Назначение |
|---|---|
|  | Включение и выключение навигатора модели ModelExplorer (соответствующий пункт меню View/Model Explorer) |
|  | Включение и выключение панели инструментов работы с ModelMart (соответствующий пункт меню Model Mart) |

2.1.6. Описание команд и пунктов главного меню

Описание команд меню File

В состав данного меню входят следующие команды и пункты:

- New – Создать новую модель;
- Open – Открыть существующую модель;
- Close – Закрыть модель;
- Save – Сохранить открытую модель;
- Save as - Сохранить модель под новым именем;
- Save all – Сохранить все открытые модели;
- Print – Печать диаграммы;

- Print Setup – Настройка принтера;
- Page Setup – Установка размеров страницы;
- Export – Экспорт;
- Import – Импорт;
- Exit – Выход.

Пункт меню Page Setup предназначен для установки размеров полей стандартного бланка диаграммы. Данный пункт содержит подпункты установки размеров полей для текущей диаграммы, для новой диаграммы и для новой модели. На рис.1.9 приведено диалоговое окно установки размеров полей стандартного бланка диаграммы для новой модели. Диалоговое окно позволяет установить размеры бланка диаграммы и размеры его служебных полей.

Пункты меню Export и Import позволяют экспортировать или импортировать словарь работ, словарь связей, диаграмму дерева работ, данные модели и т.п. из текущей модели в другую или наоборот.

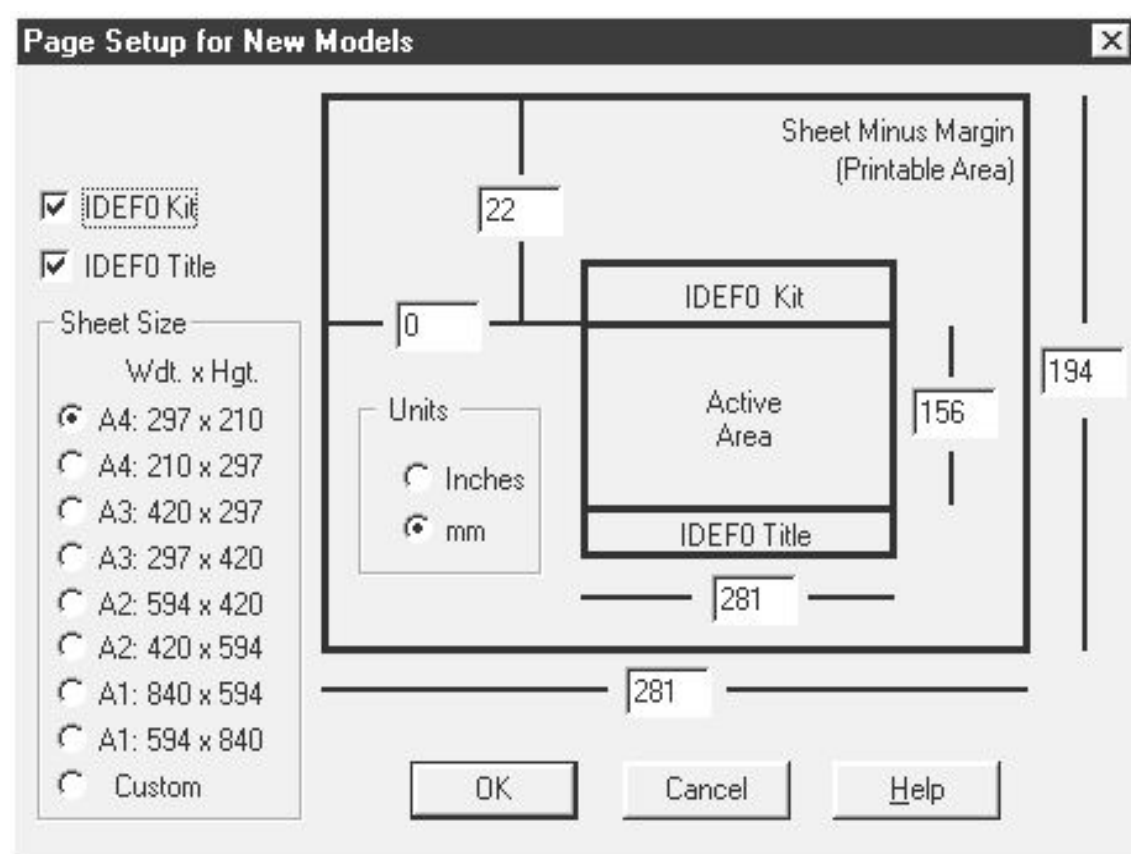


Рис.1.9. Диалоговое окно установки размеров полей стандартного бланка диаграммы для новой модели

Описание команд меню ModelMart

Команды данного меню используются для передачи/принятия модели в/из репозитория ModelMart.

Описание команд меню Edit

В состав данного меню входят следующие основные группы команд и пунктов.

1. Группа пунктов редактирования свойств существующей или создаваемой модели или ее компонент; в состав данной группы входят пункты:

- Diagram Properties (свойства диаграммы) - позволяет установить название диаграммы, ее автора, С-номер, статус диаграммы, дату создания и комментарий к диаграмме;
- Model Properties (свойства модели) - позволяет установить название модели, ее автора, цель, точку зрения, статус, правила записи номера узла, требования к представлению модели, требования к изменению размеров блоков при их редактировании;
- Diagram Object Dictionary (словарь объектов диаграмм) – содержит перечень всех объектов модели (работ для IDEF0, единиц работ для IDEF3) и позволяет добавлять, удалять и проводить иные действия по редактированию объектов и их свойств;
- Arrow Dictionary (словарь стрелок) - содержит перечень всех связей модели и позволяет проводить различные действия по редактированию связей и их свойств;
- Arrow Style (стиль стрелок) – позволяет устанавливать стиль стрелок.

2. Группа пунктов редактирования информационной модели, созданной в среде ERwin и связанной с разработанной функциональной моделью предметной области; в состав данной группы входят пункты:

- Entity/Attribute Dictionary (словарь сущностей/атрибутов);
- Arrow Data (данные взаимодействующих связей).

3. Команды редактирования диаграмм и их элементов:

- Copy Picture - копировать диаграмму в буфер обмена;
- Cut/Delete - вырезать и поместить в буфер выделенную работу или удалить группу выделенных объектов;
- Move/Rename - переслать или переименовать выделенную работу;
- Paste - вставить работу из буфера обмена.

4. Группа пунктов перехода:

- Go To Activity - перейти к заданной работе;

- Go To Diagram - перейти к заданной диаграмме;
- Delete Diagram – удалить диаграмму;
- Redraw Diagram - перерисовывать диаграмму.

Описание команд меню View

Команды данного меню дают возможность изменения визуального изображения диаграмм и пользовательского интерфейса.

Описание пунктов меню Insert

Данное меню содержит следующие пункты:

- Node Tree - создать дерево узлов;
- FEO Diagram - создать FEO-диаграмму текущей диаграммы.

Описание пунктов меню Report

Пункты данного меню предназначены для управления созданием семи видов отчетов, отражающих результаты проектирования модели, а также для установки шрифтов отчетов и установки параметров печати отчетов. Подробнее виды отчетов будут описаны в разд.4.

Описание пунктов и команд меню Tools

В состав данного меню входят следующие пункты и команды:

- Preferences – установка привилегированных свойств модели;
- New Model Settings – установка свойств новой модели; данные свойства аналогичны свойствам существующей модели, описанным в пункте Model Properties меню Edit;
- Auto Save – установка параметров автосохранения;
- Spelling – контроль правописания текущей модели;
- Spelling Options – установка параметров контроля правописания;
- Merge Model Dictionary – слияние словарей работ и стрелок при слиянии моделей;
- Default Fonts – установка шрифтов для описания компонентов диаграмм, свойств модели, отчетов, а также для различных полей бланка диаграммы; содержит пункты:
 - Context Activity – работа на контекстной диаграмме;
 - Context Arrow – стрелки на контекстной диаграмме;
 - Decomposition Activity – работы на диаграмме декомпозиции;
 - Decomposition Arrow – стрелки на диаграмме декомпозиции;
 - NodeTree Text – текст на диаграмме дерева узлов;
 - Frame User Text – текст, вносимый пользователем в поля бланка диаграмм;

- Frame System Text – системный текст на бланке диаграмм;
- Text Blocks – текстовые блоки;
- Parent Diagram Text – текст на родительской диаграмме;
- Parent Diagram Text – текст заголовка родительской диаграммы;
- Report Text – текст отчетов.

1.3. СОЗДАНИЕ МОДЕЛЕЙ В ИНСТРУМЕНТАЛЬНОЙ СРЕДЕ BPWIN 2.5

2.1.7. Создание IDEF0-модели

Начальные этапы создания модели

При создании новой модели возникает диалог, приведенный на рис.1.10. В данном диалоге необходимо указать, создается новая модель или она открывается из файла либо из репозитория ModelMart, внести имя новой модели и выбрать методологию, в которой она будет строиться.

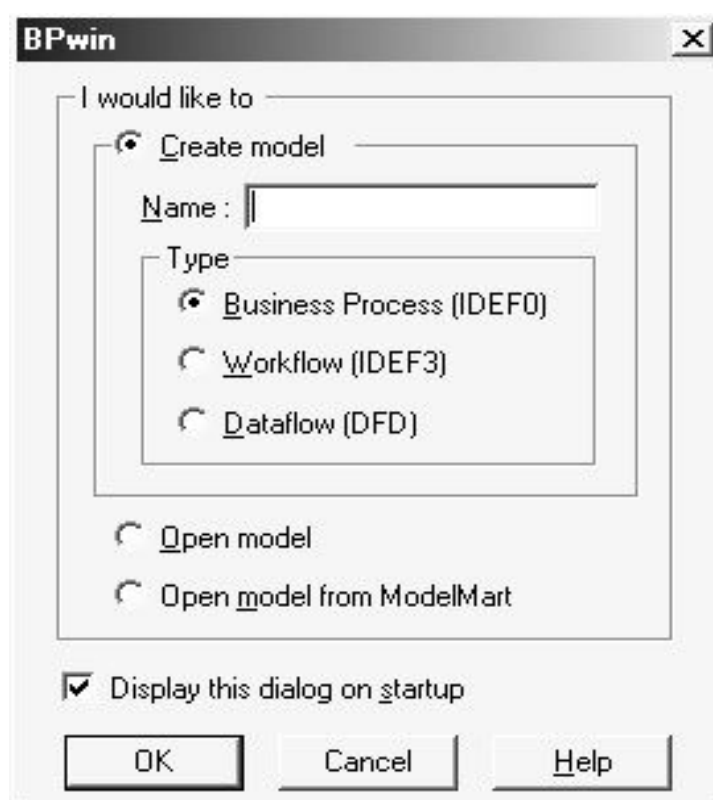


Рис.1.10. Диалоговое окно создания модели

BPwin поддерживает три методологии - IDEF0, IDEF3 и DFD. Каждая из них решает свои задачи. В BPwin возможно построение смешанных моделей: модель может содержать одновременно любые из диаграмм IDEF0, IDEF3 и DFD [4]. Состав палитры инструментов (BPwin Toolbox на рис.1.7) изменяется автоматически, когда происходит переключение с одной методологии на другую.

Как уже отмечалось, основополагающими понятиями IDEF0-методологии являются цель моделирования, точка зрения и субъект моделирования (предметную область).

Для внесения субъекта, цели и точки зрения модели IDEF0 в BPwin необходимо выбрать пункт меню *Edit / Model Properties* (Свойства модели), вызывающий диалог Model Properties. В закладке Purpose (рис.1.11) следует указать цель и точку зрения.

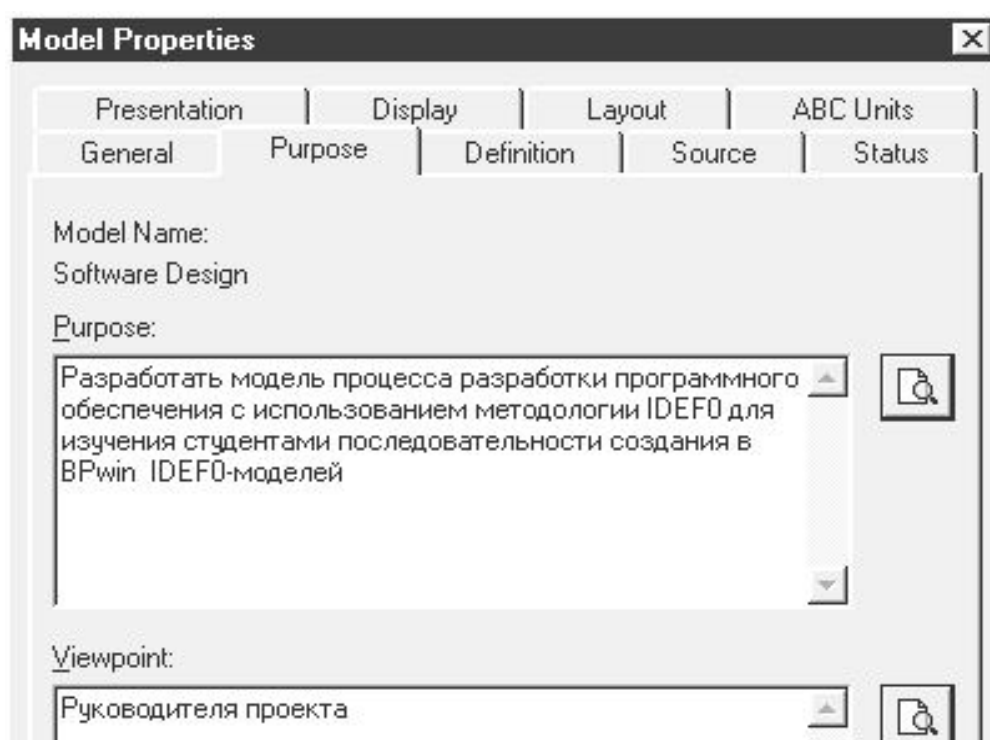


Рис.1.11. Закладка Purpose диалога Model Properties

В закладке Definition (рис.1.12) необходимо определить субъект моделирования (Definition) и его границы (Scope). В закладке Status определяется статус модели (черновой, рабочий, окончательный и т.д.), время создания или последнего редактирования. В закладке Source (рис.1.13) описываются источники информации для построения модели. Закладка General служит для внесения имени проекта и модели, фамилии и инициалов автора и вида модели - AS-IS или TO-BE (рис.1.14).

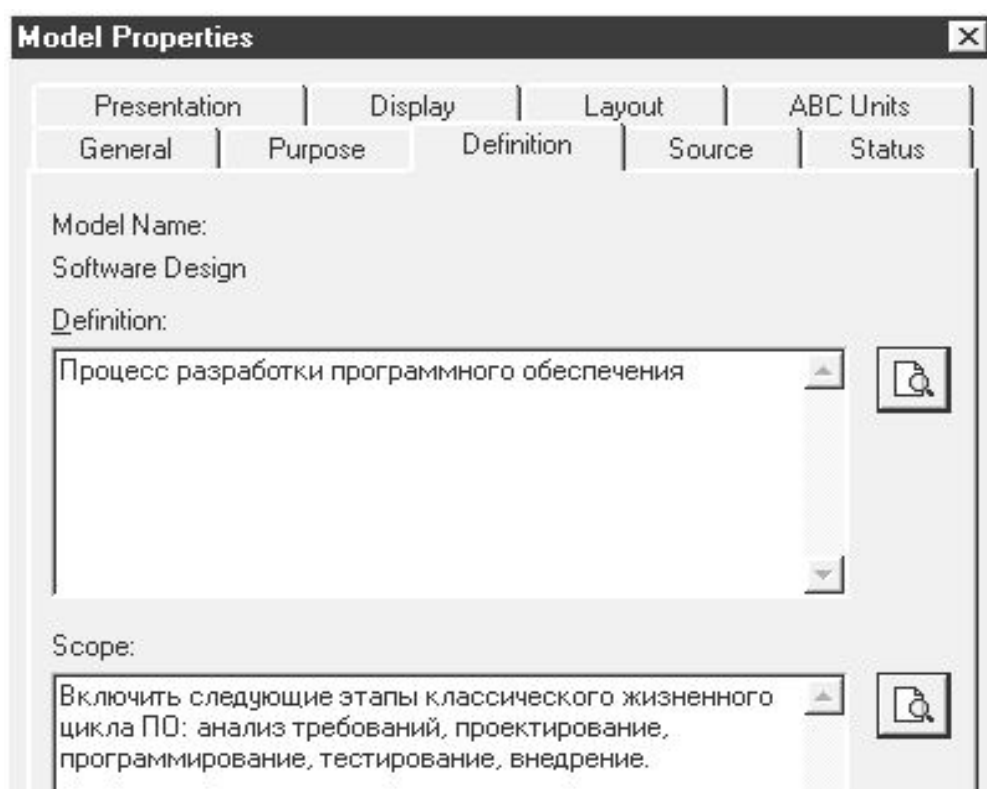


Рис.1.12. Закладка Definition диалога Model Properties



Рис.1.13. Закладка Source диалога Model Properties

Рис.1.14. Диалог свойств модели

Создание контекстной IDEF0-диаграммы

Если в диалоге создания модели (см. рис.1.10) выбрана методология IDEF0, то на рабочей панели инструментов появится палитра инструментов, содержащая команды применительно к IDEF0 (рис.1.15). Кнопки данной палитры предназначены для создания компонентов IDIF0-диаграмм (работ, связей, их названий и др.). Назначение кнопок приведено на рис.1.15.

Основными компонентами диаграмм IDEF0 являются функции или работы (Activity) и связи (Arrow) между ними.

При создании новой модели (пункт меню File/New) автоматически создаётся контекстная диаграмма с единственной работой, изображающая систему в целом (рис.1.16). Для внесения имени работы следует в ее рабочей области щёлкнуть правой кнопкой мыши, выбрать в возникшем контекстном меню (рис.1.17) пункт Name Editor и в появившемся диалоге внести имя работы и фамилию автора диаграммы (рис.1.18). Имя работы должно быть основано на использовании отглагольного существительного, обозначающего действие (например, «Вычисление результата», «Определение условий», «Обработка события»).

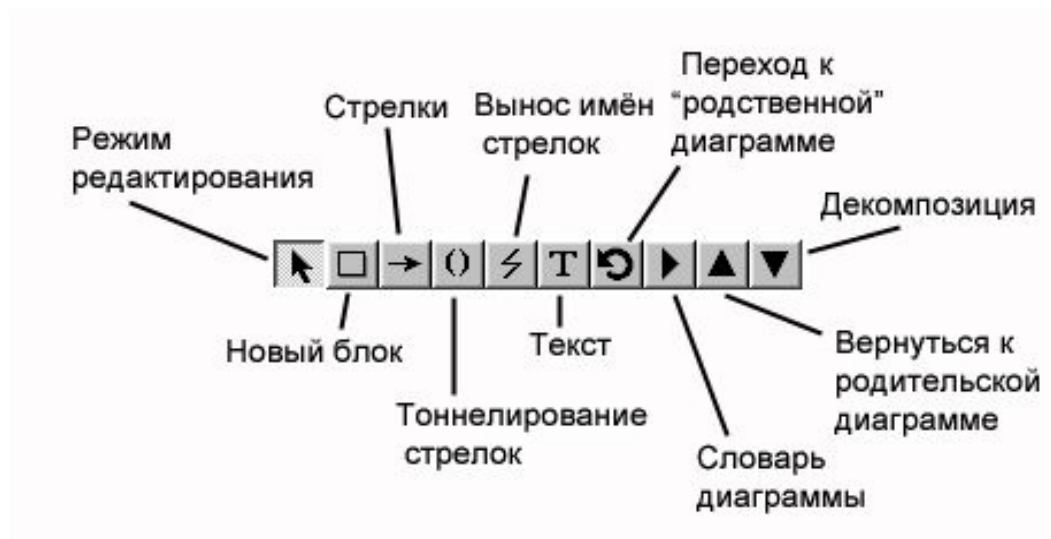


Рис.1.15. Палитра инструментов для IDEF0-методологии

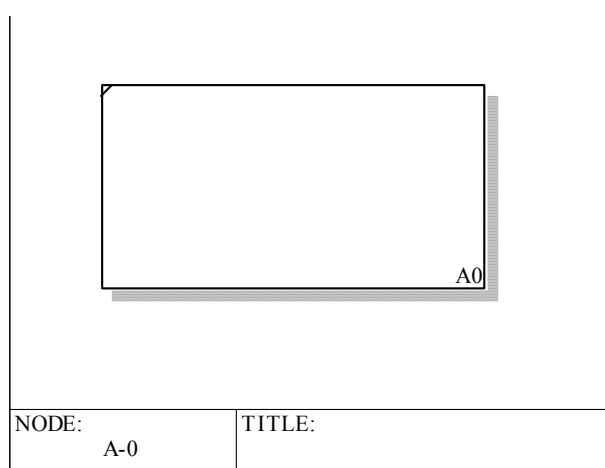


Рис.1.16. Автоматически создаваемый шаблон контекстной диаграммы

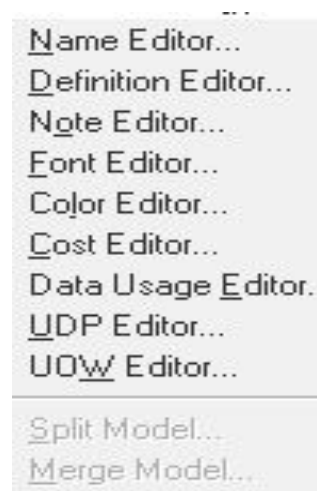


Рис.1.17. Контекстное меню редактирования объекта

Чтобы не возникло проблем с переходом к русской раскладке клавиатуры, следует в рабочей области работы предварительно щелкнуть правой кнопкой мыши, в появившемся контекстном меню выбрать пункт Font Editor и затем шрифт Times New Roman.

Для описания других свойств работы следует использовать пункт Definition Editor (редактор документирования объекта) контекстного меню, при выборе которого появляется закладка Definition диалога Activity Properties (см. рис.1.17, 1.18).

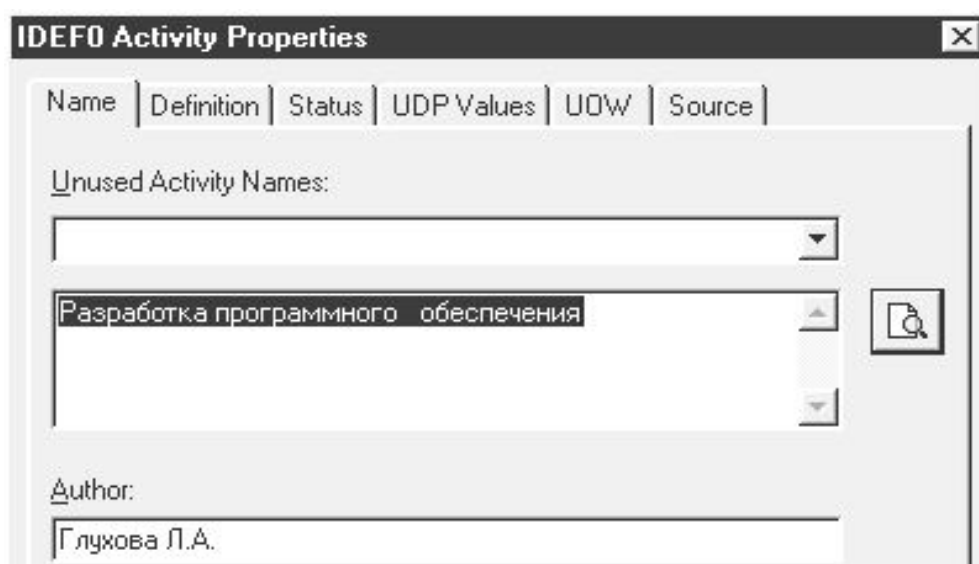


Рис.1.18. Закладка Name редактора свойств работы

Дуги на контекстной диаграмме описывают взаимодействие моделируемого процесса с окружающей средой. Они начинаются или заканчиваются у границы диаграммы. Такие дуги называются *граничными*.

Для создания граничной входной дуги необходимо [4]:

- щелкнуть по кнопке с символом стрелки (режим рисования стрелок) в палитре инструментов (см. «Стрелки» на рис.1.15), перенести курсор к левой стороне экрана до появления левой границы диаграммы, выделенной полосой;
- щелкнуть один раз по левой границе диаграммы (отмечается место, откуда выходит стрелка), затем по левой границе работы (отмечается место, где заканчивается стрелка);
- вернуться в палитру инструментов и для присваивания стрелке названия выбрать в палитре инструментов режим редактирования (см. рис.1.15);
- щелкнуть правой кнопкой мыши на линии стрелки, во всплывающем меню редактирования связей (рис.1.19) выбрать пункт Name Editor и в появившемся диалоге IDEF0 Arrow Properties внести имя дуги и фамилию автора диаграммы; основу названия дуги на IDEF0-диаграммах должно составлять существительное (например, «Код программы», «Студент», «Результаты», «График выполнения»).

Следует отметить, что большинство пунктов всплывающего меню редактирования связей аналогично рис.1.17, а большинство закладок диалогового окна IDEF0 Arrow Properties аналогично рис.1.18.

Для создания граничных стрелок выхода, управления и механизмов следует выполнить аналогичные действия с учетом правил расположения таких стрелок по отношению к работе и к границам диаграммы (стрелка выхода

начинается на правой стороне работы и заканчивается на правой границе диаграммы, стрелка управления начинается на верхней границе диаграммы и заканчивается на верхней стороне работы, стрелка механизмов начинается на нижней границе диаграммы и заканчивается на правой стороне работы).



Рис.1.19. Контекстное меню редактирования связей

При желании удалить созданную стрелку необходимо щелкнуть левой кнопкой мыши на линии стрелки и нажать клавишу Del.

В результате выполненных действий будет создана контекстная диаграмма, приведенная на рис.1.20.

Создание диаграмм декомпозиции

Диаграммы декомпозиции содержат родственные работы, т.е. дочерние работы, имеющие общую родительскую. Для создания диаграммы декомпозиции необходимо левой кнопкой мыши выделить родительскую работу и щёлкнуть по кнопке «Декомпозиция» палитры инструментов (см. рис.1.15).

В результате возникает диалог Activity Box Count (рис.1.21), в котором следует указать нотацию новой диаграммы и количество работ на ней.

В итоге будет получена диаграмма декомпозиции (рис.1.22), содержащая пять работ и несвязные стрелки. Несвязными стрелками являются дуги, касающиеся декомпозированного блока родительской диаграммы.

Для связывания граничных стрелок входа, управления или механизма с соответствующими работами необходимо перейти в режим редактирования стрелок (кнопка «Режим редактирования» на рис.1.15), щелкнуть по наконечнику стрелки и щелкнуть по соответствующей стороне работы.

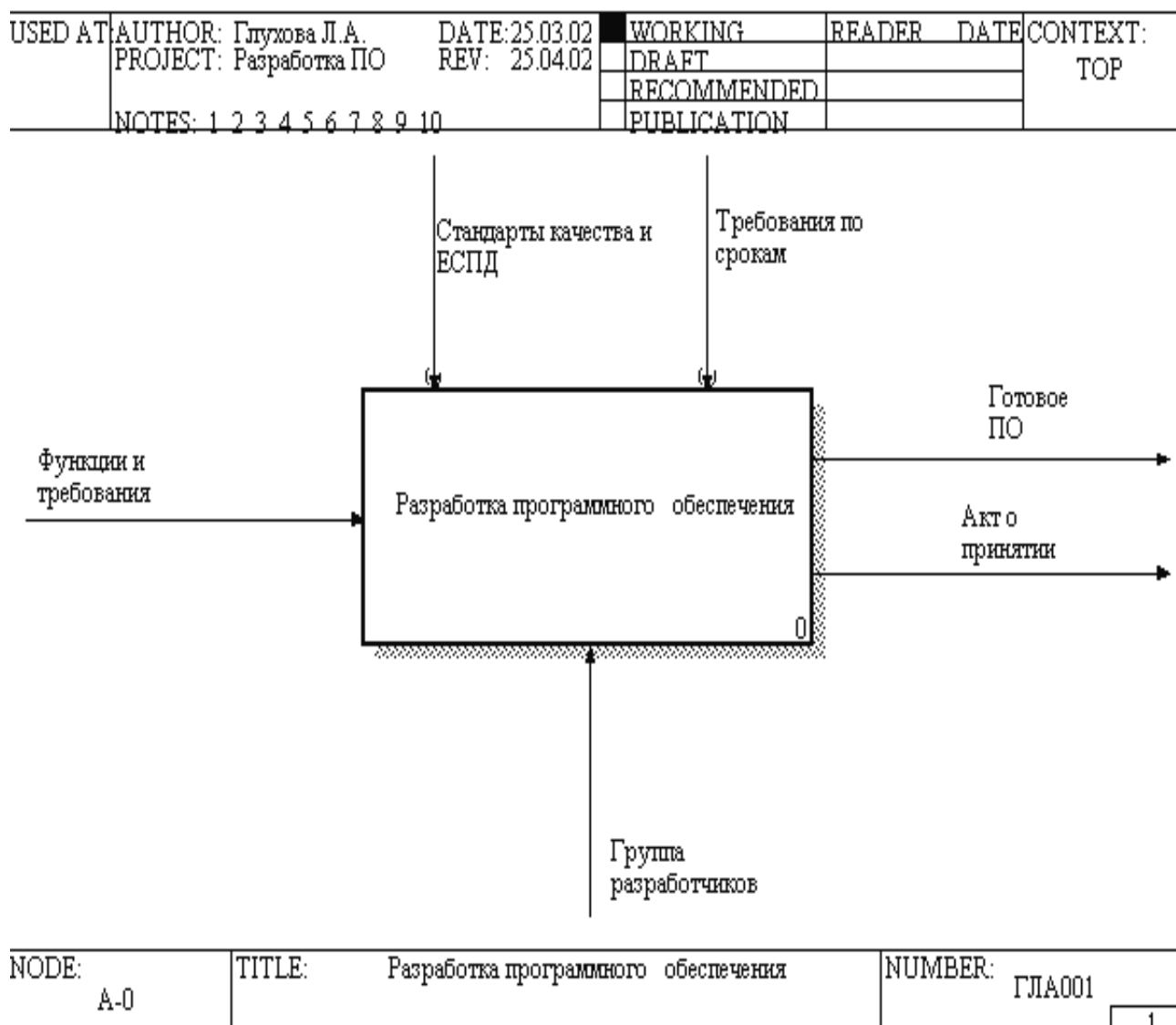


Рис.1.20. Контекстная диаграмма

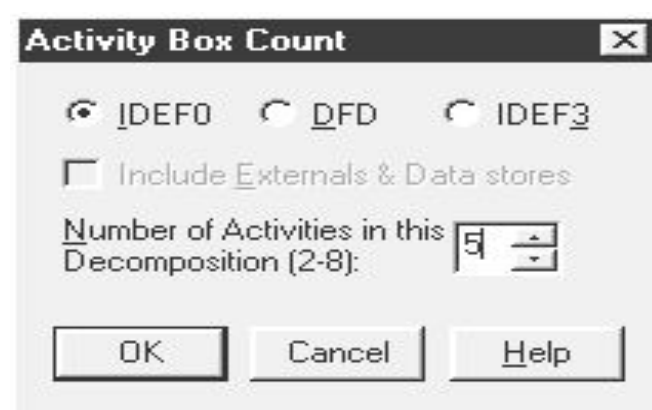


Рис.1.21. Диалог Activity Box Count

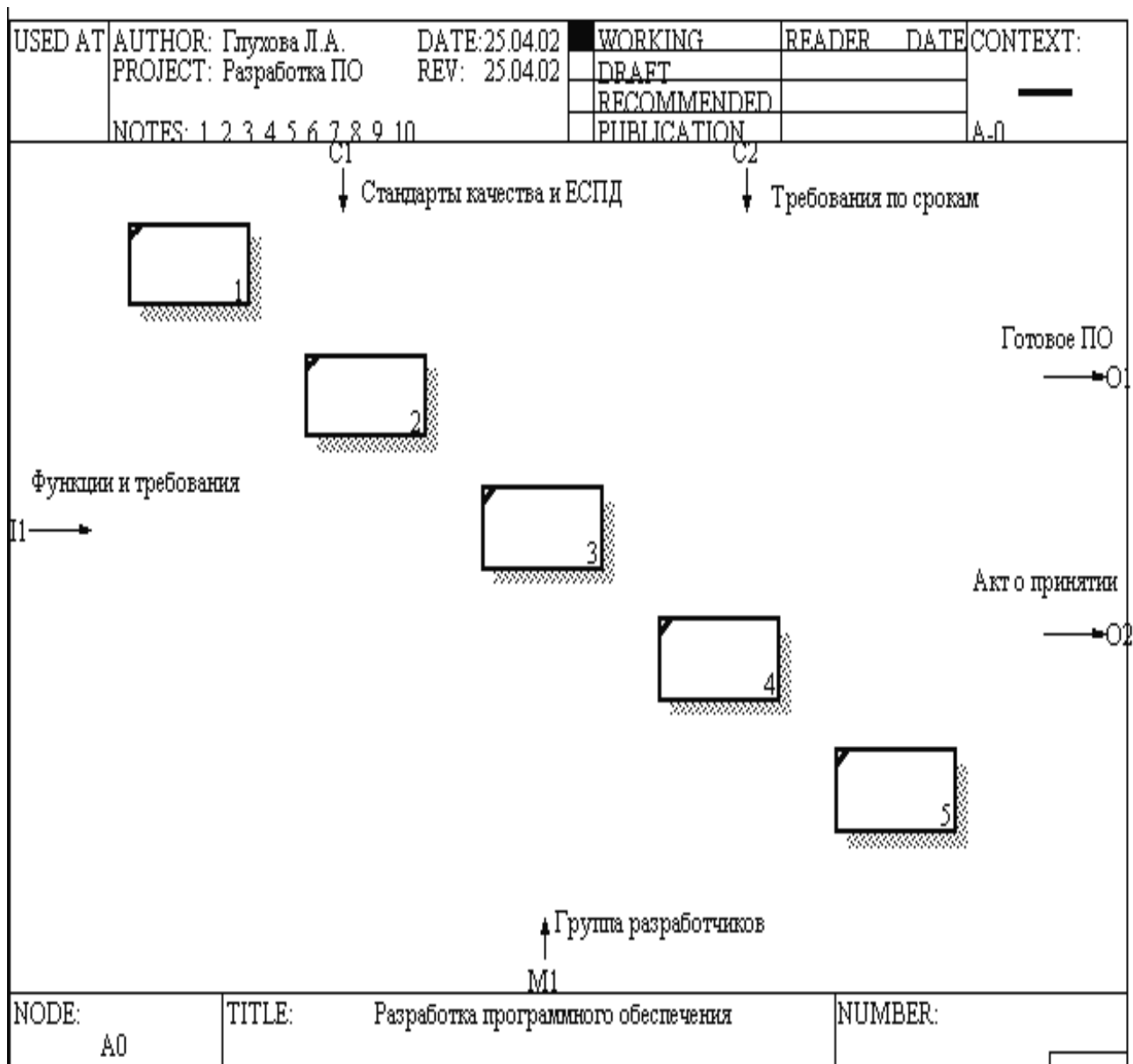


Рис.1.22. Диаграмма декомпозиции с несвязными стрелками

Для связывания граничных стрелок выхода с соответствующими работами необходимо в режиме редактирования стрелок щелкнуть по соответствующей стороне работы и затем по наконечнику стрелки.

Для связи работ между собой используются внутренние стрелки, т.е. стрелки, начинающиеся у одной и кончающиеся у другой работы и не касающиеся границ диаграммы.

Для рисования внутренней стрелки необходимо в режиме рисования стрелок (кнопка «Стрелки» на рис.1.15) щелкнуть по стороне выхода работы-источника стрелки и затем по соответствующей стороне (входа, управления или

механизмов) работы-приемника стрелки.

Для разветвления стрелки следует в режиме рисования стрелок (кнопка «Стрелки» на рис.1.15) щелкнуть по сегменту стрелки, которую нужно разветвить, и затем по соответствующей стороне (входа, управления или механизмов) работы-приемника ветви стрелки.

Для слияния стрелок следует в режиме рисования стрелок (кнопка «Стрелки» на рис.1.15) щелкнуть по стороне выхода работы-источника ветви стрелки и затем по сегменту стрелки, которую нужно слить с ветвью.

Для удаления блока (стрелки) необходимо его (ее) выделить с помощью мыши и нажать на клавишу Del.

Для идентификации граничных стрелок диаграммы используются коды ICOM. Для отображения кодов ICOM на диаграммах модели необходимо включить опцию Show ICOM Codes на закладке Presentation диалогового окна Molel Properties. Данное окно может быть вызвано с помощью меню Edit/Model Properties или нажатием правой кнопки мыши в свободной области диаграммы.

Если в результате разработки диаграммы декомпозиции окажется, что в нее необходимо добавить работу, то для этого следует войти в режим рисования работ (кнопка «Работа» на рис.1.15) и щелкнуть в нужном месте рабочего поля диаграммы левой кнопкой мыши. В результате в выбранном месте диаграммы появится новая работа.

Для присвоения работам и стрелкам имен необходимо выполнить действия, аналогичные описанным для контекстной диаграммы.

В некоторых случаях удобно использовать механизм тоннелирования связей. Существует два вида тоннелирования связей: со скрытым приемником и со скрытым источником.

Связь со скрытым приемником удобно использовать, если связь, входящую в родительский блок, нежелательно изображать на диаграмме декомпозиции. Например, если граничная связь управления или механизмов поступает на все работы диаграммы, она может быть неинформативной и будет лишь загромождать диаграмму. Связи со скрытым приемником изображаются стрелками с круглыми скобками вокруг наконечника стрелки (см. связи «Стандарты качества и ЕСПД» и «Требования по срокам» на рис.1.20).

Связь со скрытым источником удобно использовать, если на диаграмме декомпозиции нужно показать граничную связь, которой не было на родительской диаграмме, поскольку для родительской диаграммы данная связь является несущественной. Связи со скрытым источником изображаются стрелками с круглыми скобками вокруг начала стрелки.

Для получения связи со скрытым приемником необходимо на диаграмме декомпозиции удалить соответствующую граничную связь, а затем перейти в родительскую диаграмму. На наконечнике соответствующей стрелки, примыкающей к родительскому блоку, в результате удаления граничной связи появились квадратные скобки (рис.1.23). Затем на палитре инструментов (см.

рис.1.15) следует выбрать режим тоннелирования стрелок и щелкнуть мышью по квадратным скобкам. В результате появится диалоговое окно тоннелирования стрелок (рис.1.24).

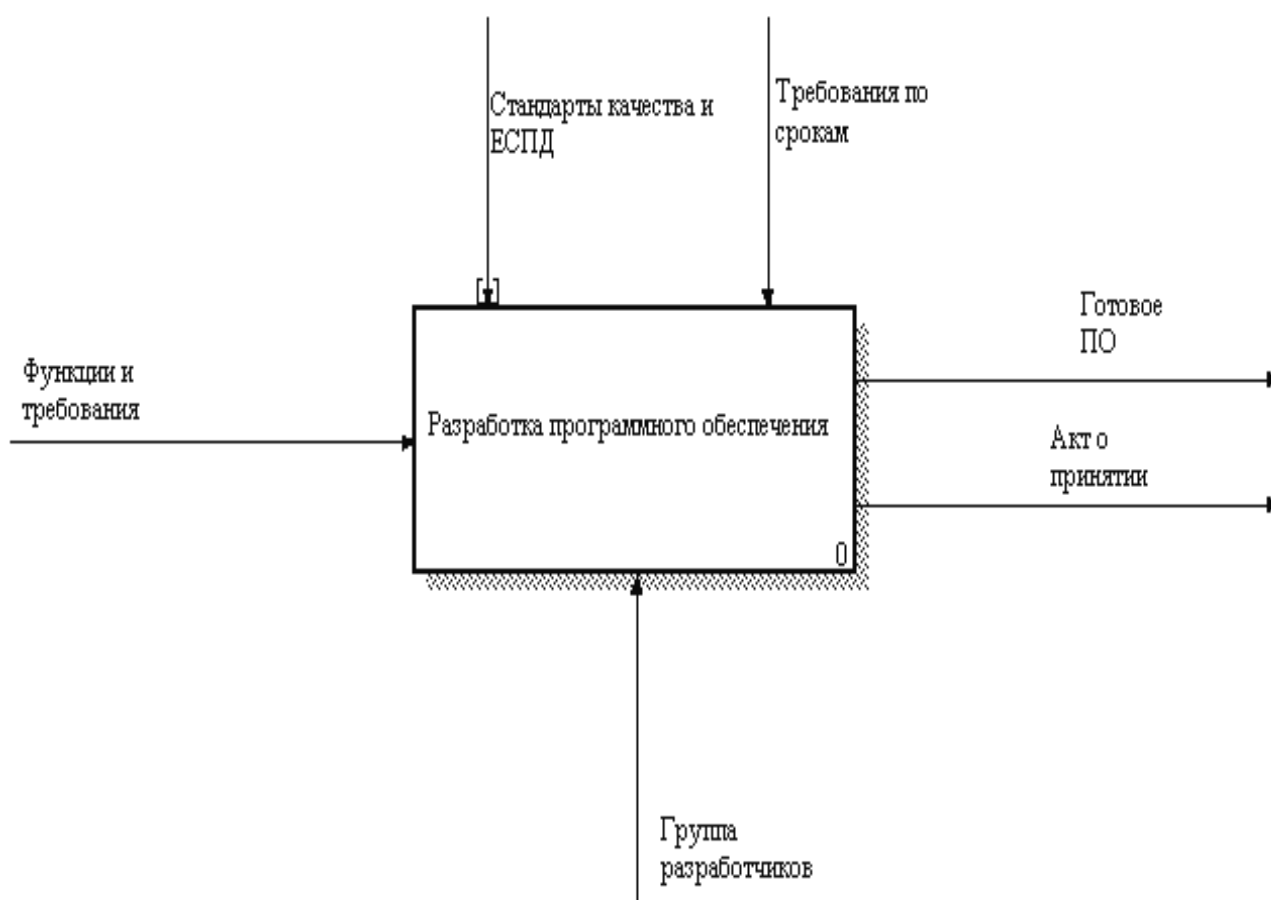


Рис.1.23. Родительская диаграмма с квадратными скобками на наконечнике стрелки

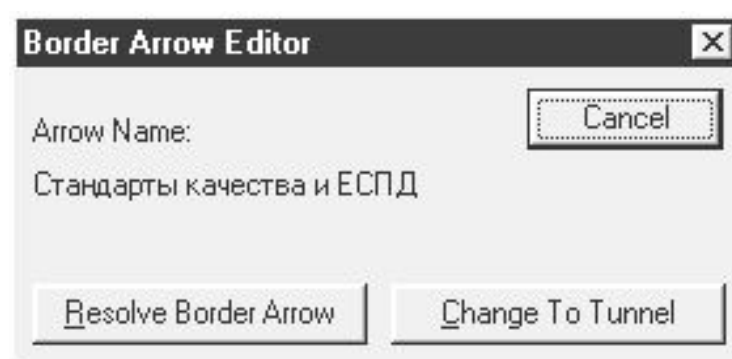


Рис.1.24. Диалог тоннелирования связей

Это же окно можно вызвать, если в режиме редактирования (см. рис.1.15) щелкнуть правой кнопкой мыши по квадратным скобкам. В появившемся контекстном меню редактирования связей (см. рис.1.19) следует выбрать пункт Arrow Tunnel. Для тоннелирования стрелки в окне (см. рис.1.24) необходимо щелкнуть по кнопке Change To Tunnel (Изменить тоннелирование).

Для получения связи со скрытым источником следует в диаграмме-потомке нарисовать граничную стрелку. На ее конце появятся квадратные скобки. Затем следует войти в режим тоннелирования стрелок (см. рис.1.15) и щелкнуть по квадратным скобкам. В появившемся диалоге (см. рис.1.24) следует щелкнуть по кнопке Change To Tunnel.

Если граничная стрелка на диаграмме была удалена по ошибке, необходимо вернуться в родительскую диаграмму и выполнить действия, аналогичные тоннелированию дуги со скрытым приемником. В появившемся диалоге (см. рис.1.24) необходимо щелкнуть по клавише Resolve Border Arrow (Разрешить граничную стрелку) и затем вернуться в диаграмму декомпозиции. На ней снова появится несвязная граничная стрелка.

Диаграмма, полученная в результате вышеописанных действий при декомпозиции контекстной диаграммы (см. рис.1.20), приведена на рис.1.25. Диаграмма декомпозиции блока «Внедрение» представлена на рис.1.26. На данном рисунке используется три дуги со скрытым источником: «Требования по срокам» и «Представитель заказчика», «Программное обеспечение среды».

Треугольники в верхнем левом углу блоков обозначают, что данные работы не были декомпозированы.

Общие правила рисования диаграмм

Блоки должны располагаться по диагонали с левого верхнего в правый нижний угол с учетом доминирования. Блок с наибольшим доминированием располагается в левом верхнем углу, блок с наименьшим доминированием – в правом нижнем углу.

Следует максимально увеличивать расстояние между стрелками на одной грани работы и между работами.

Две параллельные стрелки, начинающиеся на одной грани одной работы и заканчивающиеся на одной грани другой работы, по возможности следует объединить.

Следует минимизировать число пересечений, петель и поворотов стрелок, а также максимально увеличить расстояние между ними [4].

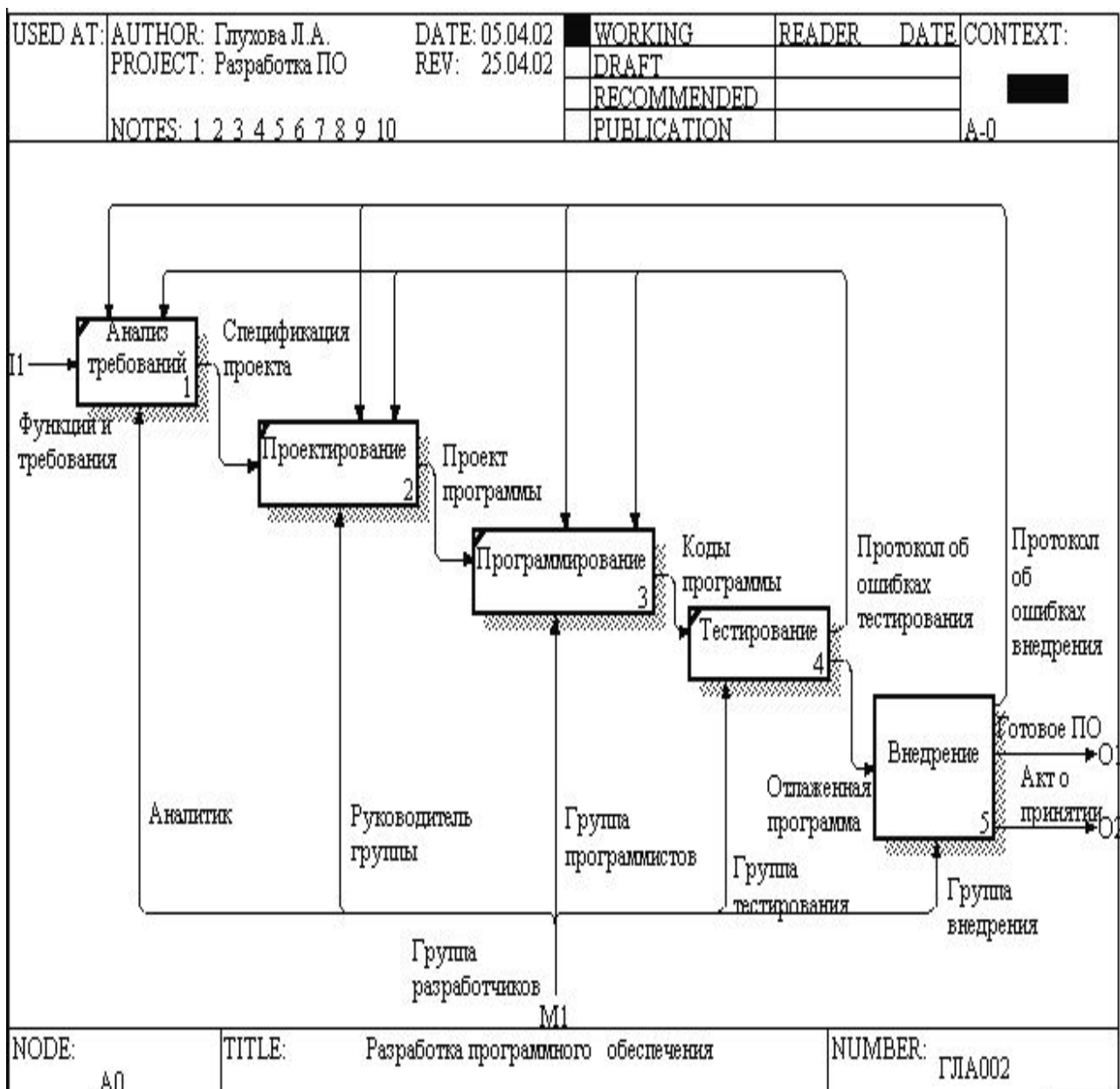


Рис.1.25. Диаграмма декомпозиции первого уровня иерархии

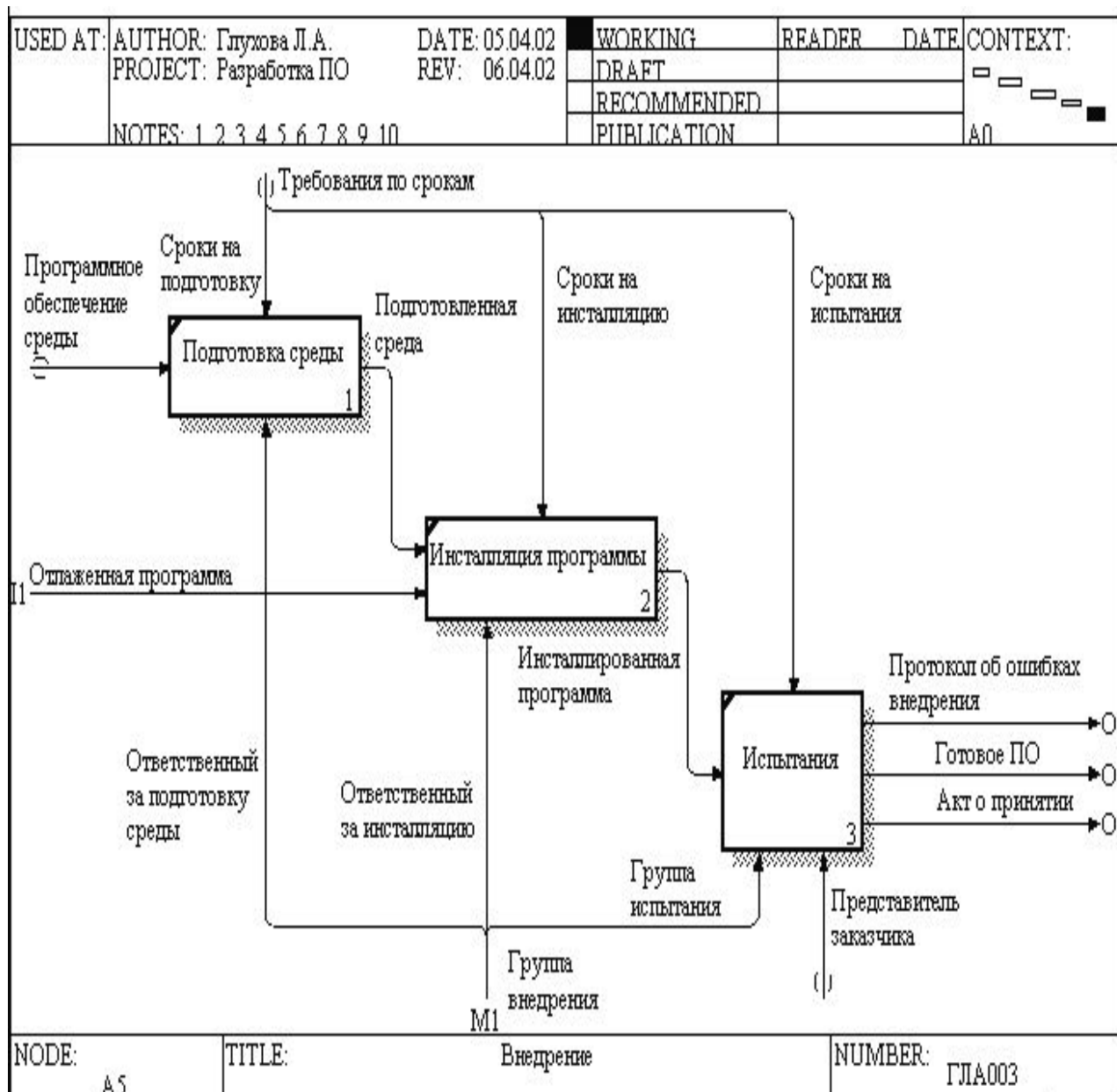


Рис.1.26. Диаграмма декомпозиции блока «Внедрение»

Создание диаграмм дерева узлов

Диаграмма дерева узлов позволяет рассмотреть целиком всю модель или выбранную часть модели. На данном виде диаграмм представляется иерархия работ в модели без указания взаимосвязей (дуг) между работами [4].

Для создания диаграммы дерева узлов следует выбрать в меню (см. рис.1.8) пункт *Insert/Node Tree*. В результате возникает диалоговое окно формирования диаграммы дерева узлов Node Tree Definition (рис.1.27). В данном окне следует указать корневую работу дерева и его глубину

(количество уровней иерархии).

Вид дерева зависит от опции Bullet Last Level. При включенной опции работы нижнего уровня иерархии дерева представляются в виде вертикального списка (рис.1.28). При выключенной опции работы нижнего уровня иерархии дерева представляются в виде прямоугольников (рис.1.29).

Node Tree Definition

Model Name: Software Design
Project Name: Разработка ПО

Diagram Name: Разработка программного обеспечения

Top Activity: A0: Разработка программного обеспечения

Number of Levels: 3

☒ Bullet Last Level
☒ Draw Node Numbers
☒ Draw Boxes

Box Width:
☒ Fit each box to text.
☐ One size per row.
☐ All one size.

☒ Include Kit
☒ Include Title

Author Name: Глухова Л.А.

Page Number:

C-Number:

Used At:

Status:
☒ WORKING
☐ DRAFT
☐ RECOMMENDED
☐ PUBLICATION
☐ Other

Creation Date: 25.03.02 (dd.mm.yy(yy))
User Last Revision Date:
System Last Revision Date: 26.04.02

OK Cancel Help

Рис.1.27. Диалоговое окно настройки диаграммы дерева узлов

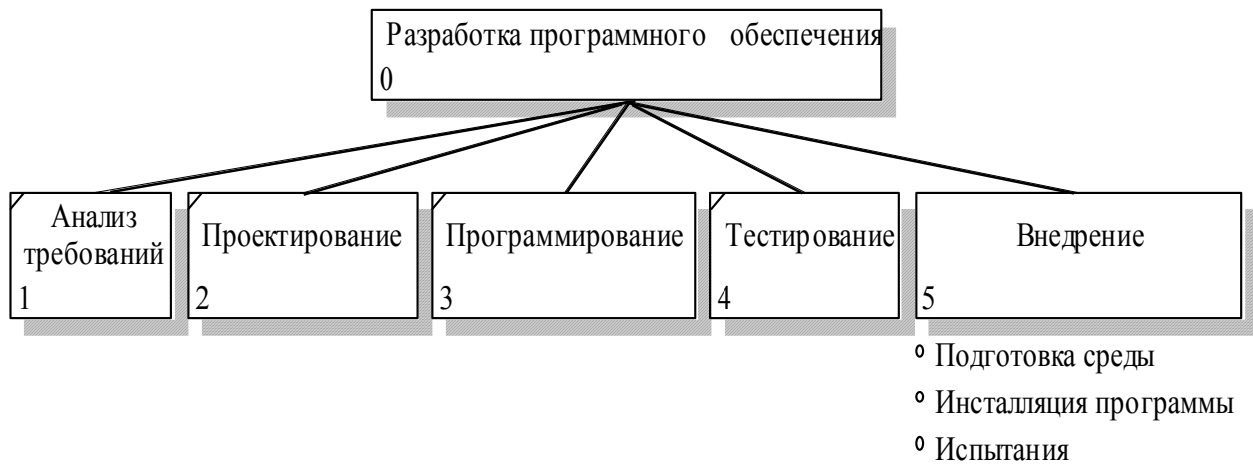


Рис.1.28. Диаграмма дерева узлов с включенной опцией Bullet Last Level

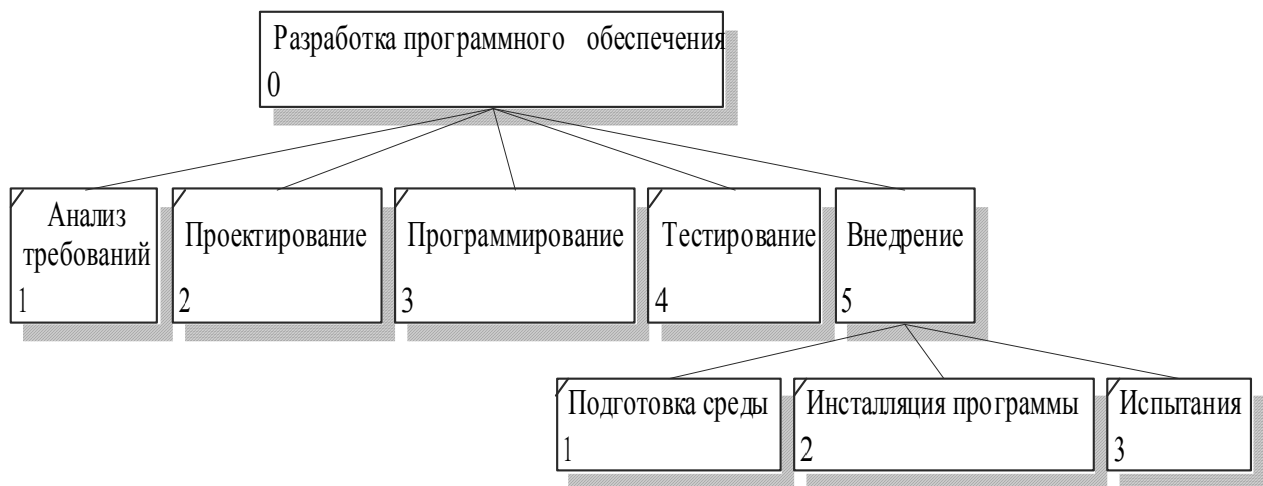


Рис.1.29. Диаграмма дерева узлов с выключенной опцией Bullet Last Level

2.1.8. Особенности построения DFD-диаграмм

Диаграммы потоков данных могут использоваться для представления всей модели или для более наглядного представления отдельных частей IDEF0-модели.

Основными компонентами диаграмм DFD являются работы, внешние сущности, хранилища данных и потоки данных.

Для дополнения IDEF0-модели диаграммой DFD нужно в ходе декомпозиции щелкнуть по радиокнопке DFD диалога Activity Box Count (см. рис.1.21). В результате палитра инструментов изменяет свой вид - на ней появляются новые кнопки (рис.1.30), соответствующие данной нотации диаграмм. Назначение данных кнопок приведено в табл.2 [4]. Назначение остальных кнопок аналогично методологии IDEF0.

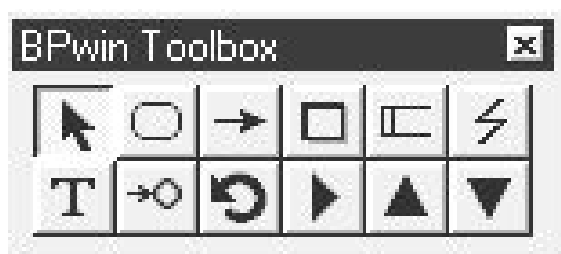


Рис.1.30. Палитра инструментов для DFD-диаграмм

Таблица 2

Кнопки палитры инструментов, соответствующие нотации диаграмм DFD

| Вид кнопки | Назначение кнопки |
|------------|--|
| | Добавить в диаграмму внешнюю ссылку |
| | Ссылка на другую страницу (off-page reference), позволяет направить стрелку на любую диаграмму |
| | Добавить в диаграмму хранилище данных (Data store) |

Пример DFD-диаграммы, построенной в BPwin, приведен на рис.1.4.

2.1.9. Особенности построения IDEF3-диаграмм

Целью методологии IDEF3 является описание последовательности выполнения процессов во времени и описание объектов, участвующих совместно в одном процессе [4].

Основными компонентами диаграмм DFD являются единицы работы (Unit of Work – UOW), связи (Arrow) и перекрёстки (Junction).

В IDEF3 существует три типа стрелок:

- > изображает связь предшествования или старшую связь (Precedence);
-> изображает связь отношения (Relational Link);
- >> изображает поток объектов (Object Flow).

Типы стрелок устанавливается с помощью меню *Edit/Arrow Style*.



Для дополнения IDEF0-модели диаграммой IDEF3 нужно в ходе декомпозиции щелкнуть по радиокнопке IDEF3 диалога Activity Box Count (см. рис.1.21). В результате палитра инструментов изменяет свой вид на соответствующий данной нотации диаграмм (рис.1.31). Назначение новых кнопок приведено в табл.3. Назначение остальных кнопок соответствует методологии IDEF0.



Рис.1.31. Палитра инструментов для IDEF3-диаграмм

Таблица 3

Кнопки палитры инструментов,
соответствующие нотации диаграмм IDEF3

| Вид кнопки | Назначение кнопки |
|---|-------------------------|
|  | Внесение объекта ссылки |
|  | Внесение перекрестка |

При нажатии кнопки «Внесение перекрестка» появляется диалоговое окно (рис.1.32), в котором выбирается тип перекрёстка. Существующие типы перекрестков сведены в табл.4 [4].

На диаграммах IDEF3 стрелки могут сливаться и разветвляться только через перекрёстки. Все перекрёстки на диаграмме нумеруются. Номер перекрёстка начинается символом J. Свойства перекрёстка редактируются при помощи диалога Definition Editor.

Пример диаграммы IDEF3, построенной в BPWin, приведен на рис.1.5.

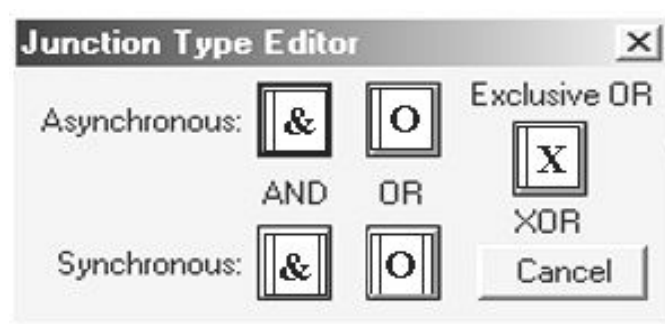


Рис.1.32. Окно выбора типа перекрёстка

Таблица 4

Типы перекрестков

| Обозначение | Наименование | Значение для перекрестков слияния | Значение для перекрестков разветвления |
|-------------|--------------------|--|---|
| | Asynchronous AND | Все предшествующие процессы должны быть завершены | Все следующие процессы должны быть запущены |
| | Synchronous AND | Все предшествующие процессы завершены одновременно | Все следующие процессы запускаются одновременно |
| | Asynchronous OR | Один или несколько предшествующих процессов должны быть завершены | Один или несколько следующих процессов должны быть запущены |
| | Synchronous OR | Один или несколько предшествующих процессов завершены одновременно | Один или несколько следующих процессов запускаются одновременно |
| | XOR (Exclusive OR) | Только один предшествующий процесс завершен | Только один следующий процесс запускается |

1.4. СОЗДАНИЕ ОТЧЁТОВ В BPWIN 2.5

Модель, выполненная в BPwin, представляет собой набор иерархически упорядоченных диаграмм. При размещении на диаграмме некоторого элемента этот элемент вместе со своими свойствами автоматически заносится в словарь BPwin. В результате вместе с графическим изображением моделируемой

системы автоматически формируется подробное текстовое описание системы.

В VPwin встроены средства для автоматической генерации отчетов. Отчеты по модели вызываются из пункта меню Report.

В VPwin существуют следующие типы отчетов [4]:

- Model Report – отчет по модели. Содержит общую информацию о модели (ее контекстной диаграмме) – имя модели, точку зрения, предметную область моделирования, цель, имя автора, дату создания и др.
- Diagram Report - отчет по диаграмме. Включает список всех элементов, входящих в состав выбранной диаграммы (сюда входят работы, стрелки, хранилища данных, внешние ссылки и т.п.).
- Diagram Object Report – отчет по объектам диаграмм. Является самым полным отчетом по модели. В общем случае включает полный список элементов модели с описанием их свойств.
- Activity Cost Report – отчет о результатах стоимостного анализа.
- Arrow Report – отчет по связям. В общем случае содержит информацию из словаря связей, информацию о работе-источнике связи, работе-приемнике связи и информацию об иерархической структуре связей (разветвлении и слиянии стрелок).
- Data Usage Report – отчет о результатах связывания модели процессов и модели данных, созданных соответственно в VPwin и в ERwin для одной и той же предметной области.
- Model Consistency Report – отчет, содержащий список синтаксических ошибок модели.

Существует три типа возможных синтаксических ошибок модели IDEF0:

- Ошибки, которые VPwin не обнаруживает. К данному типу ошибок относятся ошибки в названиях элементов модели. Например, в соответствии с синтаксисом методологии IDEF0 название работы должно быть выражено существительным, обозначающим действие («Проектирование проекта», «Контроль проекта» и т.п.). Название связи должно быть выражено существительным («Промежуточный вариант проекта», «Результат контроля» и т.п.). Обнаружение ошибок данного типа осуществляется разработчиком или руководителем проекта.
- Ошибки, которые VPwin не пропускает. К данному типу ошибок относятся ошибки графического представления модели. Например, в соответствии с синтаксисом методологии IDEF0 не допускается создание на диаграмме связи, выходящей из левой или верхней стороны работы, входящей в правую сторону работы.
- Ошибки, которые VPwin пропускает, но обнаруживает и фиксирует их список в отчете Model Consistency Report. К данному типу ошибок относятся, например, работы и связи без названия, несвязанные стрелки, работы без связей выхода или управления.

При выборе пункта меню Report, соответствующего некоторому типу отчета, появляется диалоговое окно настройки отчета. Диалоговые окна различаются для каждого из типов отчетов. Например, для отчета по объектам диаграмм Diagram Object Report диалоговое окно имеет вид, приведенный на рис.1.33.

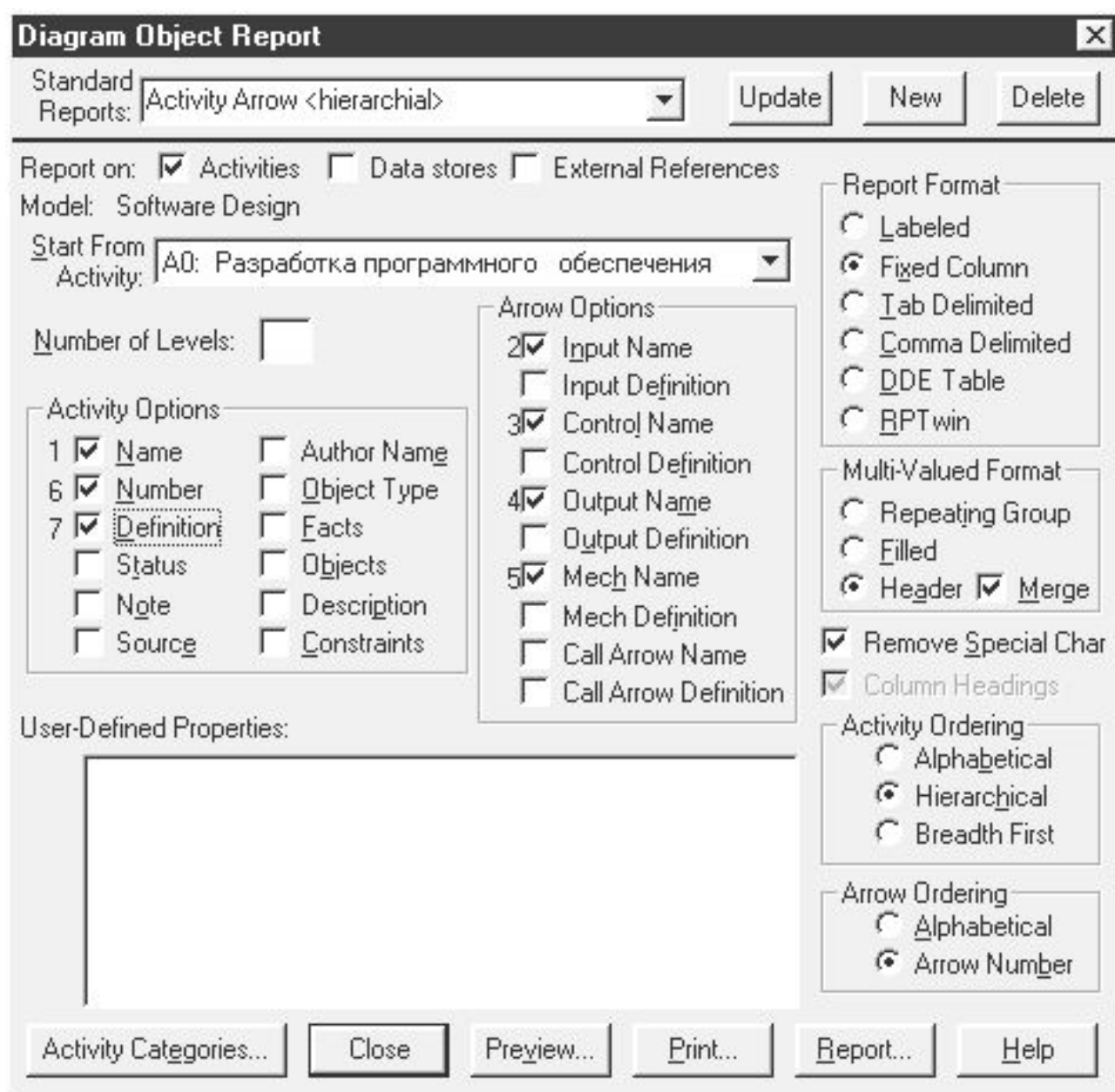


Рис.1.33. Диалоговое окно Diagram Object Report

Раскрывающийся список Standard Reports предназначен для выбора одного из стандартных отчетов. Стандартный отчет создается при определенной комбинации элементов управления диалога (переключателей, флажков и других). Информация о стандартном отчете сохраняется в файле BPWINRPT.INI. Все установки стандартного отчета, кроме определяемых

пользователем свойств (User-Defined Properties), доступны из любой модели.

Кнопка Update служит для изменения стандартного отчета, кнопка Delete – для его удаления.

Для создания собственного отчета следует ввести имя отчета в поле списка выбора, задать опции отчета и затем щелкнуть по кнопке New.

Для выбора формата отчета используется группа управляющих элементов Report Format (см. рис.1.33):

- Labeled – в отчет включается метка поля, содержимое поля печатается в следующей строке;
- Fixed Column – каждое поле печатается в своей колонке;
- Tab Delimited – поля печатаются в своих колонках, разделяемых знаком табуляции;
- Comma Delimited – поля печатаются в своих колонках, разделяемых запятыми;
- DDE Table – данные передаются по DDE приложению (например, MS Word, Excel);
- RPTwin – отчет создается в формате Platinum RPTwin – генератора отчетов, входящего в поставку BPwin.

Группа управляющих элементов Multi-Valued Format предназначена для управления выводом полей при группировке данных:

- Repeating Group – детальные данные объединяются в одно поле, между значениями вставляется +;
- Filled – для каждого заголовка группы данные дублируются;
- Header – печатается заголовок группы, затем – подробная информация (используется по умолчанию).

Опции Activity Ordering и Arrow Ordering предназначены для сортировки соответственно работ и связей.

На рис.1.34 приведен фрагмент отчета Diagram Object Report, созданный при выбранной радиокнопке Fixed Column. Фрагмент того же отчета, созданный при выбранной радиокнопке Labeled, приведен на рис.1.35.

| Activity Name | Input Name | Control Name | Output Name |
|-------------------------------------|----------------------|----------------------------------|----------------------------------|
| Разработка программного обеспечения | Функции и требования | Стандарты качества и ЕСПД | Готовое ПО |
| | | Требования по срокам | Акт о принятии |
| Анализ требований | Функции и требования | Протокол об ошибках внедрения | Спецификация проекта |
| | | Протокол об ошибках тестирования | |
| Проектирование | Спецификация проекта | Протокол об ошибках внедрения | Проект программы |
| | | Протокол об ошибках тестирования | |
| Программирование | Проект программы | Протокол об ошибках внедрения | Коды программы |
| | | Протокол об ошибках тестирования | |
| Тестирование | Коды программы | | Протокол об ошибках тестирования |
| | | | Отлаженная программа |
| Внедрение | Отлаженная программа | | Протокол об ошибках внедрения |

Рис.1.34. Diagram Object Report,
созданный при выбранной радиокнопке Fixed Column

Activity Name: Разработка программного обеспечения

Input Name: Функции и требования

Control Name: Стандарты качества и ЕСПД

Output Name: Готовое ПО

Mechanism Name: Группа разработчиков

Activity Number: 0

Control Name: Требования по срокам

Output Name: Акт о принятии

Activity Name: Анализ требований

Input Name: Функции и требования

Control Name: Протокол об ошибках внедрения

Output Name: Спецификация проекта

Mechanism Name: Аналитик

Activity Number: 1

Control Name: Протокол об ошибках тестирования

Activity Name: Проектирование

Input Name: Спецификация проекта

Control Name: Протокол об ошибках внедрения

Output Name: Проект программы

Mechanism Name: Руководитель группы

Activity Number: 2

Control Name: Протокол об ошибках тестирования

Activity Name: Программирование

Input Name: Проект программы

Control Name: Протокол об ошибках внедрения

Output Name: Коды программы

Рис.1.35. Diagram Object Report,
созданный при выбранной радиокнопке Labeled

1.5. ПОСЛЕДОВАТЕЛЬНОСТЬ ОСНОВНЫХ ШАГОВ ПРИ СОЗДАНИИ IDEF0-МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ

При выполнении лабораторной или индивидуальной учебной работы по созданию IDEF0-модели предметной области необходимо выполнить следующие основные шаги.

- Шаг 1.* Создание новой модели.
- Шаг 2.* Описание главной работы.
- Шаг 3.* Описание цели, точки зрения, предметной области и границ.
- Шаг 4.* Описание других свойств модели.
- Шаг 5.* Нанесение стрелок.
- Шаг 6.* Описание стрелок.
- Шаг 7.* Заполнение всех полей бланка диаграммы.
- Шаг 8.* Декомпозиция.
- Шаг 9.* Объединение работ и стрелок.
- Шаг 10.* Нанесение внутренних стрелок.
- Шаг 11.* Описание стрелок.
- Шаг 12.* Заполнение всех полей бланка диаграммы.
- Шаг 13.* Создание диаграмм дерева узлов.
- Шаг 14.* Создание отчета по модели.

Разработанная модель должна содержать контекстную диаграмму и два уровня декомпозиции. Предусмотреть наличие связей со скрытым источником и скрытым приемником (тоннелирование связей).

1.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Понятие Case-средств и их назначение.
2. Назначение и сущность методологии IDEF0.
3. Назначение и сущность методологии DFD.
4. Назначение и сущность методологии IDEF3.
5. Направления IDEF0-моделирования.
6. Этапы жизненного цикла программных средств, для которых наиболее эффективно использование методологии IDEF0.
7. Достоинства методологии IDEF0.
8. Цель модели в IDEF0.
9. "Точка зрения" модели в IDEF0.
10. Субъект моделирования в IDEF0. Принцип ограничения субъекта.
11. Правила представления работ на IDEF0-диаграмме.
12. Назначения сторон функциональных блоков на IDEF0-диаграмме.
13. Принцип доминирования и его представление на IDEF0-диаграмме.
14. Назначение связей на IDEF0-диаграмме.
15. Описание связей на IDEF0-диаграмме.
16. Виды отношений между блоками и дугами на IDEF0-диаграмме.
17. Типы взаимосвязей между блоками на IDEF0-диаграмме.
18. Разветвления дуг и правила их обозначения на IDEF0-диаграмме.
19. Слияние дуг и правила их обозначения на IDEF0-диаграмме.
20. Понятие диаграммы декомпозиции, родительского блока,

родительской диаграммы в IDEF0-модели.

21. Контекстная диаграмма модели.
22. Номер узла IDEF0-диаграммы. Назначение и правила записи.
23. Граничные дуги IDEF0-диаграммы и система их обозначений.
24. Тоннелирование связей. Назначение и правила обозначения.
25. Основные этапы процесса моделирования в IDEF0.
26. Методологии, поддерживаемые BPwin.
27. Последовательность действий по созданию IDEF0-модели в BPwin.
28. Назначение пунктов главного меню BPwin.
29. Установка размеров полей стандартного бланка диаграммы в BPwin.
30. Правила внесения субъекта, его границ, цели и точки зрения модели IDEF0 в BPwin.
31. Назначение видов модели AS-IS или TO-BE в BPwin.
32. Правила создания контекстной диаграммы модели в BPwin.
33. Назначение кнопок палитры инструментов для IDEF0-методологии в BPwin.
34. Правила установки шрифтов для элементов диаграммы и полей ее бланка в BPwin.
35. Правила создания граничных связей в BPwin.
36. Правила создания диаграмм декомпозиции в BPwin.
37. Правила создания, разветвления и слияния граничных связей в BPwin.
38. Правила тоннелирования связей в BPwin.
39. Правила рисования диаграмм в BPwin.
40. Диаграммы дерева узлов и правила их создания в BPwin.
41. Особенности построения DFD-диаграмм в BPwin.
42. Особенности построения IDEF3-диаграмм в BPwin.
43. Типы перекрестков на IDEF3-диаграмм в BPwin.
44. Виды отчетов в BPwin.
45. Создание отчетов в BPwin.

1.7. ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНЫХ И ИНДИВИДУАЛЬНЫХ РАБОТ

В среде BPwin разработать функциональные модели следующих предметных областей:

- 1) библиотечный каталог;
- 2) отдел кадров университета;
- 3) студенты университета;
- 4) деканат (сотрудники и студенты факультета);
- 5) расписание занятий преподавателей кафедры;
- 6) расписание занятий студентов университета;
- 7) учет результатов экзаменационных сессий студентов за весь период

обучения;

- 8) общежитие;
- 9) школа;
- 10) делопроизводство предприятия;
- 11) делопроизводство профкома организации;
- 12) учет материальных ценностей (склад);
- 13) магазины города;
- 14) общественный транспорт города;
- 15) клинические больницы города;
- 16) справочная служба аптек;
- 17) справочная бытовых услуг;
- 18) бронирование места на железнодорожном транспорте;
- 19) домоуправление;
- 20) бухгалтерия предприятия;
- 21) учет налогоплательщиков;
- 22) оплата услуг телефонной сети;
- 23) почта;
- 24) банк.

ЛИТЕРАТУРА

К ЛАБОРАТОРНОЙ РАБОТЕ № 1

1. Глухова Л.А., Бахтизин В.В. Методология структурного анализа и проектирования SADT: Учеб. пособие по курсу “Технология проектирования программ” для студ. спец. Т10.02.00. – Мн.: БГУИР, 2001.

2. Калянов Г.Н. CASE-технологии. Консалтинг при автоматизации бизнес-процессов. – М., 2000.

3. Каменнова М.С. Системный подход к проектированию сложных систем // Журнал доктора Добба. – 1993. - № 1. – С.9-14.

4. Маклаков С.В. BPwin и ERwin. CASE-средства разработки информационных систем. – М., 1999.

5. Марка Д.А., МакГоуэн К. Методология структурного анализа и проектирования SADT. – М., 1993.

ЛАБОРАТОРНАЯ РАБОТА № 2.

Информационное моделирование в среде CASE-средства ERwin

2.1. ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ

Целью информационного моделирования является идентификация концептуальных сущностей, составляющих предметную область. Информационное моделирование используется, как правило, при разработке информационных систем с целью проектирования структуры применяемой базы данных. Для ускорения процесса проектирования в настоящее время используются различные CASE-средства. Одним из широко применяемых недорогих CASE-средств является ERwin.

ERwin - средство разработки структуры базы данных (БД), сочетающее графический интерфейс Windows, инструменты для построения ER-диаграмм (Entities-Relationships-диаграммы или диаграммы «сущность-связь»), редакторы для создания логического и физического описания модели данных и прозрачную поддержку ведущих реляционных СУБД и настольных баз данных. С помощью ERwin можно создавать базы данных или проводить их обратное проектирование (реинженеринг).

Реализация моделирования в ERwin базируется на теории реляционных баз данных и на стандартизированной методологии семантического моделирования данных IDEF1X.

Методология IDEF1X определяет стандарты терминологии, используемой при информационном моделировании, и графического изображения типовых элементов на ER-диаграммах.

Возможны две точки зрения на информационную модель и соответственно два уровня модели. Первый - логический (точка зрения пользователя) - описывает реальные сущности (объекты) предметной области. Второй - физический (точка зрения программиста) - определяет представление информации в БД. ERwin объединяет их в единую диаграмму, имеющую несколько уровней представления.

2.1.1. Реляционные базы данных

В реляционной модели все данные представляются в виде сущностей и связей между ними. Например, система резервирования билетов содержит сущности "Пассажир" и "Рейс", между которыми существует связь "перевозит" ("Рейс" "перевозит" "Пассажиров").

Сущность (объект) – это абстракция некоторого множества предметов реального мира (например, человек, место, вещь, событие, концепция), для которого все предметы множества (*экземпляры*) имеют одни и те же характеристики, подчинены и согласуются с одним и тем же набором правил и

линий поведения. Сущности именуются существительными (например, "Покупатель", "Компьютер", "Служащий", "Продажа" и т.д.).

Связь - это абстракция набора отношений, которые систематически возникают между различными видами предметов в реальном мире. Таким образом, связи определяют функциональные зависимости между сущностями. Связи именуются глаголами. Например, "Служащий" *«совершает»* "Продажи".

Все предметы в реальном мире имеют некоторые характеристики (например, высота, температура, возраст). *Атрибут* – это абстракция характеристики, которой обладают все возможные экземпляры сущности. Каждый атрибут обеспечивается именем, уникальным в пределах сущности. Для определенного экземпляра сущности атрибут принимает конкретное значение. Например, сущность «Служащий» имеет атрибуты «Фамилия», «Имя», «Дата рождения» и т.д. Экземпляр данной сущности имеет значения данных атрибутов «Иванов», «Петр», «12.08.78».

Общепринятым видом графического изображения реляционной модели данных является ER-диаграмма. На такой диаграмме сущности (таблицы) изображаются прямоугольниками, соединенными между собой линиями (связями). Такое графическое представление облегчает восприятие структуры базы данных по сравнению с текстовым описанием. Представление ER-диаграмм в ERwin подробно описано ниже.

2.1.2. Сущности и атрибуты в реляционной модели

Сущность вместе с атрибутами на информационной модели может интерпретироваться как таблица. Столбцы таблицы соответствуют атрибутам. Каждая строка таблицы описывает один экземпляр сущности. Строка заполняется значениями атрибутов, соответствующими данному экземпляру. Набор атрибутов каждого экземпляра постоянен.

Так, сущность «Служащий» интерпретируется в виде таблицы, содержащей три колонки:

| Служащий | | |
|----------|---------|---------------|
| Фамилия | Имя | Дата рождения |
| Сидоров | Николай | 25.12.53 |
| Иванов | Петр | 12.08.78 |
| Петров | Федор | 18.03.64 |

Имя таблицы и имена ее колонок составляют структуру таблицы: *Служащий (Фамилия, Адрес, Идентификатор)*. В реляционной модели все значения данных являются элементарными, т.е. нельзя в клетке таблицы хранить список значений.

2.2. МОДЕЛИРОВАНИЕ В ERWIN

Процесс построения информационной модели в ERwin состоит из следующих *шагов*:

- определение сущностей;
- определение взаимосвязей между сущностями;
- задание первичных и альтернативных ключей;
- определение атрибутов сущностей;
- приведение модели к требуемому уровню нормальной формы;
- переход к физическому описанию модели: назначение соответствий между именем сущности и именем таблицы, атрибутом сущности и колонкой таблицы;
- задание триггеров, процедур и ограничений;
- генерация базы данных.

ERwin создает визуальное представление (модель данных) для решаемой задачи. Это представление может использоваться как часть документации, необходимой в цикле разработки. ERwin автоматически создает базу данных (таблицы, индексы, хранимые процедуры, триггеры для обеспечения ссылочной целостности и другие объекты, необходимые для управления данными).

2.2.1. Отображение логического и физического уровня модели данных в ERwin

В ERwin существуют два уровня представления и моделирования - логический и физический. На логическом уровне отображаются предметы (факты) из реальной жизни (например, люди, отделы, явления, события) являются реальными объектами. Они именуются на естественном языке. На логическом уровне не рассматривается использование конкретной СУБД, не определяются типы данных и не определяются индексы для таблиц.

Целевая СУБД, имена объектов и типы данных, индексы составляют второй (физический) уровень модели ERwin.

ERwin предоставляет возможности создавать и управлять этими двумя различными уровнями представления одной диаграммы (модели) и иметь много вариантов отображения на каждом уровне.

2.2.2. Компоненты диаграммы ERwin и основные виды представлений диаграммы

Диаграмма ERwin содержит три основных компонента - сущности, атрибуты и связи между сущностями. Если рассматривать диаграмму как графическое представление правил предметной области, то сущности являются существительными, а связи - глаголами.

Выбор между логическим и физическим уровнем отображения осуществляется через линейку инструментов или меню. Внутри каждого из этих уровней есть следующие *режимы отображения*:

- режим "сущности" - внутри прямоугольников отображается имя сущности (для логической модели) или имя таблицы (для физической модели); данный режим служит для удобства обзора большой диаграммы или размещения сущностей на диаграмме;
- режим "определение сущности" - служит для презентации диаграммы другим людям;
- режим "атрибуты" - при переходе от предметной области к модели требуется вводить информацию о том, что представляет собой сущность; эта информация вводится путем задания атрибутов (на физическом уровне - колонок таблиц); в данном режиме прямоугольник-сущность делится линией на две части - в верхней части отображаются атрибуты (колонок), составляющие первичный ключ, а в нижней - остальные атрибуты; этот режим является основным при проектировании на логическом и физическом уровнях;
- режим "первичные ключи" - внутри прямоугольников-сущностей показываются только атрибуты (колонок), составляющие первичный ключ;
- режим "пиктограммы" - для презентационных целей каждой таблице может быть поставлена в соответствие пиктограмма (bitmap);
- режим "показ глагольной фразы" - на связях показываются глагольные фразы, связывающие сущности (для логического уровня) или имена внешних ключей (для физического уровня).

Диаграмма может занимать более чем один экран и более чем один лист при печати. Для обзора модели предусмотрены, кроме прокруток экрана, режимы уменьшения/увеличения изображения, отображения всей модели, отображения выделенной части модели.

2.2.3. Инструменты для создания модели в ERwin

Основные инструменты создания модели доступны как из меню, так и через панель инструментов. С их помощью создаются различные виды сущностей, связей, атрибутов и текстовые элементы.

Нажатием мыши над сущностью производится вход в один из редакторов Erwin. Существуют следующие виды редакторов:

- редакторы, связанные с сущностью в целом (определение сущности, дополнительная информация, триггеры, индексы, характеристики таблицы, связанные с таблицей хранимые процедуры);
- редакторы атрибутов (определение атрибутов, колонки таблицы в физическом представлении модели, репозиторий средства 4GL).

2.2.4. Идентификация сущностей. Сущности (Entities) в ERwin

На ER-диаграмме сущность изображается прямоугольником. В зависимости от режима представления диаграммы прямоугольник может содержать имя сущности, ее описание, список ее атрибутов и другие сведения.

Горизонтальная линия прямоугольника разделяет атрибуты сущности на два набора. Атрибуты, составляющие первичный ключ, записаны в верхней части прямоугольника. Прочие атрибуты (не входящие в первичный ключ) записаны в нижней части.

Сущность представляет собой множество реальных или абстрактных экземпляров (например, люди, места, события, факты), имеющих общие характеристики. Сущность - это логическое понятие. Сущности соответствует таблица в реальной СУБД. В ERwin сущность визуально представляет три основных вида информации:

- атрибуты, составляющие первичный ключ;
- неключевые атрибуты;
- тип сущности (независимая/зависимая).

Первичный ключ - это атрибут или набор атрибутов, уникально идентифицирующий экземпляр сущности. Если несколько наборов атрибутов могут уникально идентифицировать сущность, то выбор одного из них осуществляется разработчиком на основании анализа предметной области.

Для каждого первичного ключа ERwin создает при генерации структуры БД уникальный индекс.

Экземпляры *независимой сущности* могут быть уникально идентифицированы без определения ее связей с другими сущностями. *Зависимая сущность* не может быть уникально идентифицирована без определения ее связей с другими сущностями. Зависимая сущность

отображается в ERwin прямоугольником с закругленными углами.

2.2.5. Связи (relationships) в ERwin

Связь - это функциональная зависимость между двумя сущностями. Например, между сущностями "отдел" и "сотрудник" существует связь "состоит из" (отдел *состоит* из сотрудников). Связь - это понятие логического уровня, которому соответствует внешний ключ на физическом уровне. В ERwin связи определяются пятью основными элементами информации:

- тип связи (идентифицирующая, неидентифицирующая, полная/неполная категория, неспецифическая связь);
- родительская сущность;
- дочерняя (зависимая) сущность;
- мощность связи (cardinality);
- допустимость пустых (null) значений.

Связь называется *идентифицирующей*, если экземпляр дочерней сущности идентифицируется через ее связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в первичный ключ дочерней сущности. Дочерняя сущность при идентифицирующей связи всегда является зависимой.

Связь называется *неидентифицирующей*, если экземпляр дочерней сущности не идентифицируется через связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в состав неключевых атрибутов дочерней сущности.

Для определения связи в ERwin выбирается тип связи, затем мышью указывается родительская и дочерняя сущность. Идентифицирующая связь изображается сплошной линией; неидентифицирующая - пунктирной линией. Линии заканчиваются точкой со стороны дочерней сущности.

При определении связи происходит миграция атрибутов первичного ключа родительской сущности в соответствующую область атрибутов дочерней сущности. Поэтому такие атрибуты не вводятся вручную. Атрибуты первичного ключа родительской сущности по умолчанию мигрируют со своими именами. ERwin позволяет ввести для них *роли* - новые имена, под которыми мигрирующие атрибуты будут представлены в дочерней сущности. В случае неоднократной миграции атрибута такое переименование необходимо.

Например, первичный ключ "код предприятия" сущности "Предприятие" имеет две роли в дочерней сущности "Посредническая сделка" - это атрибуты "код предприятия-продавца" и "код предприятия-покупателя". На физическом уровне имя роли - это имя колонки внешнего ключа в дочерней таблице.

Мощность связи представляет собой отношение количества экземпляров родительской сущности к соответствующему количеству экземпляров дочерней сущности. Для любой связи, кроме неспецифической, эта связь записывается как 1:n.

ERwin в соответствии с методологией IDEF1X предоставляет четыре варианта значения n :

- ноль, один или более (по умолчанию);
- один или более;
- ноль или один;
- ровно N ,

где N - конкретное число.

Допустимость пустых (NULL) значений в неидентифицирующих связях ERwin изображается пустым ромбиком на дуге связи со стороны родительской сущности.

Имя связи на логическом уровне представляет собой "глагол", связывающий сущности. Физическое имя связи (которое может отличаться от логического) для ERwin означает имя ограничения (constraint) или индекса.

2.2.6. Связи категоризации

Некоторые сущности определяют категорию подобных объектов. В ERwin в таком случае создаются сущности для определения категории и каждого элемента категории. Родительская сущность категории называется *супертипом*, а дочерние сущности категории - *подтипами*. Связь между супертипом и подтипами называется *связью категоризации (связью супертип-подтип)*.

Пример. Пусть сущность "Служащий" содержит данные о штатных работниках и совместителях. Первые и вторые имеют различные, частично пересекающиеся наборы атрибутов. Общая часть этих атрибутов, включая первичный ключ, помещается в сущность-супертип "Служащий". Различающиеся части помещаются в сущности-подтипы.

В сущности-супертипе вводится *атрибут-дискриминатор*, позволяющий различать конкретные экземпляры сущности-подтипа.

В конструкции супертип-подтип один реальный экземпляр представляется комбинацией экземпляра супертипа и экземпляра одного подтипа.

В зависимости от того, все ли возможные сущности-подтипы включены в модель, категоризационная связь является *полной* или *неполной*.

Если в предыдущем примере супертип может содержать данные об уволенных сотрудниках, то связь категоризации является неполной, если не существует соответствующих сущностей-подтипов.

В ERwin полная категория изображается окружностью с двумя подчеркиваниями, а неполная - окружностью с одним подчеркиванием.

2.2.7. Графическое редактирование модели

Все объекты модели ERwin могут редактироваться средствами, принятыми в Windows, - группировка, копирование, удаление, перемещение, использование системного буфера, установка цветов и шрифтов.

Компоненты модели, представленные текстом (имена сущностей, атрибутов, текстовые элементы), могут редактироваться непосредственно на экране.

2.2.8. Альтернативные ключи

Альтернативный ключ - это атрибут (или группа атрибутов), несовпадающий с первичным ключом и уникально идентифицирующий экземпляр сущности. Например, для сущности «Служащий» (идентификатор служащего, фамилия, имя, отчество) идентификатор служащего является первичным ключом, а группа атрибутов "Фамилия", "Имя", "отчество" является альтернативным ключом (в предположении, что на предприятии не работают полные тезки).

Для альтернативного ключа, как и для первичного, ERwin автоматически создает индексы при генерации БД.

2.2.9. Инвертированные индексы

Атрибуты, составляющие альтернативный ключ, однозначно (уникально) идентифицируют экземпляры сущности. В ERwin можно также составлять группы атрибутов, которые не идентифицируют уникально экземпляры сущности, но часто используются для доступа к данным. Для каждой такой группы атрибутов ERwin создает неуникальные индексы.

Одни и те же атрибуты сущности могут входить в различные группы ключей.

2.2.10. Унификация атрибутов

Зависимая сущность может наследовать один и тот же внешний ключ от более чем одной родительской сущности или от одной и той же родительской сущности через несколько связей. Если не введены различные роли для такого множественного наследования, ERwin считает, что в зависимой сущности атрибуты внешнего ключа появляются только один раз.

Унификация - это объединение двух или более групп атрибутов внешних ключей в один внешний ключ (группу атрибутов) в предположении, что значения одноименных атрибутов в дочерней сущности всегда одинаковы.

Пример. Пусть сущность "Служащий" имеет первичный ключ "Код служащего" и связан идентифицирующей связью с сущностями "Супруга" и "Дети". При этом происходит миграция первичного ключа в зависимые сущности. В свою очередь, сущность "Супруга" связана неидентифицирующей связью с сущностью "Дети". Имеются два пути миграции ключа, однако в сущности "Дети" атрибут "Код служащего" появляется один раз в качестве элемента первичного ключа.

Существуют случаи, когда унификация атрибутов дает неверный с точки зрения предметной области результат. Имена ролей отменяют унификацию атрибутов.

2.2.11. Реализация ссылочной целостности с помощью ERwin

Ссылочная целостность - это обеспечение соответствия значения внешнего ключа экземпляра дочерней сущности значению первичного ключа в родительской сущности. Ссылочная целостность может контролироваться при всех операциях, изменяющих данные (INSERT / UPDATE / DELETE). Средства контроля ссылочной целостности в ERwin включают автоматическую генерацию триггеров и использование механизмов декларативной ссылочной целостности (для тех СУБД, которые поддерживают данные механизмы).

Для каждой связи на логическом уровне могут быть заданы требования по обработке операций INSERT / UPDATE / DELETE для родительской и дочерней сущности. ERwin предоставляет следующие варианты обработки этих событий:

- отсутствие проверки;
- проверка допустимости;
- запрет операции;
- каскадное выполнение операции (DELETE / UPDATE);
- установка пустого (NULL-значения) или заданного по умолчанию значения.

В соответствии с выбранным вариантом ERwin автоматически создает необходимые триггеры на диалекте SQL целевой СУБД. При этом ERwin пользуется библиотекой шаблонов триггеров, которые можно модифицировать.

При генерации структуры базы данных триггеры, обеспечивающие ссылочную целостность, могут быть переопределены на трех уровнях:

- 1) триггеры, обеспечивающие правила для всей модели;
- 2) триггеры, указанные для конкретной связи;
- 3) триггеры, указанные для конкретной таблицы.

Тип переопределения указывается разработчиком при генерации схемы базы данных.

2.2.12. Хранение информации в модели ERwin

Обычно модели ERwin сохраняются на диске в виде файла. Имеется возможность хранить модель в целевой СУБД. Для этого с помощью самого ERwin в целевой СУБД создается метабаза ERwin. В этой базе данных сохраняется информация модели. В частном случае базой данных могут быть и dBase-файлы, с которыми ERwin работает через ODBC.

2.3. РАСШИРЕННЫЕ ФУНКЦИИ ERWIN

2.3.1. Обратное проектирование (Reverse engineering)

Обратное проектирование, то есть построение информационной модели по существующей базе данных, используется при выборе оптимальной платформы (rightsizing) для существующей настольной (desktop) базы данных или базы данных на mainframe, а также при модификации существующей структуры, которая была построена без необходимой сопроводительной документации. После завершения процесса восстановления модели ERwin автоматически формирует ER-диаграмму. Это позволяет выполнять модификации уже с использованием логической схемы - добавлять сущности, атрибуты, комментарии, связи и т.д. По завершении изменений одна команда - синхронизировать модель с базой данных - актуализирует все проведенные изменения.

Построение модели может быть выполнено как на основании данных каталога базы данных, так и на основании пакета операторов SQL, с помощью которого была создана база данных.

2.3.2. Синхронизация с базой данных

В процессе разработки информационной системы может возникнуть ситуация, когда структура базы данных и информационная модель не соответствуют друг другу. ERwin предоставляет возможность привести их в соответствие.

Для этого предусмотрена функция синхронизации с базой данных. После подключения к СУБД предлагается список несоответствий между существующей структурой данных и моделью. Например, если в базе данных создана новая таблица, то ERwin предложит провести включение ее в модель. Если в модель добавлена новая таблица, ERwin предложит создать ее в

реальной базе данных. Аналогично при добавлении колонок в базе данных или в модели ERwin предлагает провести соответствующие операции по синхронизации. Выбор синхронизируемых таблиц иллюстрирует рис.2.1.

Complete Compare - Set Options

Compare Current Model with—

☒ Database ☐ ModelMart Diagram

☐ Script File ☐ ER1/ERX File

File Name:

Options—

Compare—

☐ System Objects

Tables/Views Owned By—

☒ All ☐ Current User

☐ Owners (comma separated):

☒ Model Objects Only

Match—

☒ Using Owner

Case Conversion of Physical Names—

☒ None ☐ lower case ☐ UPPER CASE

Items to Compare—

Option Set:

▼

☒ Column

☒ Name

☒ PK

☒ Datatype

☒ Null Option

☒ Comment

☒ Check

☒ Default

Рис.2.1. Синхронизация с базой данных

Для синхронизации с базой данных необходимо открыть меню Edit и выбрать пункт Column... . В появившемся диалоговом окне «Column editor» необходимо выбрать кнопку Db Sync... .

В ERwin имеется возможность учитывать особенности хранения данных в отдельных СУБД. Информация о физическом размещении может быть включена в модель и использована при прямом и обратном проектировании.

2.3.3. Интерфейсы с СУБД

ERwin поддерживает прямой интерфейс с основными СУБД: DB2 версий 2 и 3, Informix версий 5.1, 6.0, 7.1, Ingres, NetWare SQL, ORACLE версий 6 и 7, Progress, Rdb версий 4 и 6, SQL/400 версий 2 и 3, SQLBase версий 5 и 6, SQL Server версий 4 и 6, Sybase версии 4.2, Sybase System 10 и 11, Watcom SQL. Поддерживаются как современные, так и предыдущие версии основных СУБД (рис.2.2).

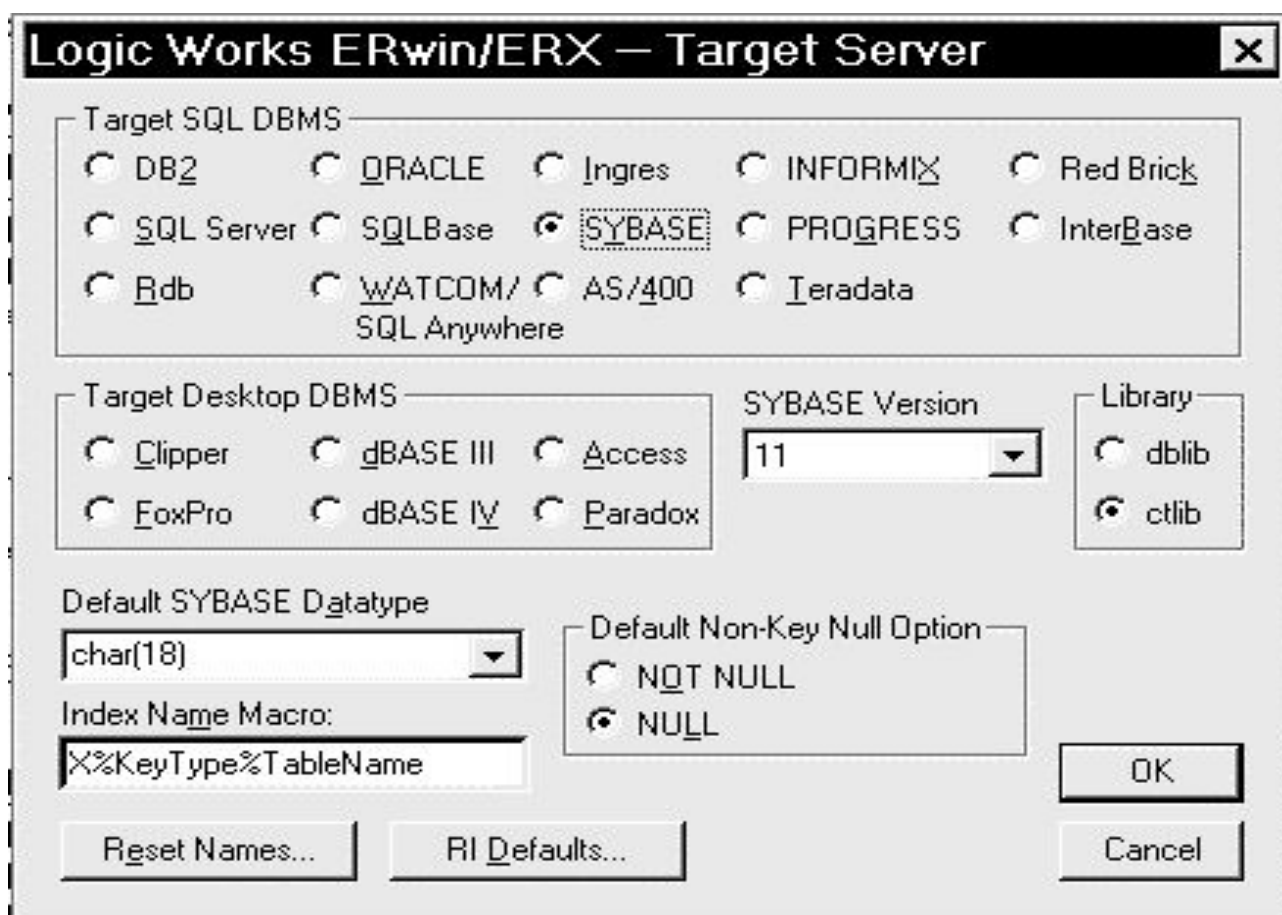


Рис.2.2. Выбор СУБД для создания модели

ERwin поддерживает также настольные (desktop) СУБД: Microsoft Access, FoxPro, Clipper, dBASE III, dBASE IV и Paradox.

Проектирование на физическом уровне выполняется в терминах той базы данных, которую предполагается использовать в системе. ERwin позволяет учесть возможности СУБД различных производителей. Поэтому возможно преобразование физической схемы, спроектированной для одной СУБД, в другую.

Для создания физической структуры БД может быть запрошена генерация DDL-скрипта (data definition language). При этом используется диалект SQL для выбранного типа и версии сервера. Имеется возможность сохранить сгенерированный код в файл или распечатать.

2.3.4. Поддержка средств 4GL

ERwin выпускается в нескольких различных редакциях, ориентированных на наиболее распространенные средства разработки 4GL. В числе поддерживаемых средств - PowerBuilder фирмы Powersoft, SQL Windows фирмы Gupta, Visual Basic фирмы Microsoft, Oracle*CASE фирмы Oracle.

Средства двунаправленного взаимодействия ERwin с базой данных обеспечивают управление информацией, ориентированной как на серверную, так и на клиентскую часть. Например, для PowerBuilder можно просматривать и редактировать расширенные атрибуты в редакторах ERwin.

Ориентация ERwin на средства 4GL позволяет задать для будущих приложений большинство параметров, непосредственно связанных с базой данных, уже на стадии проектирования информационной модели.

2.3.5. Программирование триггеров и процедур

ERwin реализует собственный макроязык для подготовки прототипов триггеров и процедур. Схема использования прототипов заключается в подготовке шаблона для различных типов триггеров (например, триггер, реализующий логику каскадного удаления, ON DELETE CASCADE).

Базовые шаблоны встроены в ERwin, но пользователь может определить свои собственные шаблоны и использовать их вместо стандартных.

Макроязык шаблонов реализует большое количество макросимволов, ссылающихся на различные объекты базы данных, например:

- %Action - расширяется в UPDATE/INSERT/DELETE;
- %ForEachAtt(<таблица>,<разделитель>) {<код макрокоманды>} - циклическое выполнение группы операторов над каждым атрибутом таблицы;
- %ForEachEntity() {} - циклическое выполнение функций над всеми таблицами;
- %If, %else - операторы условного управления.

Например, шаблон триггера для реализации поддержки on delete

cascade:

```
%Action  /* ERwin Builtin %Datetime */  
/* %Parent %VerbPhrase %Child ON PARENT DELETE CASCADE */  
delete %Child  
from %Child,deleted  
where  
/* %%JoinFKPK(%Child,deleted," = "," and") */  
%JoinFKPK(%Child,deleted," = "," and")
```

Все макрофункции, которые могут использоваться в триггерах, могут использоваться также и в процедурах. Процедуры, как и триггеры, связываются с таблицей. Такой подход позволяет полностью исключить хаотичное внесение изменений в базу данных, так как модель в ERwin описывает все аспекты базы, в том числе обеспечиваемые триггерами.

2.3.6. Правила и начальные значения

В ERwin поддерживаются два типа правил (проверок допустимости значений) и начальных (по умолчанию) значений. Правило и умолчание может быть указано для проверки со стороны клиента (например, в PowerBuilder) и со стороны сервера.

При задании правила или умолчания для клиентской части эти атрибуты переносятся в репозиторий средства 4GL.

На рис.2.3 показан диалог для задания значений по умолчанию, устанавливаемых в PowerBuilder. В одном и том же диалоге задаются умолчания, используемые как на стороне клиента, так и на стороне сервера.

2.3.7. Домены

Часто используемые комбинации свойств можно поименовать. Такая комбинация свойств, называемая доменом, может наследоваться.

Например, можно определить домен "Дата" для отображения в приложении всех колонок с датами в одном стиле, домен "Дата рождения ребенка" наследует все атрибуты домена "Дата" и вносит дополнительный атрибут - цвет отображения.

Пример определения домена показан на рис.2.4.

Назначение доменов для сервера аналогично назначению доменов для клиента. Различие заключается в том, что правила и начальные значения для сервера определяются в генерации схемы базы данных, а аналогичные атрибуты для клиента сохраняются в репозитории средства 4GL.

Другое назначение доменов для сервера - определение пользовательских типов данных. Пользовательскому типу данных ставится в соответствие тип, "известный" СУБД. При выполнении синхронизации с базой данных для СУБД,

поддерживающих пользовательские типы, выполняются соответствующие команды.

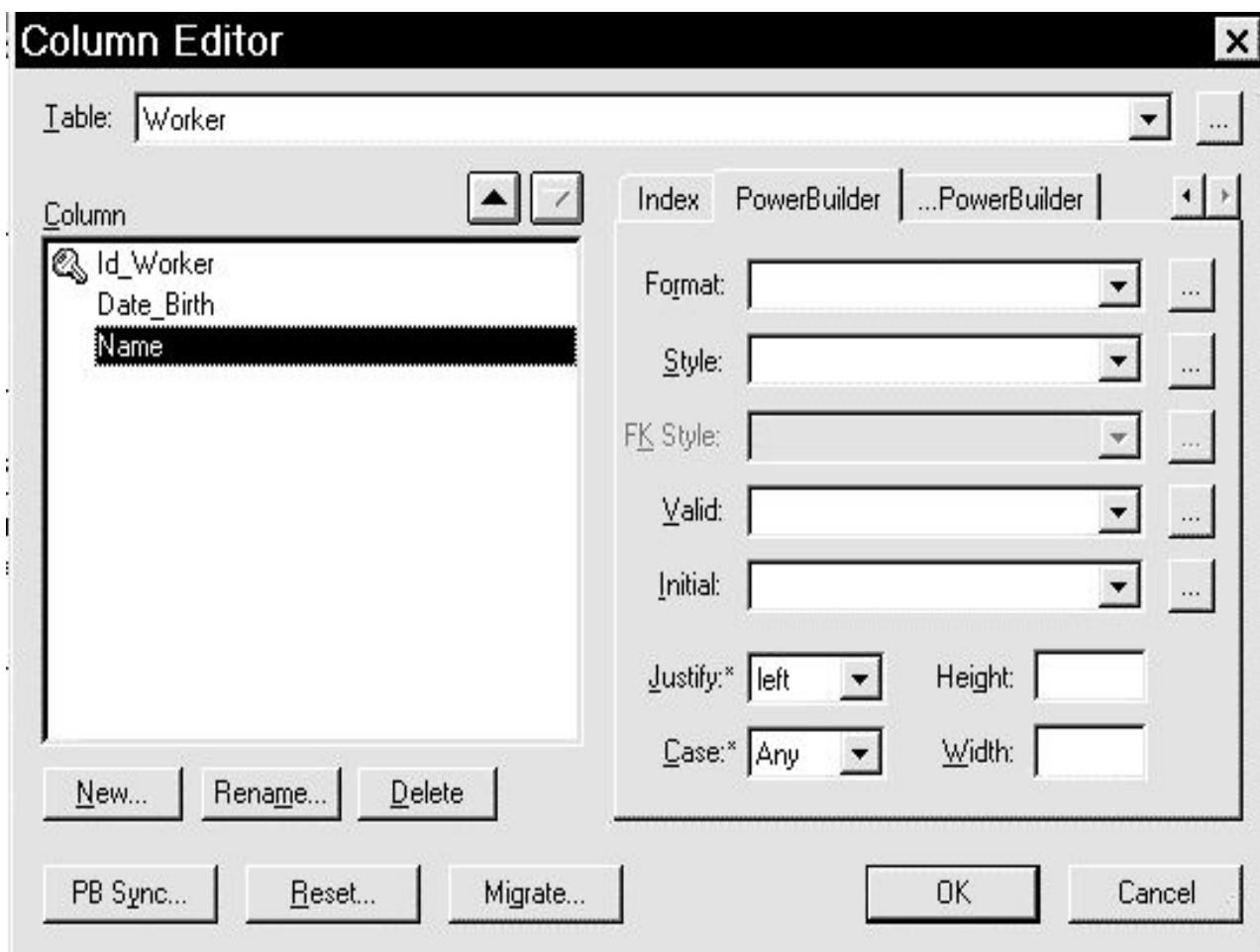


Рис.2.3. Определение умолчания для PowerBuilder



Рис.2.4. Определение домена

2.3.8. Генерация отчетов

По завершении работы над информационной моделью, как правило, распечатываются логический и физический уровни диаграммы, а также отчет по соответствиям: сущность-таблица, атрибут-имя колонки, сущность-атрибуты. Диаграмма физической модели является необходимым, почти достаточным и очень удобным материалом для разработчиков программ. Дополнительная информация для группы разработчиков прикладных программ содержится в отчете "Имена таблиц и колонок", который может быть легко построен с помощью ERwin. Выбор режима построения отчета показан на рис.2.5.

Сгенерированный отчет может быть сохранен на диске (колонки разделяются запятыми, выравниваются или разделяются табуляцией) или передан в текстовый процессор (или электронную таблицу) через интерфейс DDE.

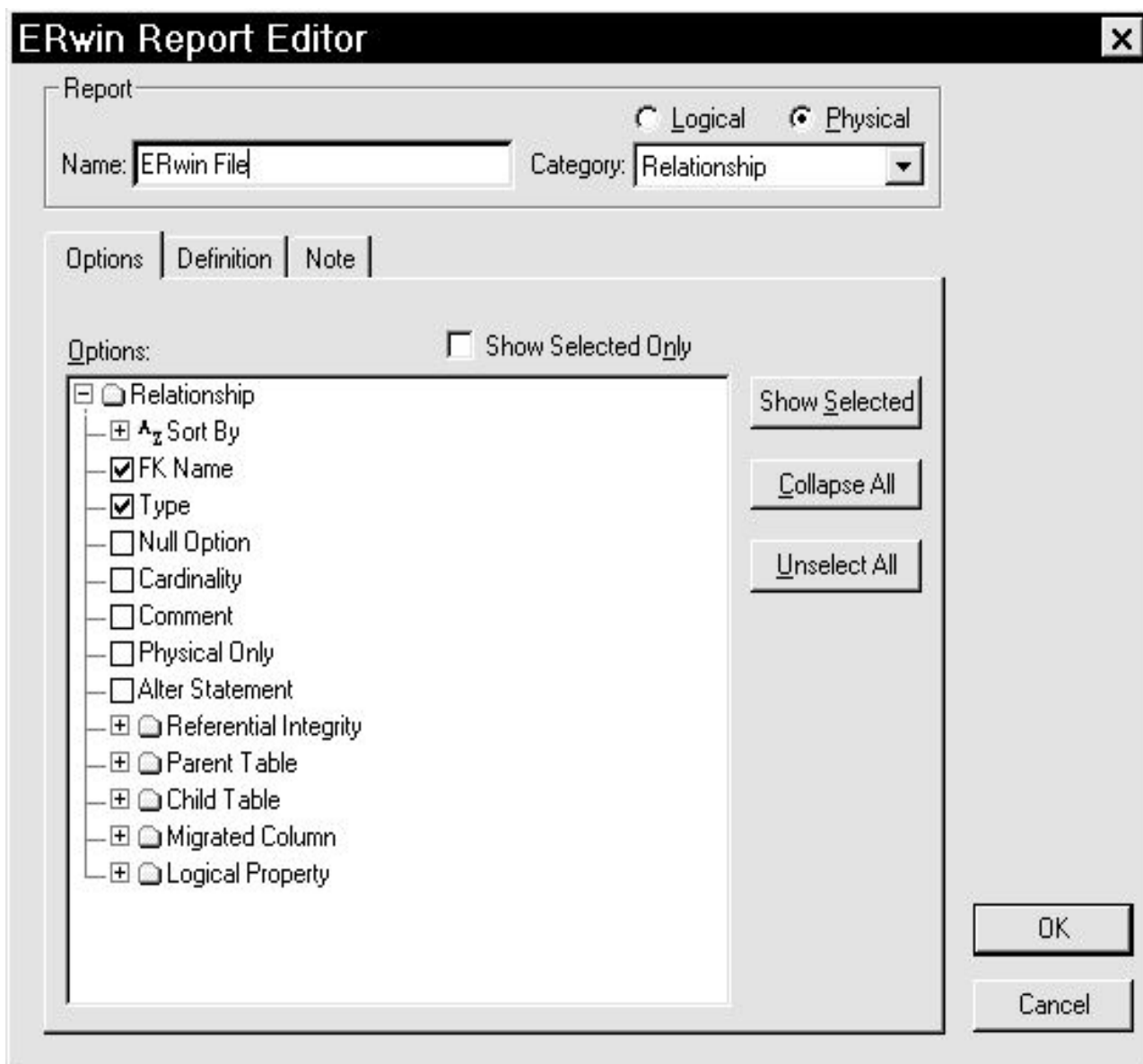


Рис.2.5. Выбор варианта отчета

2.3.9. Настройка режимов отображения

Диаграммы информационных моделей современных информационных систем обычно весьма велики, вследствие чего работать со всей диаграммой достаточно сложно как на стадии проектирования информационной модели, так и при разработке прикладного программного обеспечения. ERwin дает возможность работать не со всей диаграммой, а с логически законченной группой сущностей, называемой предметной областью (Subject Area). Переключение отображения с одной предметной области на другую производится выбором из раскрывающегося списка.

Рассмотрим информационную модель для некоторого абстрактного

предприятия. В информационной системе задействованы бухгалтерия, склад, кадры. Сущности «сотрудник», «история работы», «дети» могут быть выделены в предметную область "кадры".

Такой подход обладает рядом важных преимуществ. Во-первых, группа разработчиков программного обеспечения снабжается диаграммой той предметной области, с которой она работает. Во-вторых, при разработке информационной модели проектировщик может удалить с экрана уже спроектированные блоки, чтобы они не загромождали диаграмму. В-третьих, использование предметных областей стимулирует структурный подход к разработке информационной модели - выделение крупных блоков с последующей их детальной разработкой. Процесс разработки выполняется по технологии нисходящего проектирования.

Уровень детализации диаграммы информационной модели может изменяться проектировщиком. Например, могут отображаться только имена сущностей (таблиц), может быть включено/выключено отображение мощности связи, может быть включено/выключено отображение альтернативных ключей, может отображаться физическая или логическая модель. Для удобства проектировщика предусмотрена возможность присвоения имени группе параметров отображения. Определенные пользователем имена показываются на экране в виде закладок, что обеспечивает переключение с одного режима отображения на другой одним щелчком мыши.

Проектировщик информационной модели имеет возможность использовать цветное и шрифтовое выделение для различных компонентов диаграммы. Выделение может быть выполнено как для всей модели (например, все внешние ключи отображаются синим цветом), так и для отдельного компонента (одна таблица, все атрибуты таблицы, один атрибут таблицы, одна связь и т.д.). Использование цветного и шрифтового выделения на диаграмме информационной модели делает ее более наглядной и позволяет проектировщику обратить внимание читателей диаграммы на ее отдельные элементы.

2.4. ПРЕИМУЩЕСТВА ПРИМЕНЕНИЯ ERWIN

Применение ERwin существенно повышает эффективность деятельности разработчиков информационных систем. Основные преимущества от использования Erwin:

- существенное повышение скорости разработки за счет мощного редактора диаграмм и автоматической генерации базы данных;
- автоматическая подготовка документации;
- отсутствие необходимости ручной подготовки SQL-предложений для создания базы данных;

- возможность легкого внесения изменения в модель при разработке и расширении системы;
- возможность автоматической подготовки отчетов по базе данных, соответствующих реальной структуре БД;
- повышение эффективности разработки прикладного программного обеспечения за счет использования удобных в работе диаграмм;
- возможность задания отображения данных в приложениях на стадии информационного моделирования за счет тесной интеграции со средствами 4GL;
- возможность обратного проектирования, что позволяет документировать и вносить изменения в существующие информационные системы;
- поддержка однопользовательских СУБД; это позволяет использовать для персональных систем современные технологии, что значительно упрощает переход от настольных систем к системам в технологии клиент-сервер (upsizing).

2.5. ОПИСАНИЕ КОМАНД

Для более близкого знакомства с Case-средством ERwin в данном разделе дается описание его основных команд.

На рис.2.6 приведен внешний вид главного меню Erwin. Главное меню содержит пункты File, ModelMart, Edit, Tasks, Client, Server, Option, Window, Help.

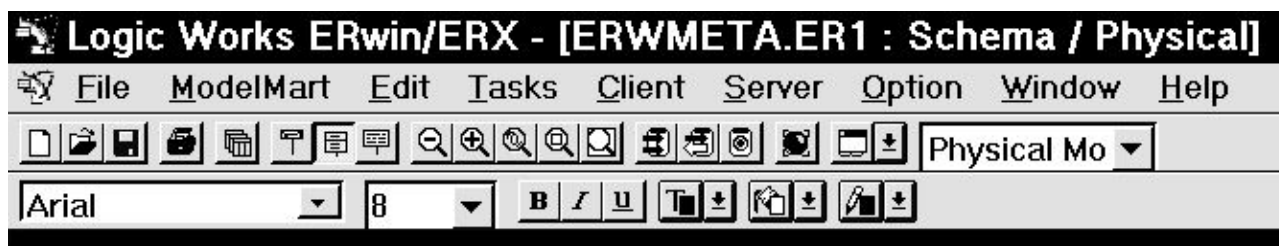


Рис.2.6. Главное меню ERWin

2.5.1. Описание команд меню File

В состав меню *Файл* входят следующие команды:

- New - создает новый Erwin-файл данных;
- Open - позволяет открыть уже существующий Erwin-файл;
- Close - закрывает текущий файл;
- Save - сохраняет текущий файл;

- Save as - сохраняет текущий файл под другим именем;
- Dictionary manager - позволяет выбрать язык, на котором будет создан SQL скрипт;
- Exit - завершает сеанс работы с Erwin;
- Print и Print Setup - осуществляют настройку печати;
- BPWin и Designer/2000 - осуществляют взаимодействие с CASE-средствами BPWin и Designer/2000.

2.5.2. Описание команд меню ModelMart

Команды меню ModelMart обеспечивают согласованное проектирование БД и приложений при коллективной разработке в сетевой версии Erwin.

2.5.3. Описание команд меню Edit

Состав команд меню *Edit* различается в зависимости от уровня моделирования (логический или физический).

Команды меню *Edit* позволяют:

- редактировать объекты схемы (под объектами в данном случае понимаются сущности, атрибуты, связи и др. в логической модели или таблицы, колонки, связи, индексы, триггеры и др. в физической модели);
- создавать новые объекты;
- перерисовывать диаграмму;
- копировать объекты;
- вставлять объекты;
- переходить к заданному объекту.

2.5.4. Описание команд меню Tasks

Команды меню Tasks предназначены для выполнения следующих задач:

- генерация базы данных (описание выполнения данной задачи приведено в п.2.6);
- обратное (реинженеринг) проектирование (см. п.2.3.1);
- синхронизация с базой данных (см. рис.2.1);
- генерация отчетов (см. рис.2.5).

2.5.5. Описание команд меню Client

ERwin выпускается в нескольких различных редакциях, ориентированных на наиболее распространенные средства разработки 4GL. В редакции Erwin, осуществляющей поддержку PowerBuilder фирмы Powersoft, в состав меню

Client входят следующие команды:

- PB Edit Styles - позволяет создать новый или изменить существующий стиль редактирования данных. Этот стиль редактирования используется PowerBuilder для представления данных пользователю. Стиль также определяет, каким способом пользователь может редактировать или выбирать данные;
- PB Display Format - позволяет создать новый или изменить существующий формат представления данных. Формат данных представляет собой фильтр, который позволяет вводить данные только по заданной маске. Маска задается с помощью цифр и букв, где цифры и буквы имеют специальное значение;
- Validation Rule - позволяет создать новое или изменить существующее правило, по которому будет происходить проверка введенных данных. Это правило накладывает определенные ограничения на тип и значение вводимых пользователем данных. В появившемся диалоговом окне можно ввести сообщение, которое будет появляться, когда пользователь базы данных ввел неправильные значения. Можно установить опции, которые определяют, проверяются данные на сервере или в клиентском приложении;
- Valid value - позволяет создать новые или изменить существующие области значений, в соответствии с которыми будут проверяться данные, вводимые пользователем в поля баз данных;
- Default/Initial - позволяет создавать новые или изменять существующие способы заполнения полей базы данных по умолчанию. При добавлении в базу новой записи поля этой записи по умолчанию могут принять одно из созданных значений;
- Target client - позволяет выбрать то средство разработки 4GL, которое будет использовать сгенерированную Erwin-схему (в данном случае выбран программный продукт PowerBuilder);
- PB Sync option - позволяет настроить опции, которые определяют, какие данные и как синхронизировать с PowerBuilder;
- Sync ERWin with PB - выполняет синхронизацию ERWin с PowerBuilder. Для того, чтобы синхронизация прошла успешно, необходимо наличие драйверов ODBC.

2.5.6. Описание команд меню Server

- СУБД Schema properties позволяет определить SQL скрипт, который будет выполняться до соединения и после соединения с базой данных на сервере;
- СУБД Trigger Template позволяет задать вид действия, которое будет выполнено по отношению к главной или подчиненной таблице при определенных действиях пользователя с данными таблицы. Например,

при удалении записи из главной таблицы запись из подчиненной таблицы тоже будет удалена;

- Validation Rule, Valid value, Default/Initial выполняют действия, аналогичные действию соответствующих команд меню Client, но для серверной части;
- Target Server позволяет выбрать необходимую СУБД с помощью диалога, представленного на рис.2.2;
- СУБД Connection позволяет подключиться к серверу СУБД. Для соединения необходимо наличие драйверов.

2.5.7. Описание команд меню Option

Меню Option предоставляет команды, с помощью которых пользователь может по своему желанию настроить опции, влияющие на внешний вид окна ERWin и ER-диаграммы.

2.6. ПРИМЕР РАЗРАБОТКИ МОДЕЛИ В ERWIN

Для начала работы в Erwin необходимо запустить файл MMOPN32.exe в рабочем Erwin-каталоге или дважды щелкнуть на иконке «Logic Works ERwin_ERX 3.0», которая находится на рабочем столе или в папке «ERwin_ERX 3.0».

Ниже на простом примере рассматривается процесс создания новой модели.

Пример.

Ведется учет служащих. Для каждого служащего хранится информация о детях и о списке занимавшихся этим служащим должностей. Для каждой должности хранится информация об окладах, получаемых служащим в период нахождения в данной должности.

Для создания новой модели БД необходимо открыть меню File (см. рис.2.6) и выбрать пункт New... . В появившемся диалоговом окне «Erwin Template selection» необходимо выбрать режим «Blank diagram».

Для более быстрого доступа и дублирования функций меню имеется панель, представляющая собой ряд кнопок, обеспечивающих выбор элементов меню с помощью мыши, а также всплывающее меню.

На первом этапе создается **логический уровень модели**.

Для этого открывается меню Edit (см. рис.2.6) и выбирается пункт Logical model. Во всплывающем меню, появляющемся при нажатии правой кнопки мыши, выбирается режим “Display Level / Entity”. Для отображения линейки

инструментов открывается меню Window (см. рис.2.6) и выбирается пункт ERwin Toolbox. Линейка инструментов изображена на рис.2.7.

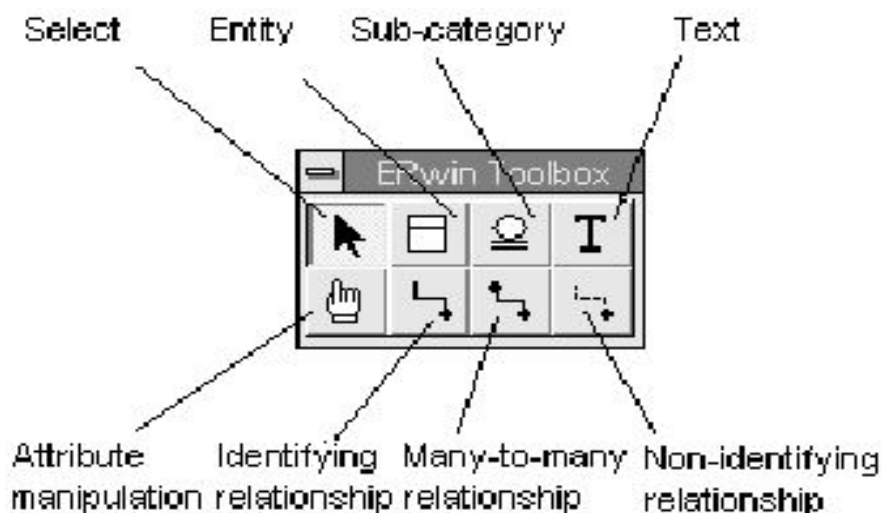


Рис.2.7. Линейка инструментов

Логическая модель создается с помощью следующей последовательности основных шагов.

Шаг 1. Создание сущностей.

Для рассматриваемого примера необходимо при помощи линейки инструментов создать сущности Worker (Служащий), Children (дети), History_Work (история_работы), History_Salary (история_зарплаты).

Для добавления новой сущности с помощью мыши выбирается кнопка Entity (см. рис.2.7). Затем указатель перемещается на то место, где необходимо поместить сущность, и кнопка мыши нажимается еще раз. Данная операция повторяется нужное количество раз по числу создаваемых сущностей, в данном примере – 4 раза. После помещения в основное окно необходимого количества сущностей следует нажать на кнопку Select (см. рис.2.7) для отмены режима выбора сущностей.

Шаг 2. Описание сущностей.

После выбора очередной сущности для нее задается подробное описание в редакторе "Entity Editor". Это описание появится в отчетах ERwin и может быть отображено на диаграмме. Для запуска редактора необходимо открыть меню Edit (см. рис.2.6) и выбрать пункт Entity... . В появившемся диалоговом окне «Entity Editor» (рис.2.8) в поле Definition записываются описания для сущностей.

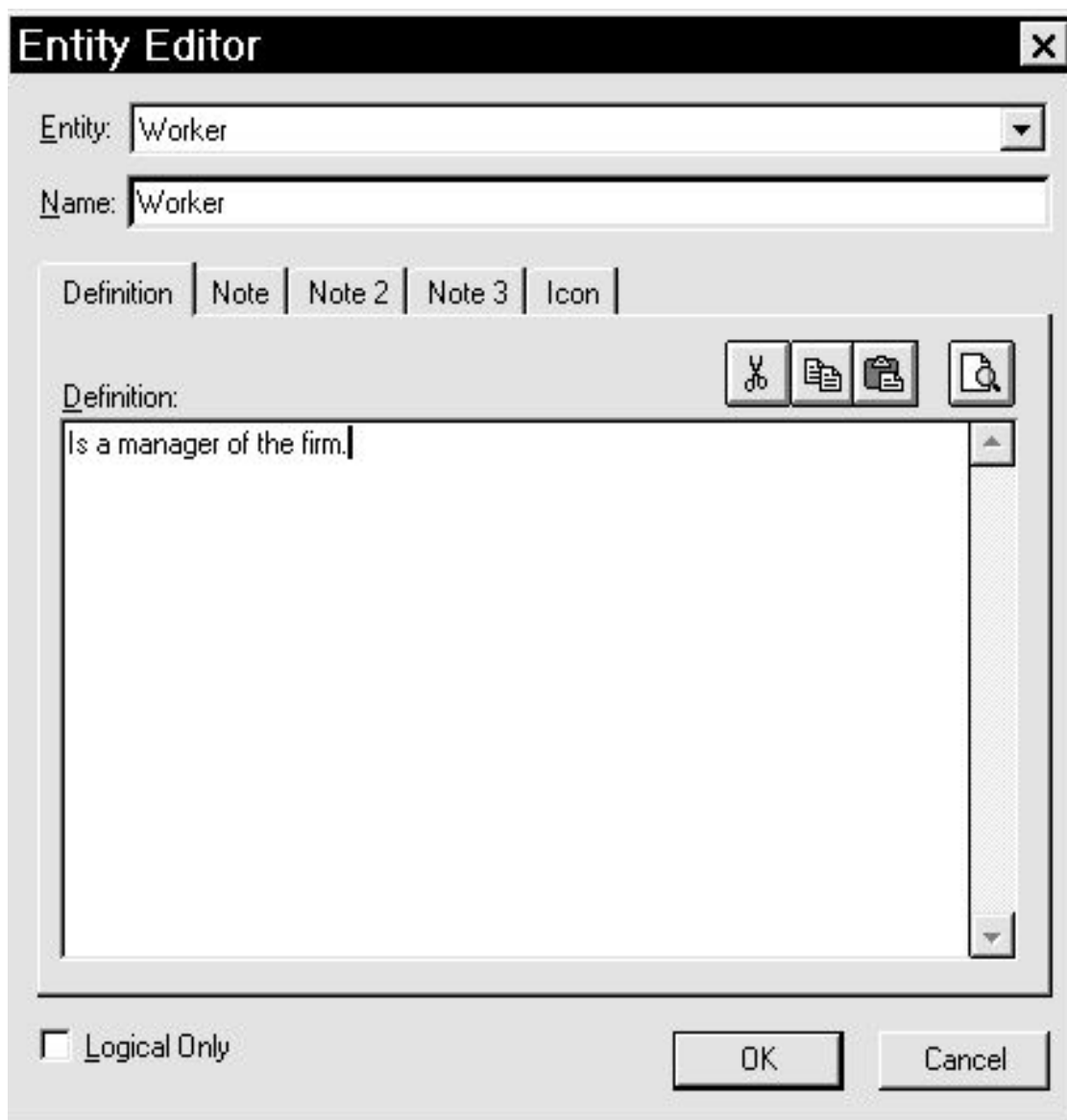


Рис.2.8. Редактор сущностей

Шаг 3. Создание связей между сущностями.

Связи между сущностями задаются при помощи линейки инструментов (см. рис.2.7). Для добавления новых связей на ней имеется три кнопки (Identifying relationship, Many-to-Many relationship, Non-Identifying relationship) с соответствующими функциями (кнопки на линейке инструментов могут изменяться). В рассматриваемом примере все сущности должны быть связаны идентифицирующей связью. Поэтому с помощью мыши необходимо выбрать кнопку Identifying relationship (см. рис.2.7). Затем следует переместить указатель на родительскую сущность, нажать кнопку мыши, переместить указатель на дочернюю сущность, нажать кнопку мыши еще раз. Эту операцию необходимо повторить нужное количество раз по числу связей между сущностями, в данном примере – 3 раза. После завершения связывания

сущностей следует нажать на кнопку Select (см. рис.2.7) для отмены режима выбора связей между сущностями.

Шаг 4. Описание связей между сущностями.

В рассматриваемом примере сущность "Worker" связана идентифицирующей связью "является родителем" с сущностью "Children".

Описание связей выполняется в редакторе Relationship Editor. Для описания связи требуется открыть меню Edit (см. рис.2.6) и выбрать пункт Relationship... . В появившемся диалоговом окне «Relationship Editor» (рис.2.9) должны быть описаны связи между сущностями. Это описание появится в отчетах Erwin.

Relationship Editor

Relationship: Worker является родителем Children

New... Delete

General | Definition | Rolename/RI Actions

Verb Phrase

Parent-to-Child: является родителем

Child-to-Parent:

Relationship Cardinality

Summary: One-to-Zero-One-or-More

Cardinality

☒ Zero, One or More

☐ One or More [P]

☐ Zero or One [Z]

☐ Exactly:

Relationship Type

☒ Identifying

☐ Non-Identifying

Nulls

☒ Nulls Allowed

☐ No Nulls

☐ Logical Only

OK Cancel

Рис.2.9. Редактор связей

Кроме того, связи могут быть прокомментированы (кратко описаны) на ER-диаграмме. Для комментирования связей на линейке инструментов (см. рис.2.7) мышью необходимо выбрать кнопку Text. Затем следует переместить указатель туда, куда желательно поместить комментарий, нажать кнопку мыши и набрать нужный текст (эту операцию повторить нужное количество раз в нужных местах). После окончания комментирования связей необходимо нажать на кнопку Select (см. рис.2.7) для отмены выбора данного режима.

Результат работы (шаги 1 - 4) отображен на диаграмме ERwin (рис.2.10).

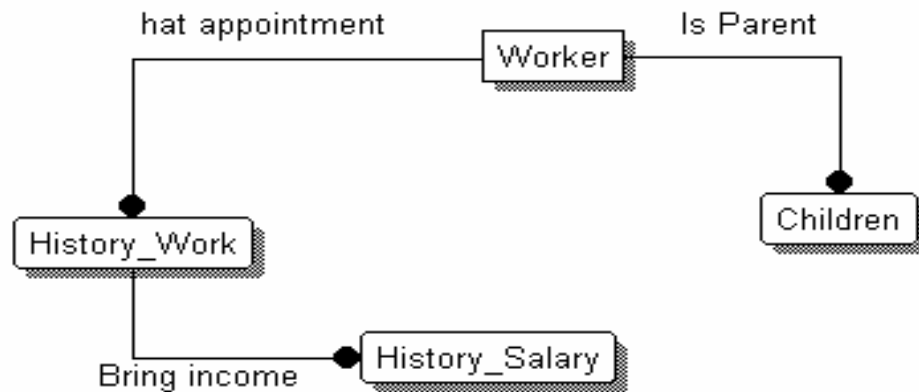


Рис.2.10. Диаграмма уровня сущности

Шаг 5. Описание атрибутов.

Для описания атрибутов сущностей во всплывающем меню выбирается режим “Display Level / Atttribute. В основном меню выбирается пункт Edit и его подпункт Atttribute... . В появившемся редакторе "Attribute Editor" (рис.2.11) задаются имена ключевых (Primary key) и неключевых атрибутов.

Для дочерних сущностей ключевые атрибуты не указываются вручную. ERwin обеспечивает их миграцию из родительской сущности (Foreign Key).

Для лучшего визуального восприятия созданной логической модели во всплывающем меню “Display options/Entities” выбираются все подпункты, кроме “Logical Datatype/Domain”.

Результат создания логической модели изображен на диаграмме ERwin в нотации IDEF1X (рис.2.12).

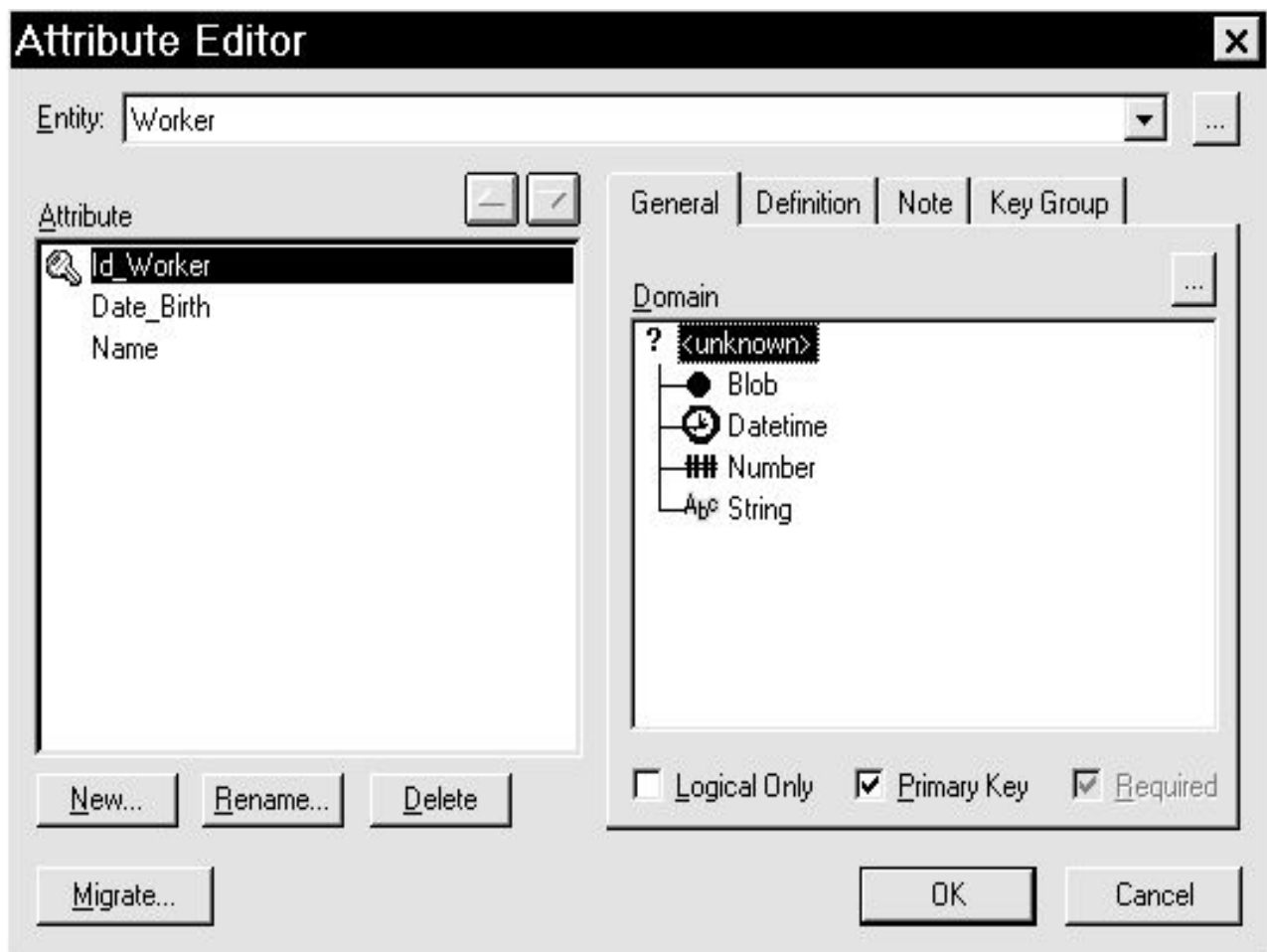


Рис.2.11. Редактор атрибутов

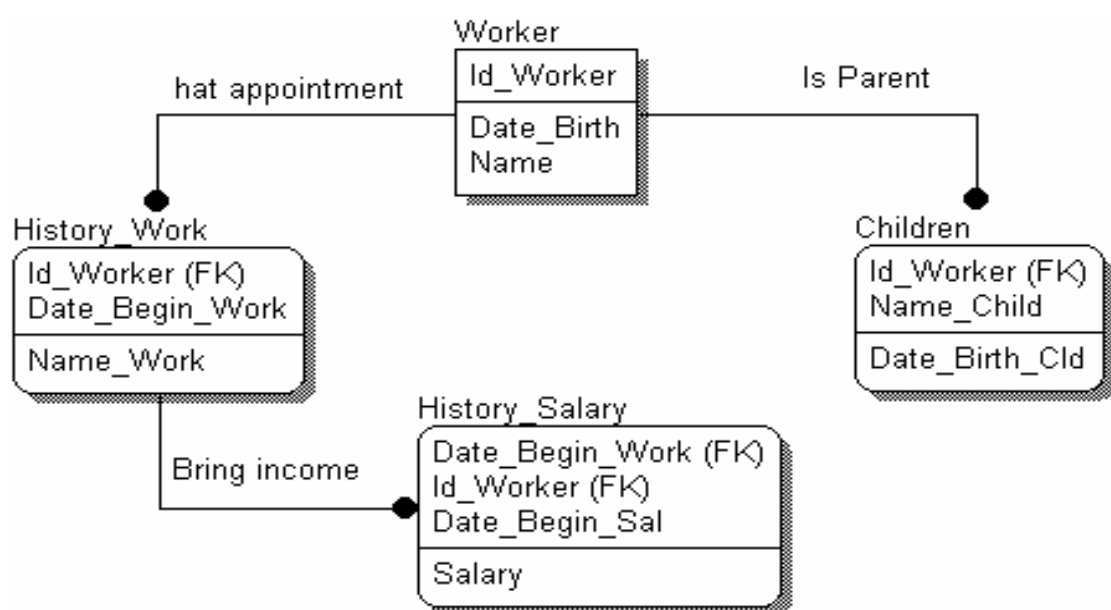


Рис.2.12. Диаграмма уровня атрибутов в нотации IDEF1X

На втором этапе создается **физический уровень модели**.

Для создания физического уровня модели следует открыть меню Edit и выбрать пункт Physical Model.

Физическая модель создается с помощью следующей последовательности основных шагов.

Шаг 1. Определение целевой СУБД.

Для определения целевой СУБД создаваемой базы данных требуется открыть меню Server и выбрать пункт Target Server..., а в появившемся окне (см. рис.2.2) произвести нужные установки. В рассматриваемом примере выбрана СУБД SYBASE.

Шаг 2. Задание типов данных.

Для задания типов данных для колонок таблиц нужно открыть меню Edit (см. рис.2.6) и выбрать пункт Column. В вызванном редакторе “Column Editor” задаются типы данных для колонок таблиц сначала на закладке General, а затем, более детально, на закладке SYBASE.

Диалог, в котором происходит выбор типов данных, приведен на рис.2.13.

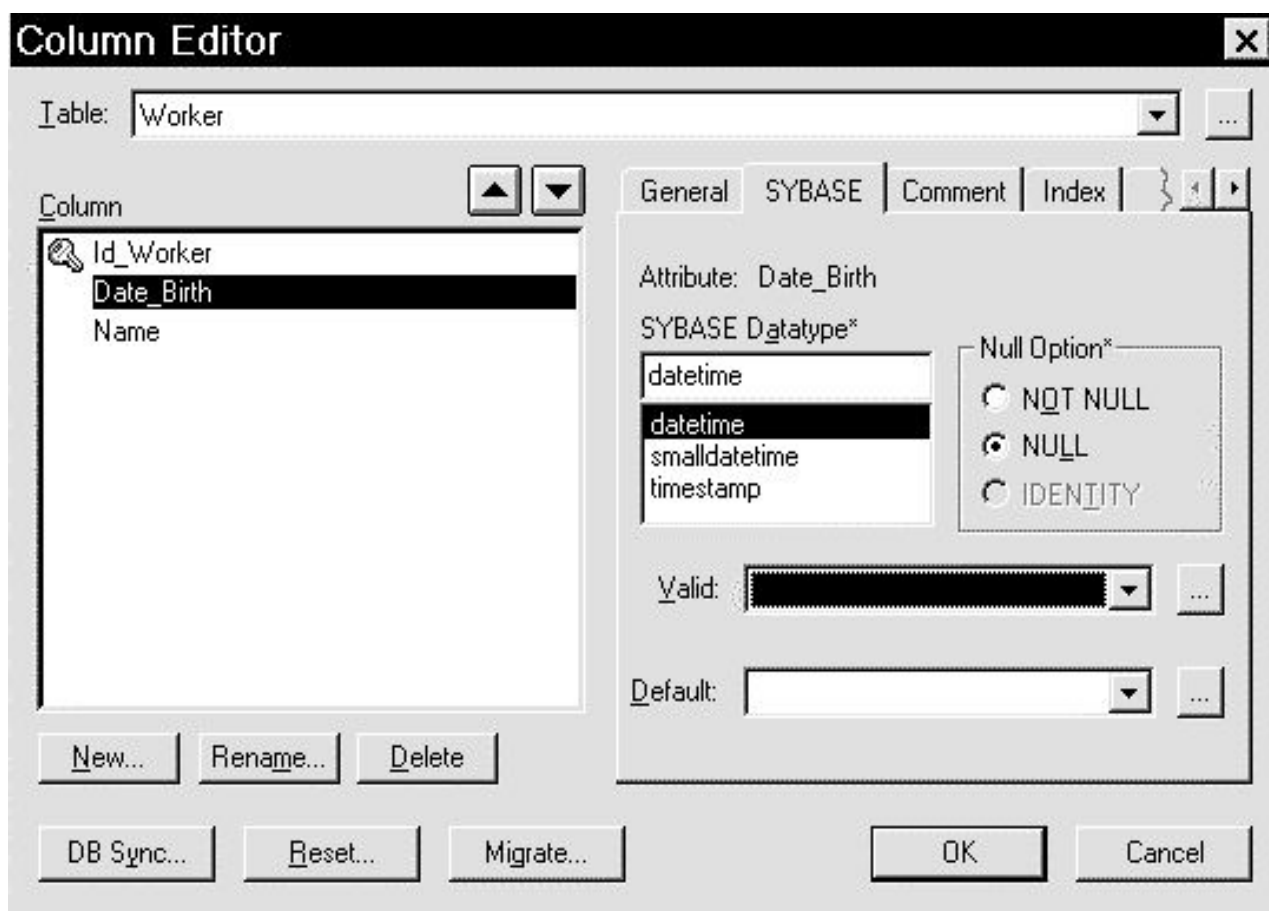


Рис.2.13. Определение типов данных для физической модели

Результат работы отображен на диаграмме ERwin в нотации IDEF1X (рис.2.14).

Шаг 3. Создание базы данных.

Для создания базы данных требуется открыть меню Tasks и выбрать пункт “Forward Engineer/Schema Generation”. В рассматриваемом примере при этом будет выполнена команда "Sybase schema generation". В результате ее выполнения ERwin построит пакет SQL-предложений генерации базы данных.

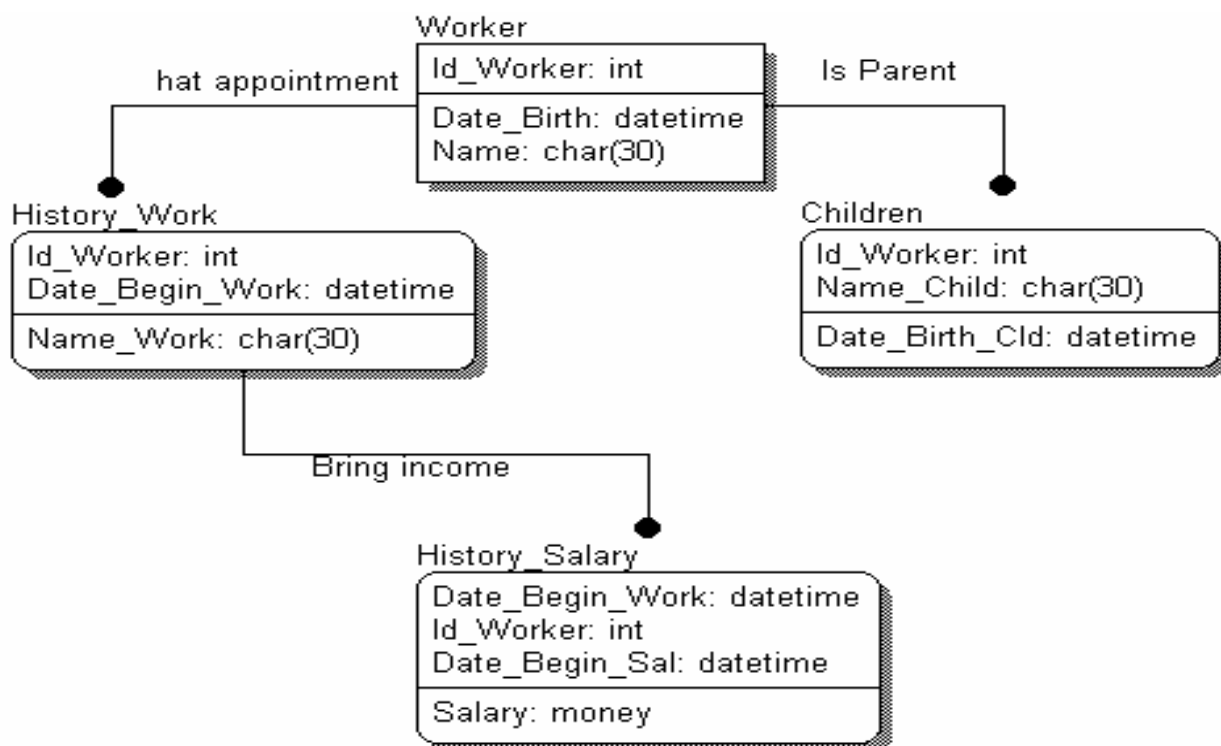


Рис.2.14. Диаграмма физической модели

На рис.2.15 показан диалог выбора параметров генерации БД. Из данного рисунка видно, что может быть задан фильтр (генерация не всех таблиц), пакет SQL-предложений можно просмотреть (preview), распечатать (print), сохранить в файл (report), выполнить генерацию (generate).

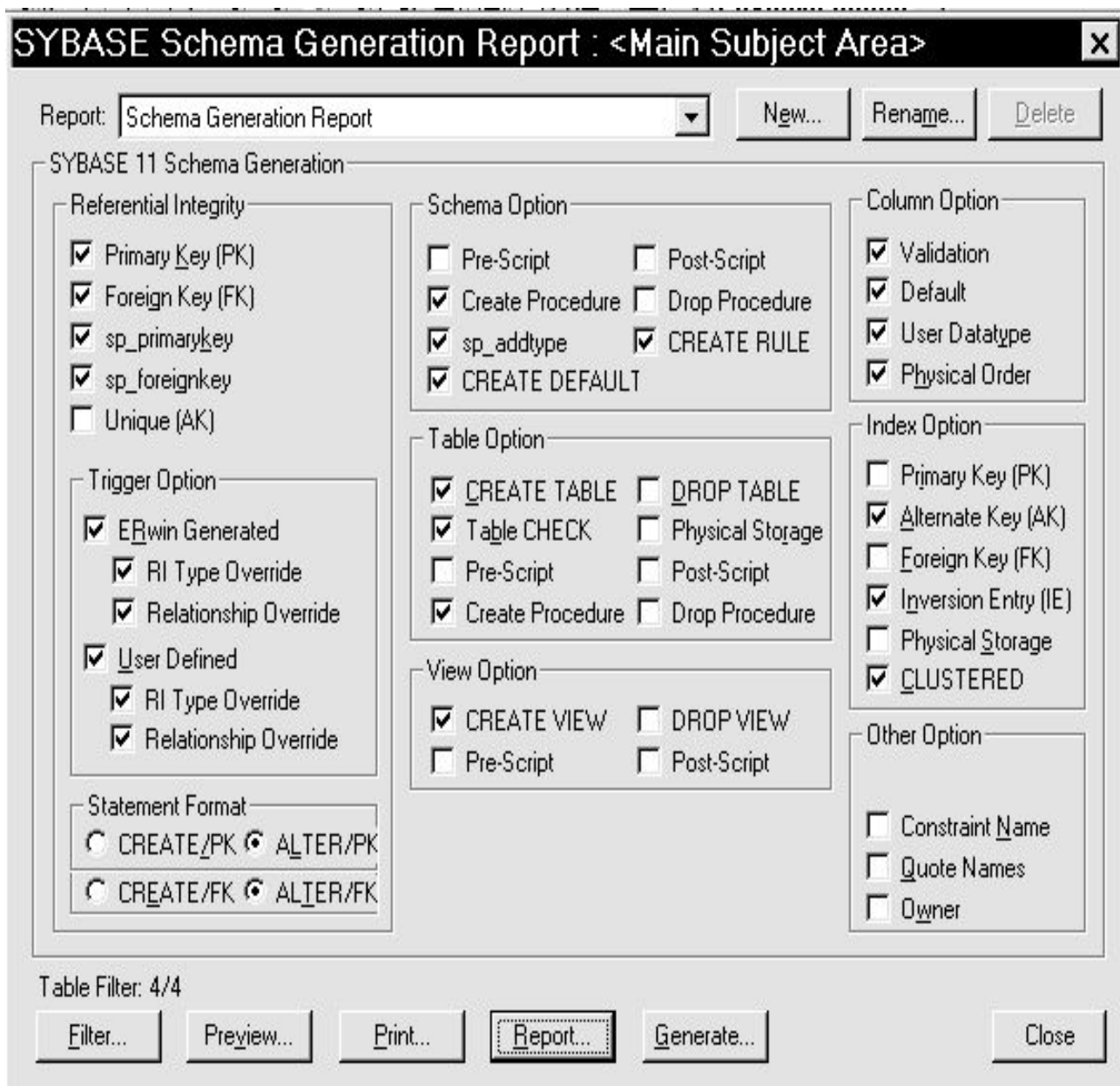


Рис.2.15. Выбор параметров генерации базы данных

2.7. ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНЫХ И ИНДИВИДУАЛЬНЫХ РАБОТ

С помощью Erwin разработать модели баз данных для следующих предметных областей:

- 1) библиотечный каталог;
- 2) отдел кадров университета;
- 3) студенты университета;
- 4) деканат (сотрудники и студенты факультета);

- 5) расписание занятий кафедры;
- 6) учет результатов экзаменационных сессий студентов за весь период обучения;
- 7) общежитие;
- 8) школа;
- 9) делопроизводство предприятия;
- 10) делопроизводство профкома организации;
- 11) учет материальных ценностей (склад);
- 12) магазины города;
- 13) общественный транспорт города;
- 14) клинические больницы города;
- 15) справочная служба аптек;
- 16) справочная бытуслуг;
- 17) бронирование места на ж/д транспорте;
- 18) домоуправление;
- 19) бухгалтерия предприятия;
- 20) учет налогоплательщиков;
- 21) оплата услуг телефонной сети.

Модель должна включать в себя не менее пяти сущностей, связанных между собой.

2.8. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Case-средства. ERwin. Его функции и задачи.
2. Методология IDEF1X.
3. Идентификация сущностей. Сущности в ERwin.
4. Связи в ERwin. Классификация связей.
5. Атрибуты в ERwin. Классификация атрибутов.
6. Логический и физический уровни модели данных в ERwin.
7. Последовательность создания логической модели данных в ERwin.
8. Последовательность создания физической модели данных в ERwin.
9. Альтернативные ключи.
10. Инвертированные индексы.
11. Унификация атрибутов.
12. Связи категоризации.
13. Реализация ссылочной целостности с помощью ERwin.

ЛИТЕРАТУРА К ЛАБОРАТОРНОЙ РАБОТЕ № 2

1. Вендров А. Современные CASE-технологии //
2. Горин С.В., Тандоев А.Ю. Применение CASE-средства Erwin 2.5 для

информационного моделирования в системах обработки данных. -

3. Горин С.В., Тандоев А.Ю. Среда разработки приложений PowerBuilder // DBMS/Russian Edition. - 1995. - № 1.

4. Кодд Е.Ф. Реляционная модель данных для больших совместно используемых банков данных // СУБД. - 1995. - № 1. - с.145-160.

5. Тандоев А.Ю. Архитектура продуктов клиент-сервер фирмы Sybase // СУБД. – 1995. - № 1. - с.62-69.

6. Шлеер С., Меллор С. Объектно-ориентированный анализ: моделирование мира в состояниях. – Киев, 1993.

7. Erwin User's Guide // Logic Works, Inc. – 1989 – 95.

8. Logic Works ERwin/ERX 3.0 - новый продукт, новые возможности // Interface Ltd. - <http://www.interface.ru>

9. Chen P.P. The Entity-Relationship Model: Toward a Unified View of Data // ACM Transactions on Database Systems. - 1976. - V.1., № 1.