

**Министерство образования Республики Беларусь
Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»**

В.А. Никишаев

ОСНОВЫ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ

Практическое пособие

Гомель 2006

**Министерство образования Республики Беларусь
Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»**

В.А. НИКИШАЕВ

ОСНОВЫ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ

**Практическое пособие по выполнению лабораторных работ
для студентов специальности I-53 01 02 –
«Автоматизированные системы обработки информации»**

Гомель 2006

Автор-составитель:

В.А. Никишаев

Рецензент:

Кафедра автоматизированных систем обработки информации
учреждения образования «Гомельский государственный
университет имени Франциска Скорины»

Рекомендовано к изданию научно-методическим советом
учреждения образования «Гомельский государственный
университет имени Франциска Скорины»

Практическое пособие по выполнению лабораторных работ по
курсу «Основы автоматизированного управления» адресовано
студентам специальности I-53 01 02 – «АСОИ» и включает краткие
теоретические сведения по основным темам курса, требования по
выполнению и оформлению лабораторных работ и содержанию
отчета.

© Никишаев В.А., 2006

© Учреждение образования «Гомельский
государственный университет имени
Франциска Скорины», 2006

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
Тема 1. Инструментальные средства создания автоматизированных систем управления.....	4
Лабораторная работа №1.....	20
Тема 2. Декомпозиция производственных процессов по методологии IDEF0.....	30
Лабораторная работа №2.....	41
Тема 3. Диаграммы дерева узлов и диаграммы FEO.....	49
Лабораторная работа №3.....	51
Тема 4. Стоимостный анализ.....	60
Лабораторная работа №4.....	63
ЛИТЕРАТУРА.....	70

ВВЕДЕНИЕ

Выполнение лабораторных работ по учебной дисциплине "Основы автоматизированного управления" предполагает изучение студентами прикладных методик и приобретения практических навыков проектирования автоматизированных систем управления с применением инструментальных средств проектирования автоматизированных систем управления.

Выполнение лабораторных работ включает:

1. Изучение студентами необходимого теоретического материала по теме лабораторной работы.
2. Постановку задачи в соответствии с темой лабораторной работы и согласование ее с руководителем.
3. Построение алгоритма решения задачи и его документирование в разделе «Краткие теоретические сведения» отчета.
4. Выполнение задания.
5. Подготовку отчета о выполненной работе и его защиту.

Структура отчета по лабораторной работе:

1. Тема лабораторной работы.
2. Цель работы.
3. Постановка задачи.
4. Ход выполнения работы.
5. Блок-схема или псевдокод алгоритма решения задачи.
6. Текст программы.
7. Распечатка результатов.
8. Выводы.

Тема 1. Инструментальные средства создания автоматизированных систем управления

Технология создания крупных информационных систем (далее - ИС) предъявляет особые требования к методикам реализации и программным инструментальным средствам, а именно:

- Реализацию крупных проектов принято разбивать на стадии анализа (прежде чем создавать ИС необходимо понять и описать бизнес-логику предметной области), проектирования (необходимо определить модули и архитектуру будущей системы), непосредственного кодирования, тестирования и сопровождения. Известно, что исправление ошибок, допущенных на предыдущей стадии, обходится примерно в десять раз дороже, чем на текущей, откуда следует, что наиболее критичными являются первые стадии проекта. Поэтому крайне важно иметь эффективные средства автоматизации ранних этапов реализации проекта.
- Крупный проект невозможно реализовать в одиночку. Коллективная работа существенно отличается от индивидуальной, поэтому при реализации крупных проектов необходимо иметь средства координации и управления коллективом разработчиков.
- Жизненный цикл создания сложной ИС сопоставим с ожидаемым временем ее эксплуатации. Другими словами, в современных условиях компании перестраивают свои бизнес - процессы примерно раз в два года, столько же требуется (если работать в традиционной технологии) для создания ИС. Может оказаться, что к моменту сдачи ИС она уже никому не нужна, поскольку компания, ее заказавшая, вынуждена перейти на новую технологию работы. Следовательно, для создания крупной ИС жизненно необходим инструмент значительно (в несколько раз) уменьшающий время разработки ИС.

- Вследствие значительного жизненного цикла может оказаться, что в процессе создания системы внешние условия изменились. Обычно внесение изменений в проект на поздних этапах создания ИС - весьма трудоемкий и дорогостоящий процесс. Поэтому для успешной реализации крупного проекта необходимо, чтобы инструментальные средства, на которых он реализуется, были достаточно гибкими к изменяющимся требованиям.

На современном рынке средств разработки ИС достаточно много систем, в той или иной степени удовлетворяющих перечисленным требованиям. Ниже будет рассмотрена вполне конкретная технология разработки, основывающаяся на решениях фирм Logic Works и Rational Software (рис. 1), которая является одной из лучших на сегодняшний день по критерию стоимость/эффективность.

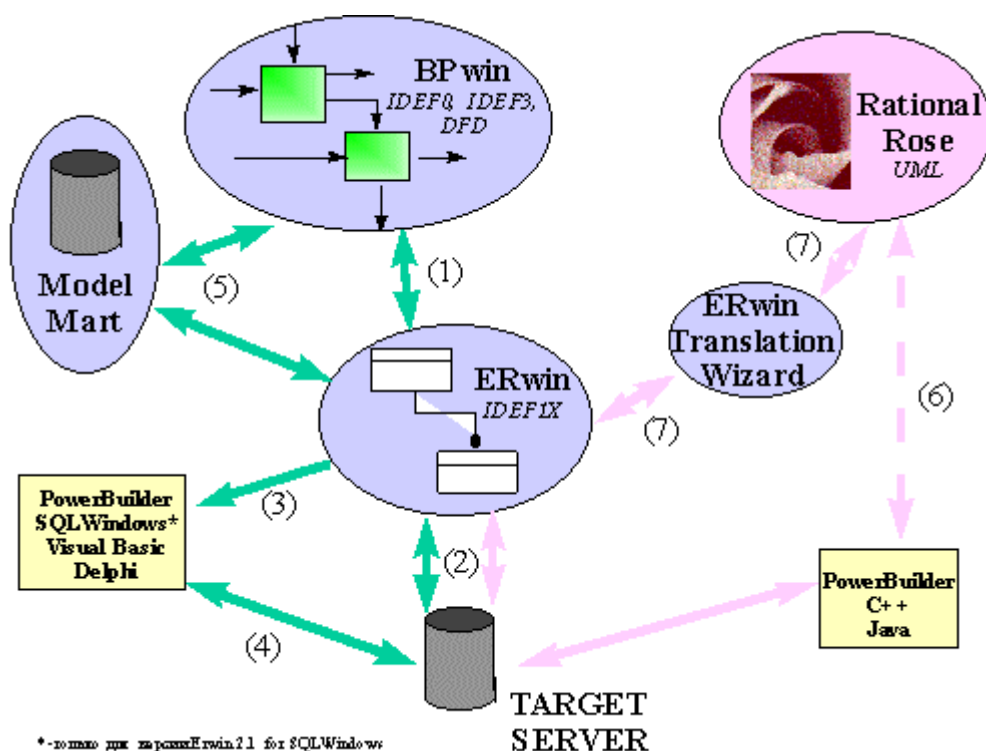


Рис. 1. Общая схема взаимодействия инструментальных средств Logic Works и Rational Software

Для проведения анализа и реорганизации бизнес-процессов Logic Works предлагает CASE - средство верхнего уровня -

VRwin, поддерживающий методологии IDEF0 (функциональная модель), IDEF3 (WorkFlow Diagram) и DFD (DataFlow Diagram). Функциональная модель предназначена для описания существующих бизнес-процессов на предприятии (так называемая модель AS-IS) и идеального положения вещей - того, к чему нужно стремиться (модель TO-BE). Методология IDEF0 предписывает построение иерархической системы диаграмм - единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция - система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности. После каждого сеанса декомпозиции проводится сеанс экспертизы, каждая диаграмма проверяется экспертами предметной области, представителями заказчика, людьми, непосредственно участвующими в бизнес - процессе. Такая технология создания модели позволяет построить модель адекватную предметной области на всех уровнях абстрагирования. Если в процессе моделирования нужно осветить специфические стороны технологии предприятия, VRwin позволяет переключиться на любой ветви модели на нотацию IDEF3 или DFD и создать смешанную модель. Нотация DFD включает такие понятия как внешняя ссылка и хранилище данных, что делает ее более удобной (по сравнению с IDEF0) для моделирования документооборота. Методология IDEF3 включает элемент "перекресток", что позволяет описать логику взаимодействия компонентов системы.

На основе модели VRwin'a можно построить модель данных. Для построения модели данных Logic Works предлагает мощный и удобный инструмент - ERwin. Хотя процесс преобразования модели VRwin в модель данных плохо формализуется и поэтому полностью не автоматизирован, Logic Works предлагает удобный инструмент для облегчения построения модели данных на основе функциональной модели - механизм двунаправленной связи VRwin - ERwin (1, рис.1). ERwin имеет два уровня представления

модели - логический и физический. На логическом уровне данные представляются безотносительно конкретной СУБД, поэтому могут быть наглядно представлены даже для неспециалистов. Физический уровень данных - это, по существу, отображение системного каталога, который зависит от конкретной реализации СУБД. ERwin позволяет проводить процессы прямого и обратного проектирования БД (2, рис.1). Это означает, что по модели данных можно сгенерировать схему БД или автоматически создать модель данных на основе информации системного каталога. Кроме того, ERwin позволяет выравнивать модель и содержимое системного каталога после редактирования того, либо другого. ERwin интегрируется с популярными средствами разработки клиентской части - PowerBuilder, SQLWindows, Visual Basic, Delphi (3, рис.1), что позволяет автоматически генерировать код приложения, который готов к компиляции и выполнению (4, рис.1).

Создание современных информационных систем, основанных на широком использовании распределенных вычислений, объединении традиционных и новейших информационных технологий, требует тесного взаимодействия всех участников проекта: менеджеров, бизнес и системных аналитиков, администраторов баз данных, разработчиков. Для этого используемые на разных этапах и разными специалистами средства моделирования и разработки должны быть объединены общей системой организации совместной работы. Фирма Logic Works разработала систему Model Mart - хранилище моделей, к которому открыт доступ для участников проекта создания информационной системы (5, рис.1). Model Mart удовлетворяет всем требованиям, предъявляемым к средствам разработки крупных информационных систем, а именно:

Совместное моделирование. Каждый участник проекта имеет инструмент поиска и доступа к интересующей его модели в любое время. При совместной работе используются три режима: незащищенный, защищенный и режим просмотра. В режиме просмотра запрещается любое изменение моделей. В защищенном режиме модель, с которой работает один пользователь не может быть изменена другими пользователями.

В незащищенном режиме пользователи могут работать с общими моделями в реальном масштабе времени. Возникающие при этом конфликты разрешаются при помощи специального модуля - Intelligent Conflict Resolution (ICR). В дополнение к стандартным средствам организации совместной работы Model Mart позволяет сохранять множество версий, снабженных аннотациями, с последующим сравнением предыдущих и новых версий. При необходимости возможен возврат к предыдущим версиям.

Создание библиотек решений. Model Mart позволяет формировать библиотеки стандартных решений, включающие наиболее удачные фрагменты реализованных проектов, накапливать и использовать типовые модели, объединяя их при необходимости "сборки" больших систем. На основе существующих баз данных с помощью ERwin возможно восстановление моделей (обратное проектирование), которые в процессе анализа пригодности их для новой системы могут объединяться с типовыми моделями из библиотек моделей.

Управление доступом. Для каждого участника проекта определяются права доступа, в соответствии с которыми они получают возможность работать только с определенными моделями. Права доступа могут быть определены как для групп, так и для отдельных участников проекта. Роль специалистов, участвующих в различных проектах может меняться, поэтому в Model Mart можно определять и управлять правами доступа участников проекта к библиотекам, моделям и даже к специфическим областям модели.

Архитектура Model Mart. Model Mart реализована на архитектуре клиент - сервер. В качестве платформы реализации хранилища выбраны РСУБД Sybase, Microsoft SQL Server и Oracle. Клиентскими приложениями являются ERwin 3.x и BPwin 2.0x. В следующих версиях Model Mart предполагается открыть доступ к хранилищу моделей через API, что позволит постоянно наращивать возможности интегрированной среды путем включения новых инструментов моделирования и анализа.

При разработке крупных проектов критичным становится время реализации проекта. Одним из решений проблемы может стать автоматическая генерация кода приложения (клиентской

части) CASE - средствами на основе модели предметной области. Хотя ERwin решает эту задачу, код генерируется на основе модели IDEF1X, то есть фактически на основе реляционной модели данных, которая непосредственно не содержит информацию о бизнес - процессах. Как следствие этого, сгенерированный код не может полностью обеспечить функциональность приложения со сложной бизнес-логикой. Существует альтернативная технология кодогенерации, которая лишена этого недостатка - объектно-ориентированное проектирование, реализованное в Rational Rose (Rational Software). Rational Rose - позволяющее строить объектные модели в различных нотациях (OMT, UML, Буч) и генерировать на основе полученной модели приложения на языках программирования C++, Visual Basic, Power Builder, Java, Ada, Smalltalk и др. Поскольку генерация кода реализована на основе знаний предметной области, а не на основе реляционной структуры данных, полученный код более полно отражает бизнес-логику. Rational Rose поддерживает не только прямую генерацию кода, но и обратное проектирование, то есть создание объектной модели по исходному коду приложения (6, рис.1).

Rational Rose предназначен для генерации клиентской части приложения. Для генерации схемы БД объектную модель следует конвертировать в модель данных IDEF1X. Модуль ERwin Translation Wizard (Logic Works) позволяет перегружать объектную модель Rational Rose в модель данных ERwin (и обратно) и, с помощью ERwin, сгенерировать схему БД (7, рис.1). Таким образом, технологическая цепочка Rational Rose - ERwin Translation Wizard - ERwin позволяет реализовывать крупные проекты в технологии клиент - сервер.

Инструментальная среда VPwin

VPwin имеет достаточно простой и интуитивно понятный интерфейс пользователя, дающий возможность аналитику создавать сложные модели при минимальных усилиях. Ниже будет описан интерфейс версии 4.0.

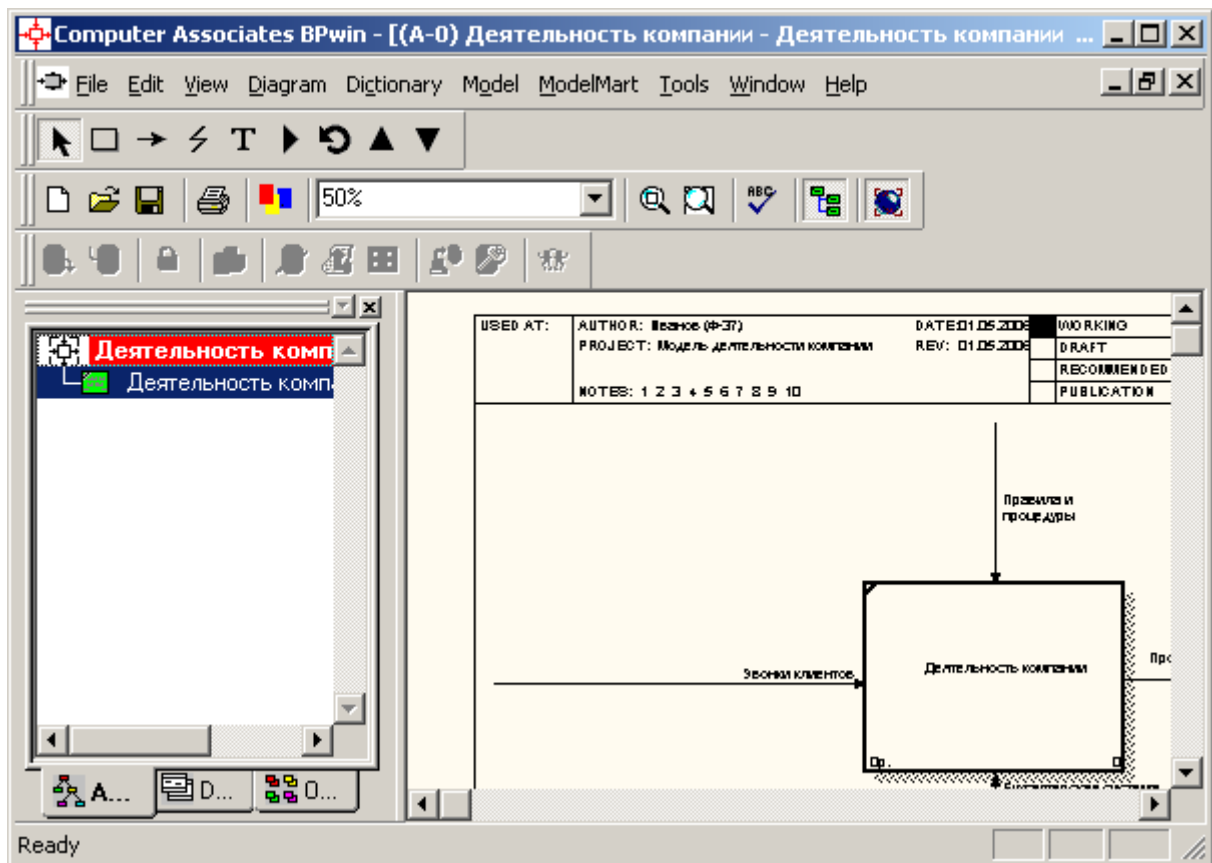


Рис. 2. Интегрированная среда разработки модели VPwin 4.0

При запуске VPwin по умолчанию появляется основная панель инструментов, палитра инструментов (вид которой зависит от выбранной нотации) и, в левой части, навигатор модели - **Model Explorer** (рис. 2).

Функциональность панели инструментов доступна из основного меню VPwin (таблица. 1).

Таблица 1. Основная панель инструментов VPwin 4.0

Элемент управления	Описание	Соответствующий пункт меню
	Создать новую модель	File/New
	Открыть модель	File/Open
	Сохранить модель	File/Save
	Напечатать модель	File/Print
	Выбор масштаба	View/Zoom
	Масштабирование	View/Zoom
	Проверка правописания	Tools/Spelling
	Включение и выключение навигатора модели Model Explorer	View/Model Explorer
	Включение и выключение дополнительной панели инструментов работы с ModelMart	ModelMart

При создании новой модели возникает диалог, в котором следует указать, будет ли создана модель заново, или она будет открыта из файла либо из репозитория ModelMart, внести имя модели и выбрать методологию, в которой будет построена модель (рис. 3).

BPwin поддерживает три методологии - IDEF0, IDEF3 и DFD, каждая из которых решает свои специфические задачи. В BPwin возможно построение смешанных моделей, т. е. модель может содержать одновременно как диаграммы IDEF0, так и IDEF3 и DFD. Состав палитры инструментов изменяется автоматически, когда происходит переключение с одной нотации на другую.

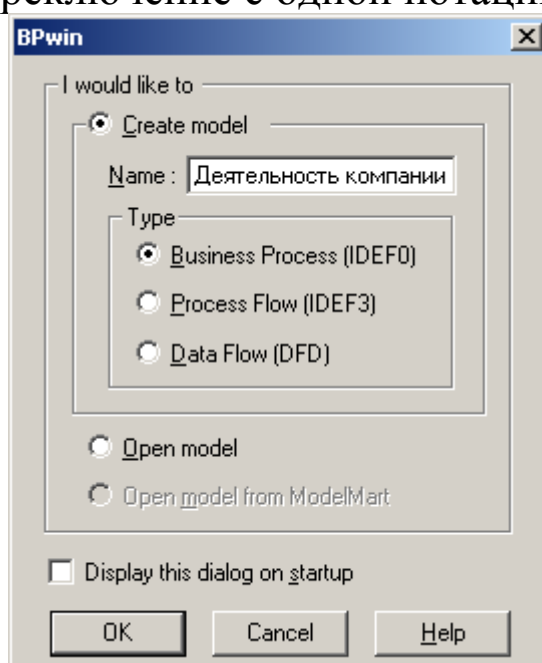


Рис. 3. Диалог создания модели

Модель в BPwin рассматривается как совокупность работ, каждая из которых оперирует с некоторым набором данных. Работа изображается в виде прямоугольников, данные - в виде стрелок. Если щелкнуть по любому объекту модели левой кнопкой мыши, появляется всплывающее контекстное меню, каждый пункт которого соответствует редактору какого-либо свойства объекта.

Установка цвета и шрифта объектов. Пункты контекстного меню **Font Editor** и **Color Editor** вызывают соответствующие диалоги для установки шрифта (в том числе его размера и стиля)

и цвета объекта. Кроме того, PWin позволяет установить шрифт по умолчанию для объектов определенного типа на диаграммах и в отчетах. Для этого следует выбрать меню **Tools/Default Fonts**, после чего появляется каскадное меню, каждый пункт которого служит для установки шрифтов для определенного типа объектов:

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

Принципы построения модели IDEF0

На начальных этапах создания ИС необходимо понять, как работает организация, которую собираются автоматизировать. Никто в организации не знает, как она работает в той мере подробности, которая необходима для создания ИС. Руководитель хорошо знает работу в целом, но не в состоянии вникнуть в детали работы каждого рядового сотрудника. Рядовой сотрудник хорошо знает, что творится на его рабочем месте, но плохо знает, как работают коллеги. Поэтому для описания работы предприятия необходимо построить модель. Такая модель должна быть адекватна предметной области, следовательно, она должна содержать в себе знания всех участников бизнес-процессов организации.

Наиболее удобным языком моделирования бизнес-процессов является IDEF0, предложенный более 20 лет назад Дугласом Россом (SoftTech, Inc.) и называвшийся первоначально SADT - Structured Analysis and Design Technique. (В начале 70-х годов вооруженные силы США применили подмножество SADT, касающееся моделирования процессов, для реализации проектов в рамках программы ICAM (Integrated Computer-Aided Manufacturing). В дальнейшем это подмножество SADT было принято в качестве федерального стандарта США под наименованием IDEF0. Подробные спецификации на стандарты IDEF можно найти на сайте <http://www.idef.com> .

В IDEF0 система представляется как совокупность взаимодействующих работ или функций. Такая чисто функциональная ориентация является принципиальной - функции системы анализируются независимо от объектов, которыми они оперируют. Это позволяет более четко смоделировать логику и взаимодействие процессов организации.

Под моделью в IDEF0 понимают описание системы (текстовое и графическое), которое должно дать ответ на некоторые заранее определенные вопросы.

Моделируемая система рассматривается как произвольное подмножество Вселенной. Произвольное потому, что, во-первых, мы сами умозрительно определяем, будет ли некий объект компонентом системы, или мы будем его рассматривать как внешнее воздействие, и, во-вторых, оно зависит от точки зрения на систему. Система имеет границу, которая отделяет ее от остальной Вселенной. Взаимодействие системы с окружающим миром описывается как вход (нечто, что перерабатывается системой), выход (результат деятельности системы), управление (стратегии и процедуры, под управлением которых производится работа) и механизм (ресурсы, необходимые для проведения работы). Находясь под управлением, система преобразует входы в выходы, используя механизмы.

Процесс моделирования какой-либо системы в IDEF0 начинается с определения контекста, т. е. наиболее абстрактного уровня описания системы в целом. В контекст входит

определение субъекта моделирования, цели и точки зрения на модель.

Под субъектом понимается сама система, при этом необходимо точно установить, что входит в систему, а что лежит за ее пределами, другими словами, мы должны определить, что мы будем в дальнейшем рассматривать как компоненты системы, а что как внешнее воздействие. На определение субъекта системы будет существенно влиять позиция, с которой рассматривается система, и цель моделирования - вопросы, на которые построенная модель должна дать ответ. Другими словами, первоначально необходимо определить область (Scope) моделирования. Описание области как системы в целом, так и ее компонентов является основой построения модели. Хотя предполагается, что в течение моделирования область может корректироваться, она должна быть в основном сформулирована изначально, поскольку именно область определяет направление моделирования и когда должна быть закончена модель. При формулировании области необходимо учитывать два компонента - широту и глубину. Широта подразумевает определение границ модели - мы определяем, что будет рассматриваться внутри системы, а что снаружи. Глубина определяет, на каком уровне детализации модель является завершенной. При определении глубины системы необходимо не забывать об ограничениях времени - трудоемкость построения модели растет в геометрической прогрессии от глубины декомпозиции. После определения границ модели предполагается, что новые объекты не должны вноситься в моделируемую систему; поскольку все объекты модели взаимосвязаны, внесение нового объекта может быть не просто арифметической добавкой, но в состоянии изменить существующие взаимосвязи. Внесение таких изменений в готовую модель является, как правило, очень трудоемким процессом (так называемая проблема "плавающей области").

Цель моделирования (Purpose). Модель не может быть построена без четко сформулированной цели. Цель должна отвечать на следующие вопросы:

- Почему этот процесс должен быть замоделирован?
- Что должна показывать модель?

- Что может получить читатель?

Формулировка цели позволяет команде аналитиков сфокусировать усилия в нужном направлении. Примерами формулирования цели могут быть следующие утверждения: "Идентифицировать и определить текущие проблемы, сделать возможным анализ потенциальных улучшений", "Идентифицировать роли и ответственность служащих для написания должностных инструкций", "Описать функциональность предприятия с целью написания спецификаций информационной системы" и т. д.

Точка зрения (Viewpoint). Хотя при построении модели учитываются мнения различных людей, модель должна строиться с единой точки зрения. Точку зрения можно представить как взгляд человека, который видит систему в нужном для моделирования аспекте. Точка зрения должна соответствовать цели моделирования. Очевидно, что описание работы предприятия с точки зрения финансиста и технолога будет выглядеть совершенно по-разному, поэтому в течение моделирования важно оставаться на выбранной точке зрения. Как правило, выбирается точка зрения человека, ответственного за моделируемую работу в целом. Часто при выборе точки зрения на модель важно задокументировать дополнительные альтернативные точки зрения. Для этой цели обычно используют диаграммы FEO (For Exposition Only).

IDEF0-модель предполагает наличие четко сформулированной цели, единственного субъекта моделирования и одной точки зрения. Для внесения области, цели и точки зрения в модели IDEF0 в VPwin следует выбрать пункт меню **Edit/Model Properties**, вызывающий диалог Model Properties (рис. 4). В закладке **Purpose** следует внести цель и точку зрения, а в закладку **Definition** - определение модели и описание области.

В закладке **Status** того же диалога можно описать статус модели (черновой вариант, рабочий, окончательный и т. д.), время создания и последнего редактирования (отслеживается в дальнейшем автоматически по системной дате). В закладке **Source** описываются источники информации для построения модели (например, "Опрос экспертов предметной области и

анализ документации"). Закладка **General** служит для внесения имени проекта и модели, имени и инициалов автора и временных рамок модели - **AS-IS** и **TO-BE**.

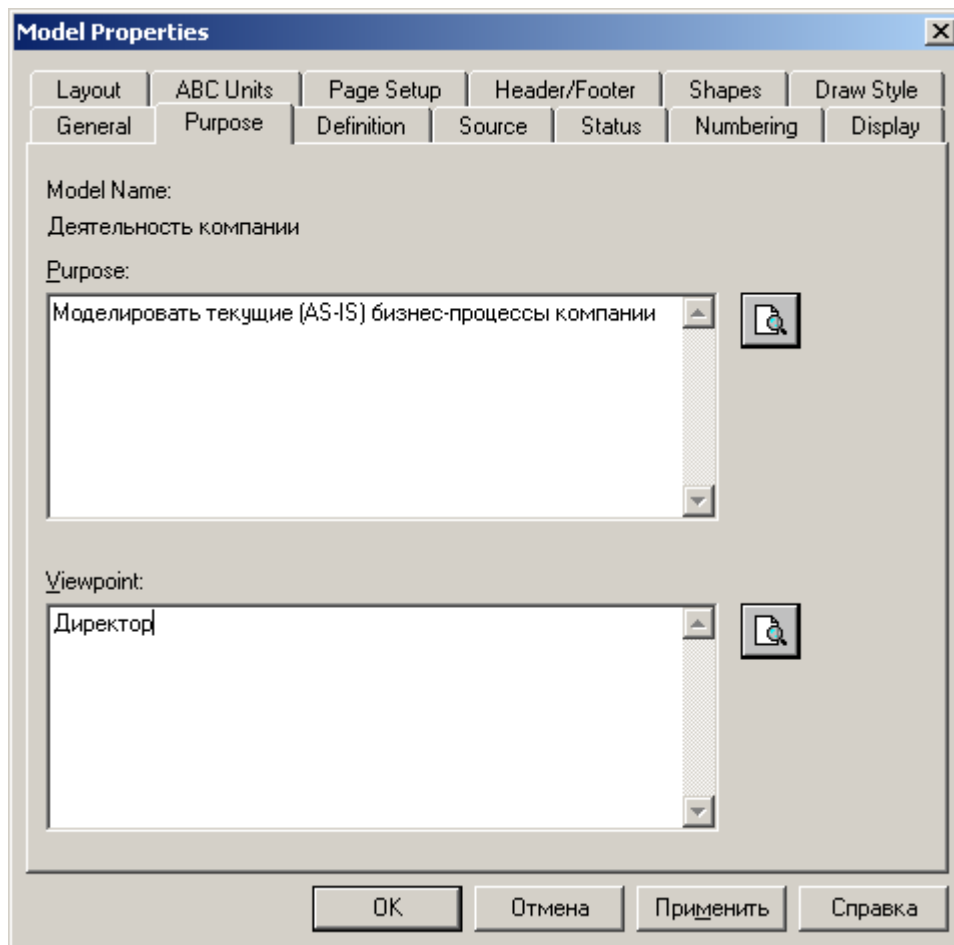


Рис. 4. Диалог задания свойств модели

Модели AS-IS и TO-BE. Обычно сначала строится модель существующей организации работы - AS-IS (как есть). На основе модели AS-IS достигается консенсус между различными единицами бизнеса по тому, "кто что сделал" и что каждая единица бизнеса добавляет в процесс. Модель AS-IS позволяет выяснить, "что мы делаем сегодня" перед тем, как перепрыгнуть на то, "что мы будем делать завтра". Анализ функциональной модели позволяет понять, где находятся наиболее слабые места, в чем будут состоять преимущества новых бизнес-процессов и насколько глубоким изменениям подвергнется существующая структура организации бизнеса. Детализация бизнес-процессов позволяет выявить недостатки организации даже там, где

функциональность на первый взгляд кажется очевидной. Признаками неэффективной деятельности могут быть бесполезные, неуправляемые и дублирующиеся работы, неэффективный документооборот (нужный документ не оказывается в нужном месте в нужное время), отсутствие обратных связей по управлению (на проведение работы не оказывает влияния ее результат), входу (объекты или информация используются нерационально) и т. д. Найденные в модели AS-IS недостатки можно исправить при создании модели TO-BE (как будет) - модели новой организации бизнес-процессов. Модель нужна TO-BE для анализа альтернативных/лучших путей выполнения работы и документирования того, как компания будет делать бизнес в будущем.

Следует указать на распространенную ошибку при создании модели AS-IS - это создание идеализированной модели. Примером может служить создание модели на основе знаний руководителя, а не конкретного исполнителя работ. Руководитель знаком с тем, как предполагается выполнение работы по руководствам и должностным инструкциям и часто не знает, как на самом деле подчиненные выполняют рутинные работы. В результате получается приукрашенная, искаженная модель, которая несет ложную информацию и которую невозможно в дальнейшем использовать для анализа. Такая модель называется SHOULD_BE (как должно бы быть).

Технология проектирования ИС подразумевает сначала создание модели AS-IS, ее анализ и улучшение бизнес-процессов, т. е. создание модели TO-BE, и только на основе модели TO-BE строится модель данных, прототип и затем окончательный вариант ИС. Построение системы на основе модели AS-IS приводит к автоматизации предприятия по принципу "все оставить как есть, только чтобы компьютеры стояли", т. е. ИС автоматизирует несовершенные бизнес-процессы и дублирует, а не заменяет существующий документооборот. В результате внедрение и эксплуатация такой системы приводит лишь к дополнительным издержкам на закупку оборудования, создание программного обеспечения и сопровождение того и другого.

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

Результат описания модели можно получить в отчете **Model Report**. Диалог настройки отчета по модели вызывается из пункта меню **Report/Model Report**. В диалоге настройки следует выбрать необходимые поля, при этом автоматически отображается очередность вывода информации в отчет (рис. 5).

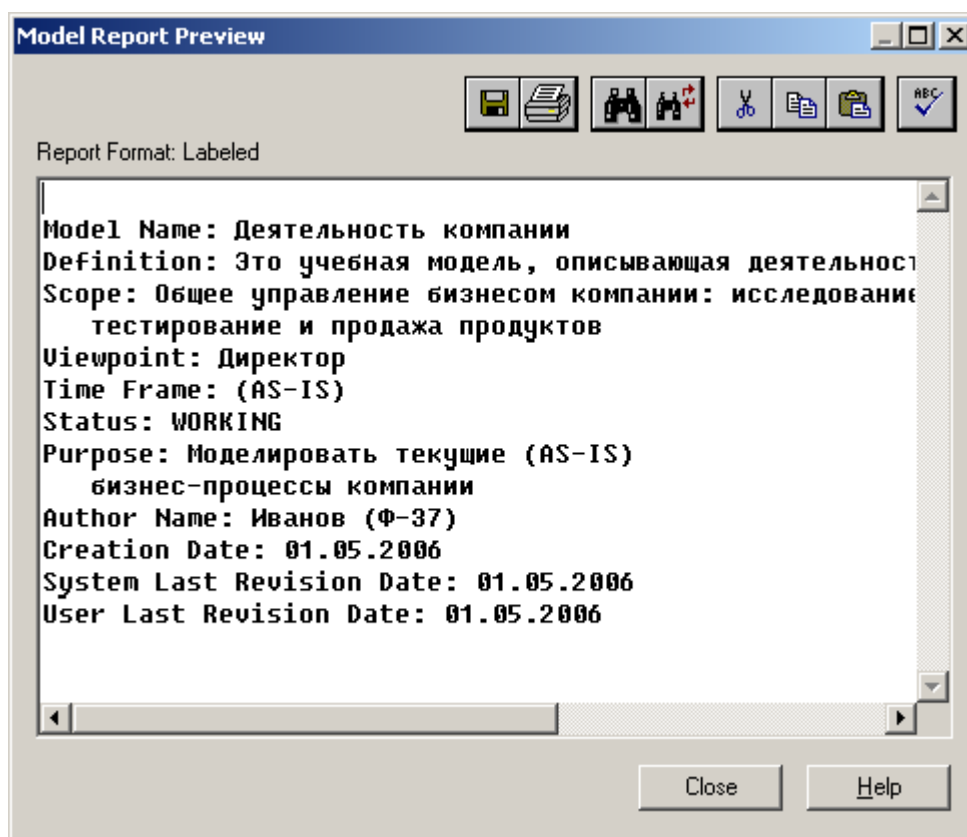


Рис. 5. Отчет по модели

Диаграммы IDEF0. Основу методологии IDEF0 составляет графический язык описания бизнес-процессов. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе.

Модель может содержать четыре типа диаграмм:

- контекстную диаграмму (в каждой модели может быть только одна контекстная диаграмма);
- диаграммы декомпозиции;
- диаграммы дерева узлов;
- диаграммы только для экспозиции (FEO).

Контекстная диаграмма является вершиной древовидной структуры диаграмм и представляет собой самое общее описание системы и ее взаимодействия с внешней средой. После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции. После декомпозиции контекстной диаграммы проводится декомпозиция каждого большого фрагмента системы на более мелкие и так далее, до достижения нужного уровня подробности описания. После каждого сеанса декомпозиции проводятся сеансы экспертизы - эксперты предметной области указывают на соответствие реальных бизнес-процессов созданным диаграммам. Найденные несоответствия исправляются, и только после прохождения экспертизы без замечаний можно приступать к следующему сеансу декомпозиции. Так достигается соответствие модели реальным бизнес-процессам на любом и каждом уровне модели. Синтаксис описания системы в целом и каждого ее фрагмента одинаков во всей модели.

Диаграмма дерева узлов показывает иерархическую зависимость работ, но не взаимосвязи между работами. Диаграмм деревьев узлов может быть в модели сколь угодно много, поскольку дерево может быть построено на произвольную глубину и не обязательно с корня.

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

Лабораторная работа №1

Цель работы: Ознакомиться с методологией IDEF0 создания функциональных моделей промышленного производства. Получить навыки создания и редактирования функциональных моделей в VPwin 4.0.

Постановка задачи: Создать контекстную диаграмму для функциональной модели производственной деятельности согласно выбранному варианту задания.

Результат: Отчет по лабораторной работе с кратким описанием производственного процесса; описанием процедуры создания контекстной диаграммы; распечаткой диаграммы и отчетом по модели.

Варианты заданий

№ варианта	Краткое описание производственной деятельности
1	Обслуживание и ремонт копировальной техники
2	Обслуживание и ремонт бытовой техники
3	Обслуживание и ремонт автомобилей
4	Сборка и продажа автомобилей
5	Сборка и продажа телевизоров
6	Сборка и продажа мебели
7	Строительно-монтажные работы
8	Сборка и установка окон и дверей
9	Тюнинг автомобилей под заказ
0	Индивидуальный пошив одежды

Пример выполнения работы

В качестве примера рассматривается деятельность вымышленной компании «Computer Word». Компания занимается

в основном сборкой и продажей настольных компьютеров и ноутбуков. Компания не производит компоненты самостоятельно, а только собирает и тестирует компьютеры.

Основные виды работ в компании таковы:


- продавцы принимают заказы клиентов;
- операторы группируют заказы по типам компьютеров;
- операторы собирают и тестируют компьютеры;
- операторы упаковывают компьютеры согласно заказам;
- кладовщик отгружает клиентам заказы.

Компания использует лицензионную бухгалтерскую информационную систему, которая позволяет оформить заказ, счет и отследить платежи по счетам.

Методика выполнения работы

1. Запустите BPwin ( BPwin 4.0).

2. Если появляется диалог **ModelMart Connection Manager**, нажмите на кнопку **Cancel** (Отмена).

3. Щелкните по кнопке . Появляется диалоговое окно **I would like to** (рис. 6). Внесите в текстовое поле **Name** имя модели "Деятельность компании" и выберите Type – **Business Process (IDEF0)**. Нажмите кнопку **OK**.

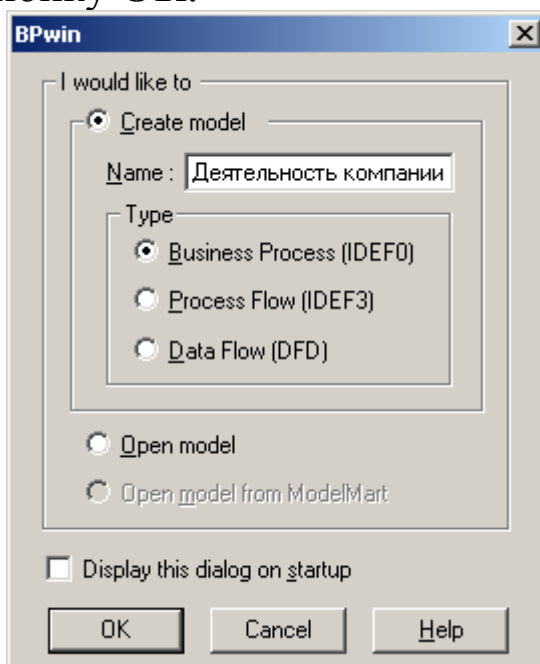


Рис. 6. Присвоение модели имени и выбор типа модели

4. Откроется диалоговое окно **Properties for New Models** (Свойства новой модели) (рис. 7). Введите в текстовое поле **Author** (Автор) имя автора модели и в текстовое поле **Author initials** его инициалы. Нажмите последовательно кнопки **Apply** и **OK**.

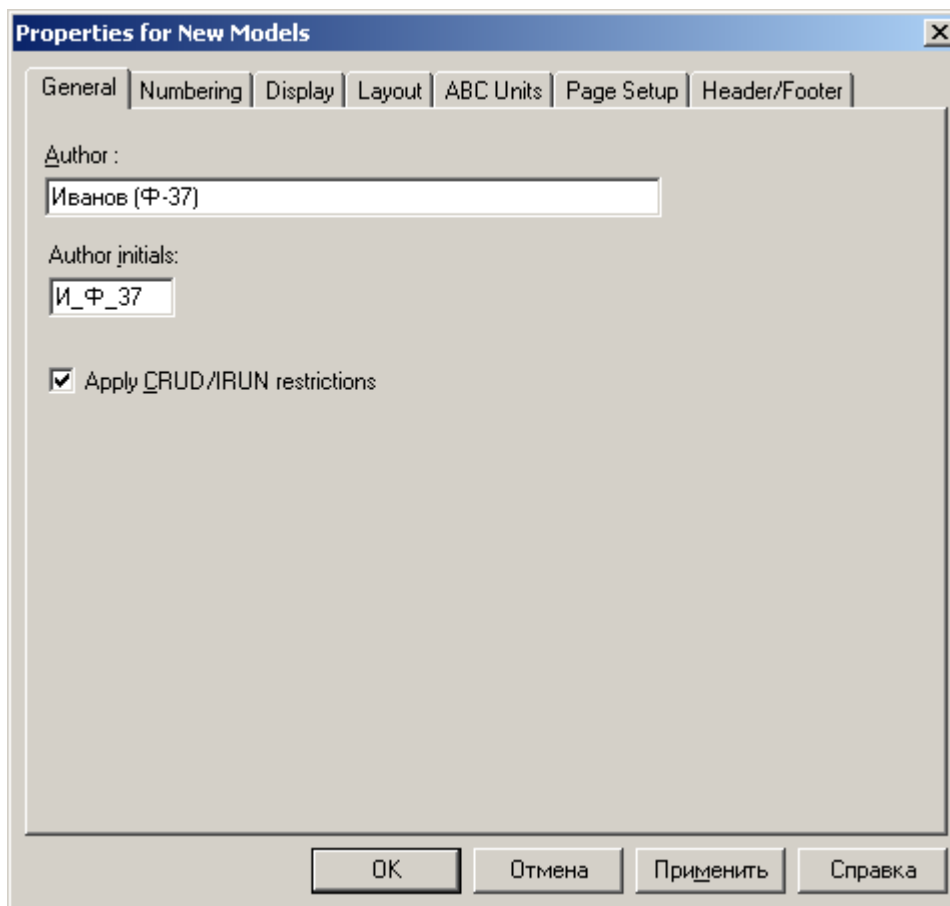






Рис. 7. Ввод имени автора модели и его инициалов

5. Автоматически создается незаполненная контекстная диаграмма (рис. 8).

6. Обратите внимание на кнопку  на панели инструментов. Эта кнопка включает и выключает инструмент просмотра и навигации - **Model Explorer** (Браузер модели). **Model Explorer** имеет три вкладки – **Activities** ( Act...), **Diagrams** ( Dia...) и **Objects** ( Obj...). Во вкладке **Activities** щелчок правой кнопкой по объекту в браузере модели позволяет выбрать опции редактирования его свойств (рис. 9).

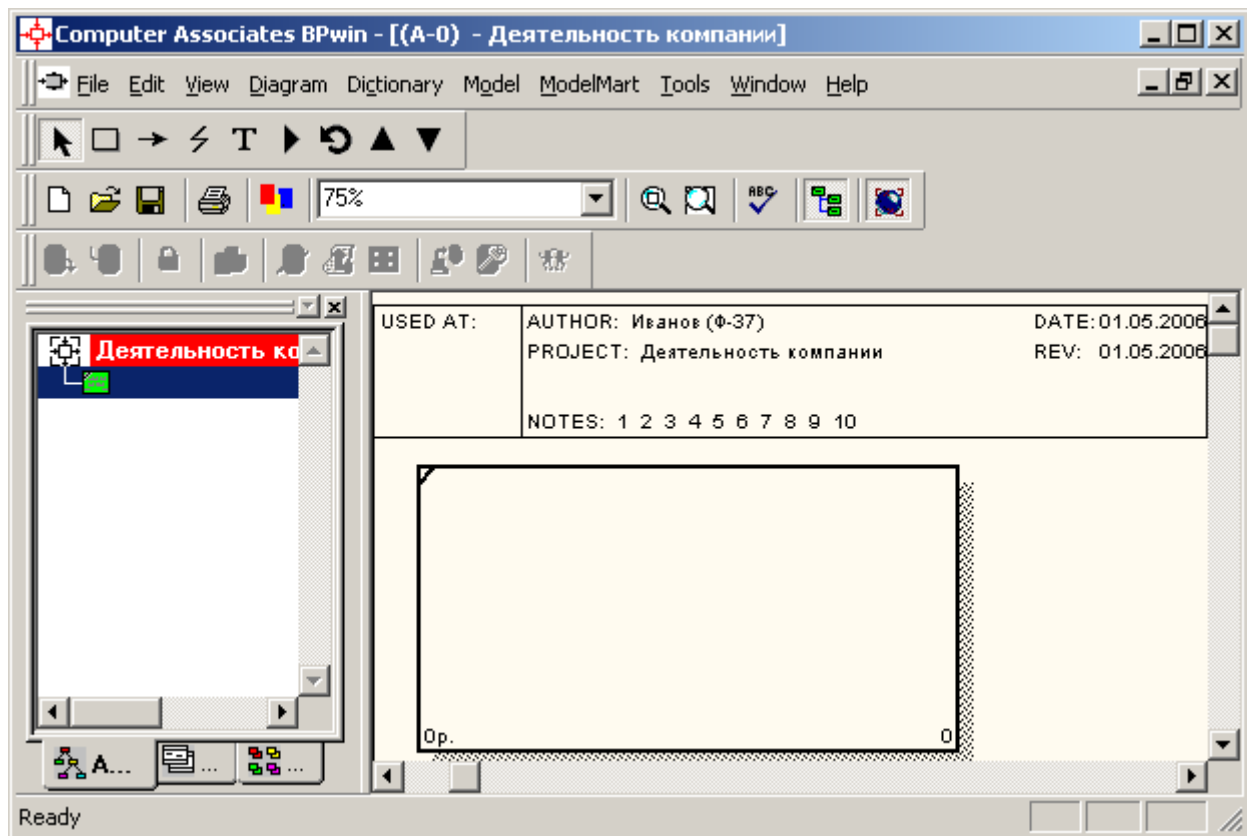


Рис. 8. Незаполненная контекстная диаграмма

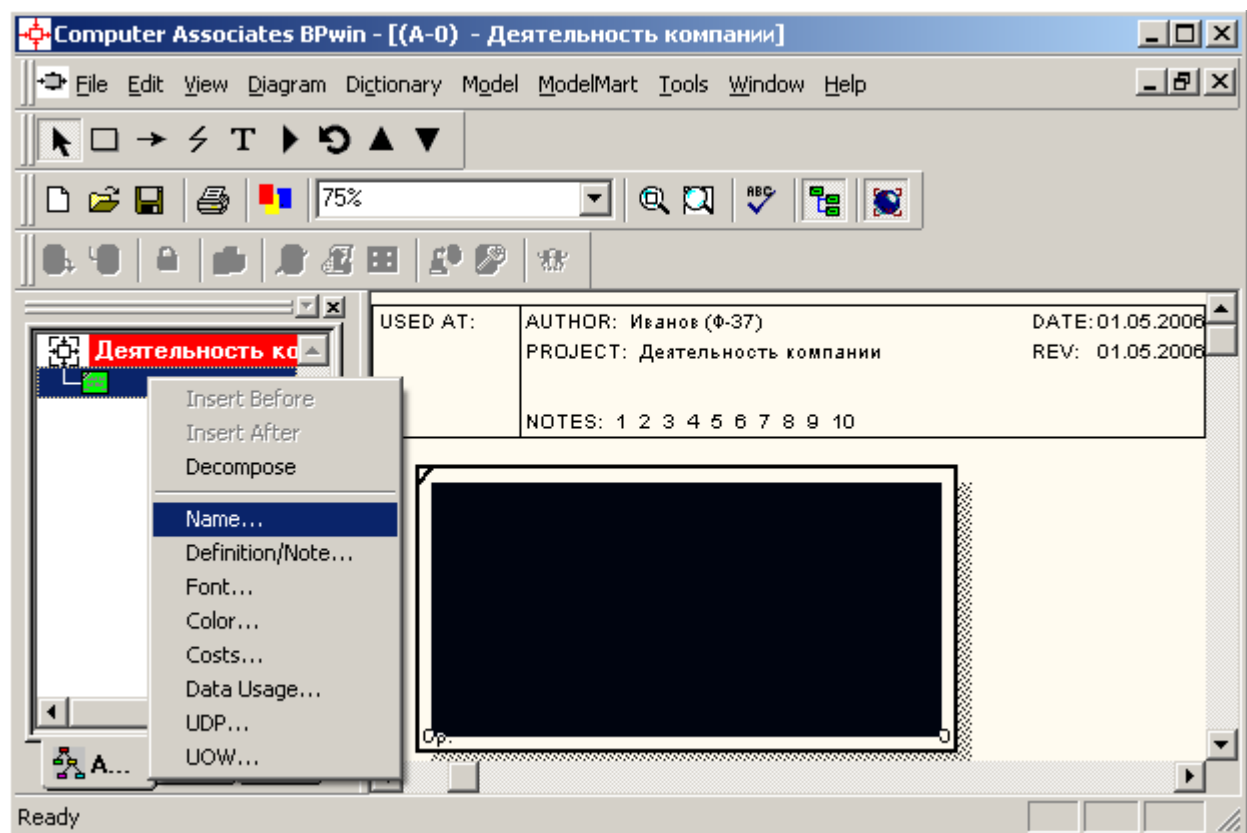


Рис. 9. Щелчок правой кнопкой по объекту во вкладке Activities позволяет воспользоваться контекстным меню для редактирования его свойств

7. Перейдите в меню **Model/Model Properties**. Во вкладке **General** диалогового окна **Model Properties** в текстовое поле **Model name** следует внести имя модели "Деятельность компании", а в текстовое поле **Project** имя проекта "Модель деятельности компании", и, наконец, в текстовое **Time Frame** (Временной охват) - **AS-IS** (Как есть) (рис. 10).

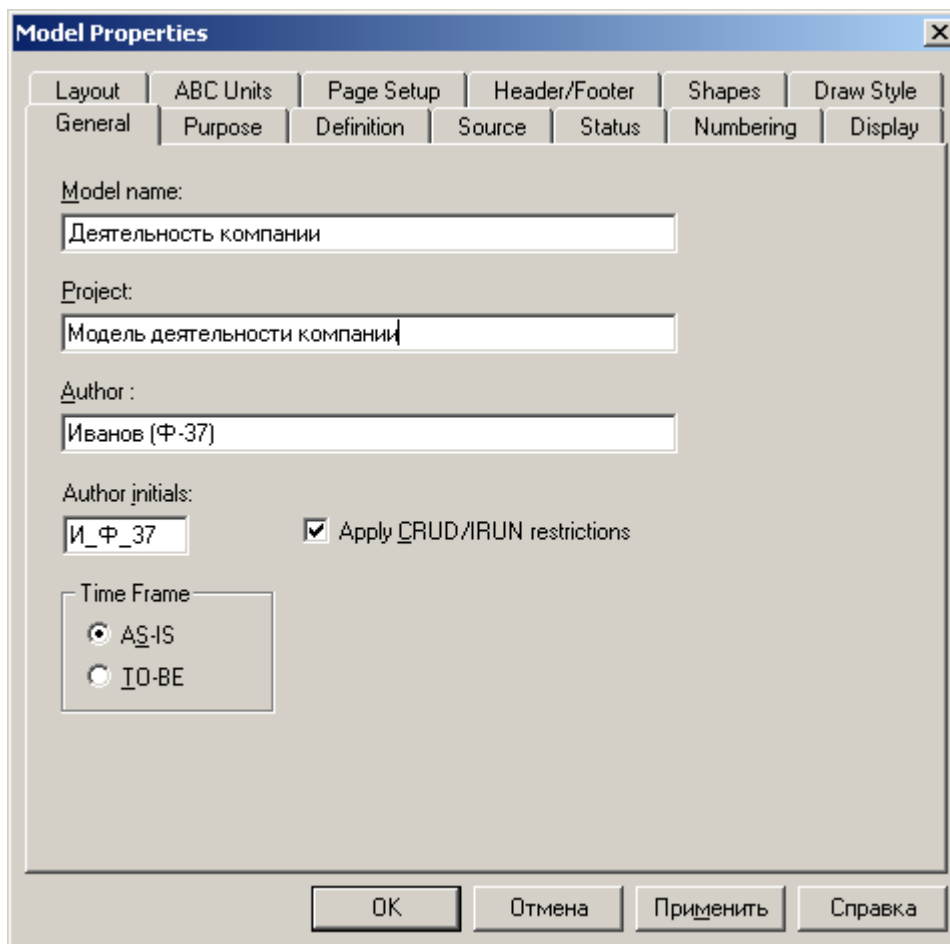


Рис. 10. Окно задания свойств модели

8. Во вкладке **Purpose** диалогового окна **Model Properties** в текстовое поле **Purpose** (цель) внесите данные о цели разработки модели - "Моделировать текущие (AS-IS) бизнес-процессы компании", а в текстовое поле **Viewpoint** (точка зрения) - "Директор" (рис. 11).

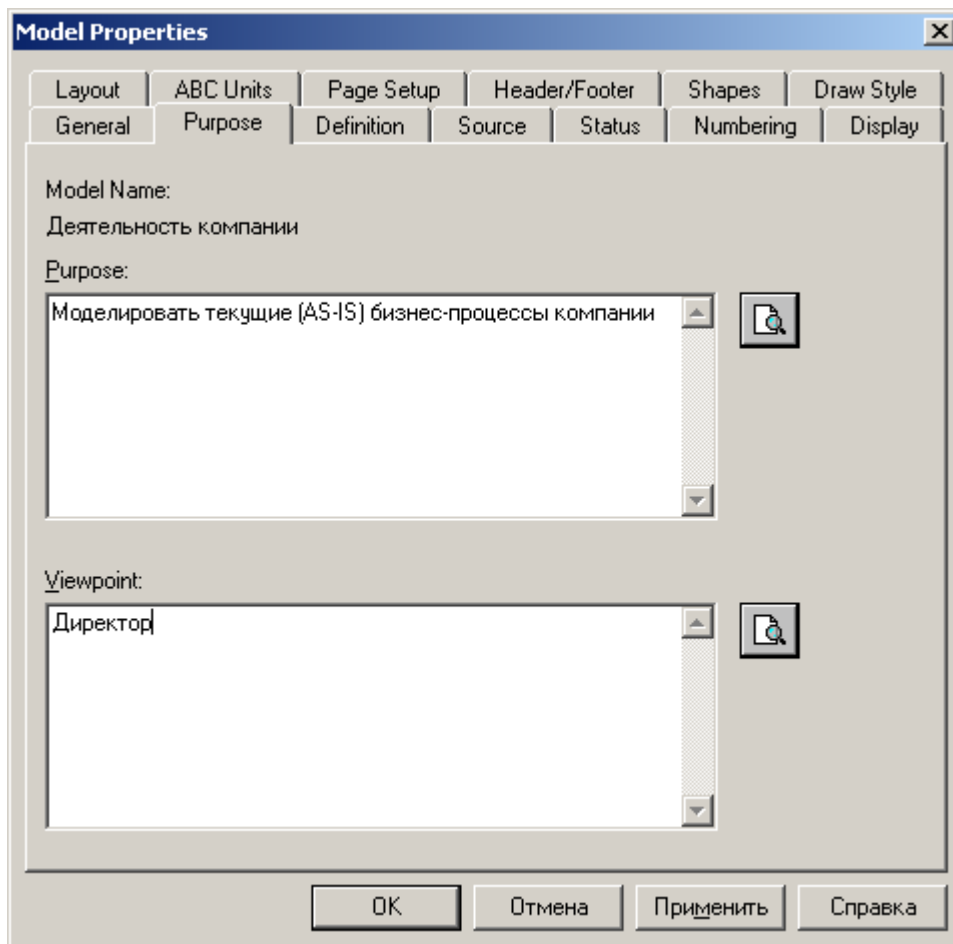


Рис. 11. Внесение данных о цели моделирования и точке зрения

9. Во вкладке **Definition** диалогового окна **Model Properties** в текстовое поле **Definition** (Определение) внесите "Это учебная модель, описывающая деятельность компании" и в текстовое поле **Scope** (охват) - "Общее управление бизнесом компании: исследование рынка, закупка компонентов, сборка, тестирование и продажа продуктов" (рис. 12).

10. Перейдите на контекстную диаграмму и правой кнопкой мыши щелкните по прямоугольнику представляющему, в нотации **IDEF0**, условное графическое обозначение работы. В контекстном меню выберите опцию **Name** (рис. 13). Во вкладке **Name** внесите имя "Деятельность компании" (рис. 14).

11. Во вкладке **Definition** диалогового окна **Activity Properties** в текстовое поле **Definition** (Определение) внесите "Текущие бизнес-процессы компании" (рис. 15). Текстовое поле **Note** (Примечания) оставьте незаполненным.

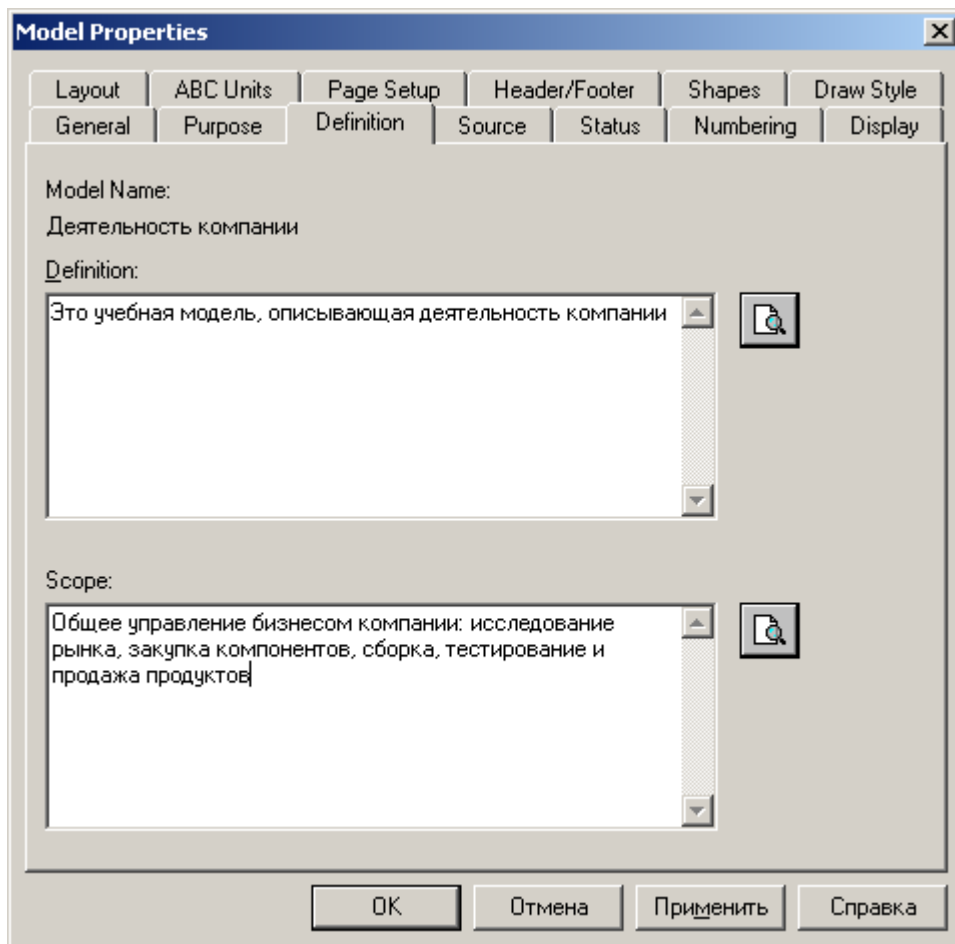


Рис. 12. Внесение дополнительных данных определяющих модель

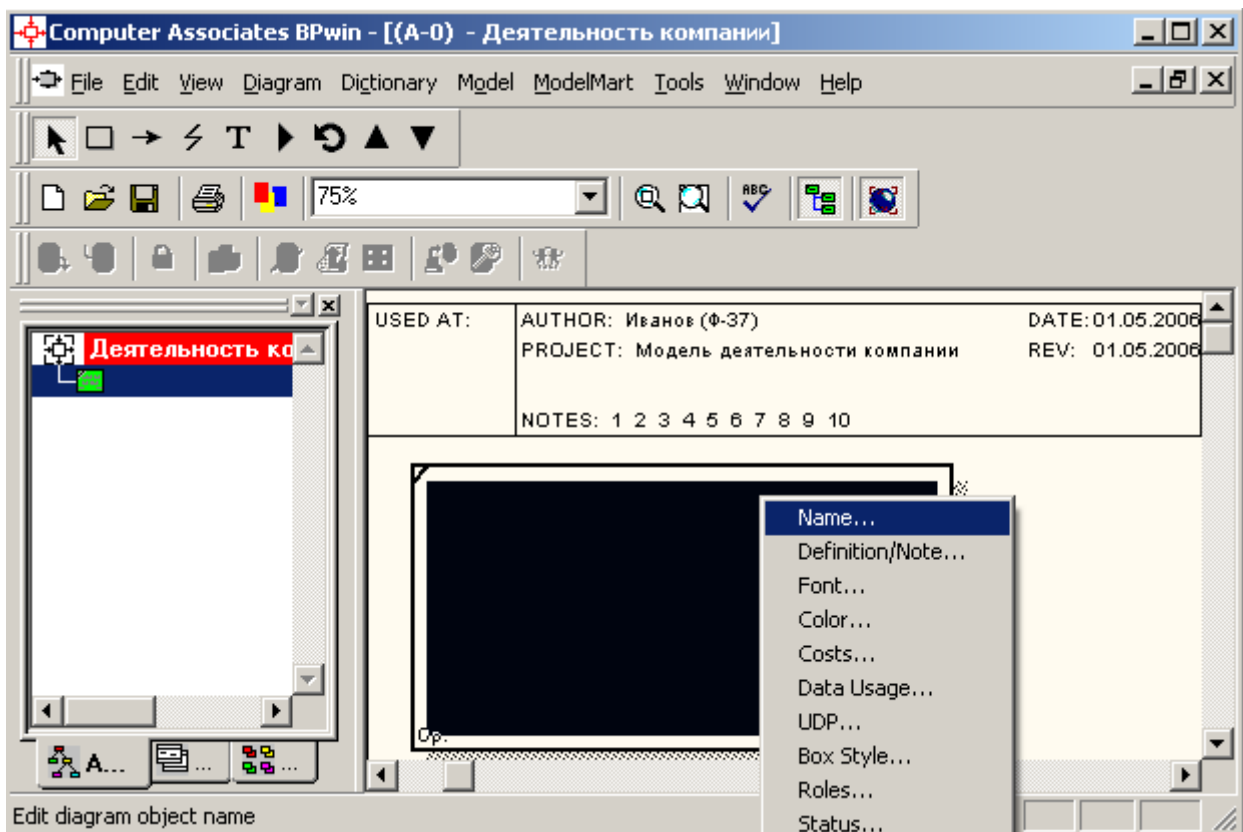


Рис. 13. Контекстное меню для работы с выбранной опцией Name

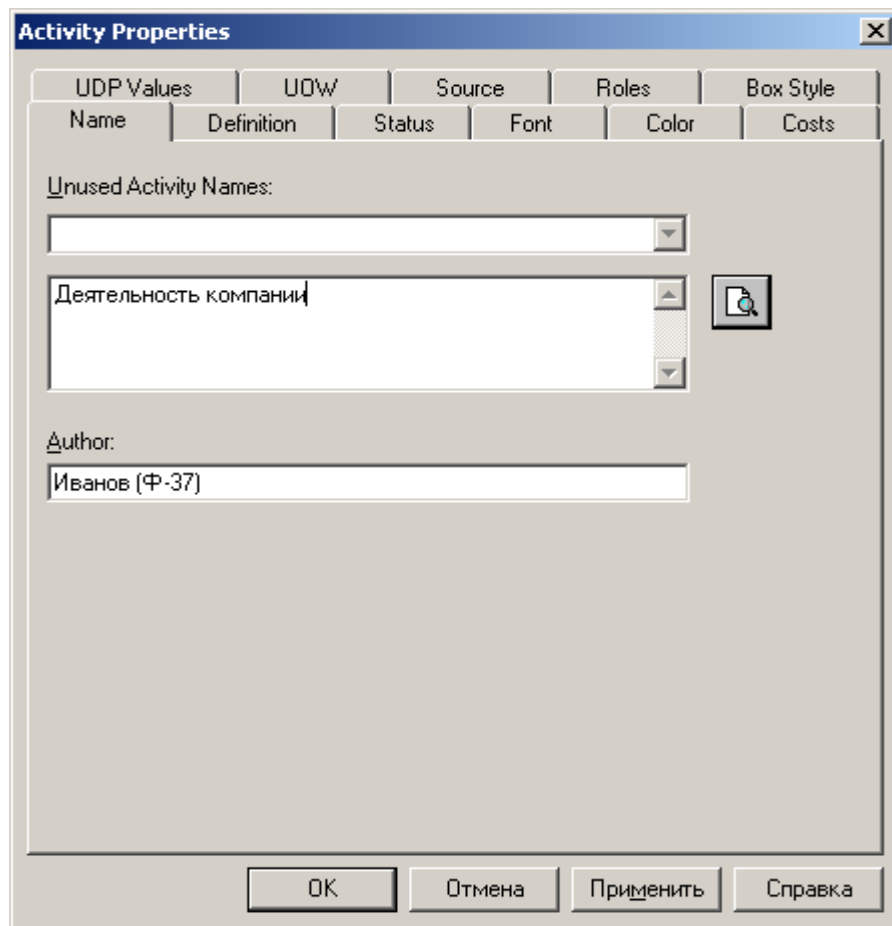


Рис. 14. Присвоение работе названия

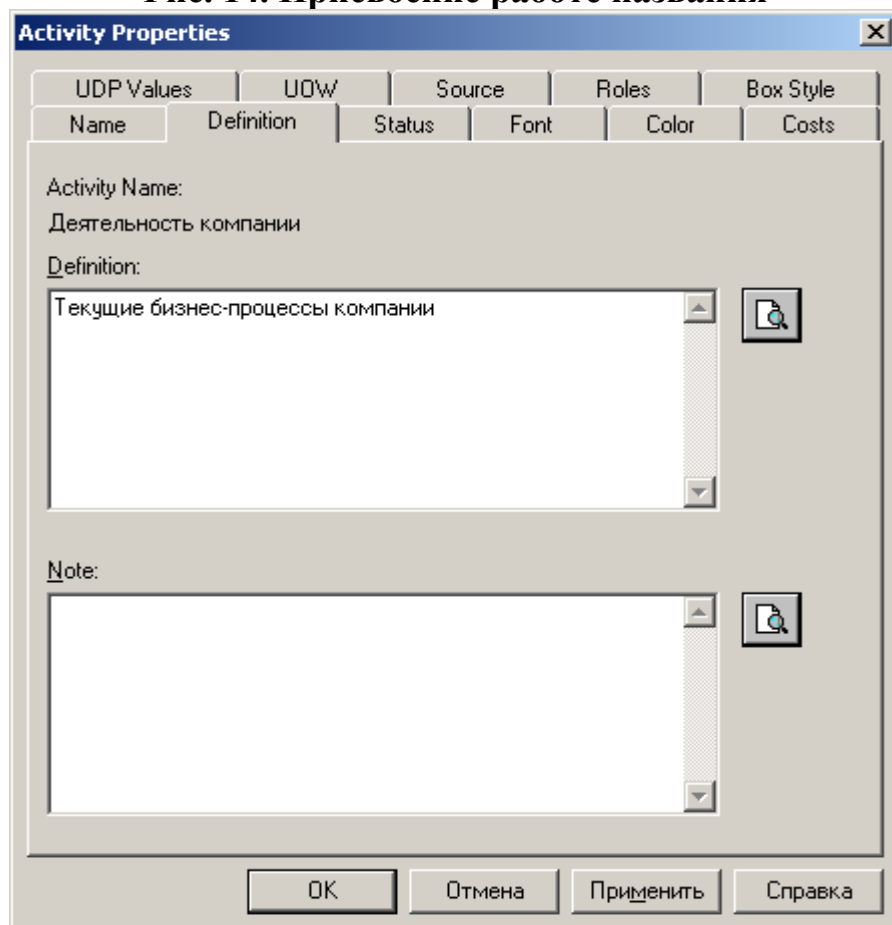


Рис. 15. Внесение дополнительных данных о работе

12. Создайте **ICOM**-стрелки на контекстной диаграмме (таблица 1).

Таблица 1 - Стрелки контекстной диаграммы

Название стрелки (Arrow Name)	Определение стрелки (Arrow Definition)	Тип стрелки (Arrow Type)
Звонки клиентов	Запросы информации, заказы, техподдержка и т.д.	Input
Правила и процедуры	Правила продаж, инструкции по сборке, процедуры тестирования, критерии производительности и т. д.	Control
Проданные продукты	Настольные и портативные компьютеры	Output
Бухгалтерская система	Оформление счетов, оплата счетов, работа с заказами	Mechanism

13. С помощью кнопки **T** внесите текст в поле диаграммы - точку зрения и цель (рис. 16).

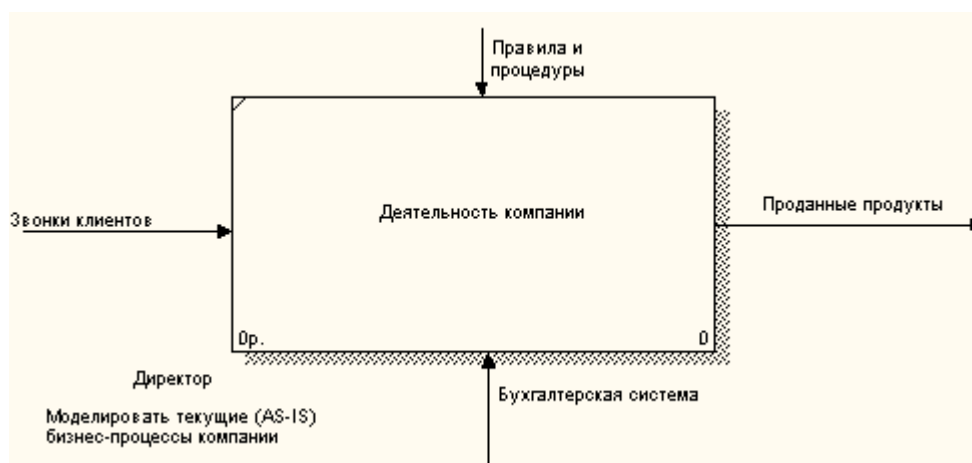


Рис. 16. Внесение текста в поле диаграммы с помощью редактора Text Block Editor

14. Создайте отчет по модели. В меню **Tools/Reports/Model Report** (рис. 17) задайте опции генерирования отчета (установите галочки) и нажмите кнопку **Preview** (Предварительный просмотр) (рис. 18).

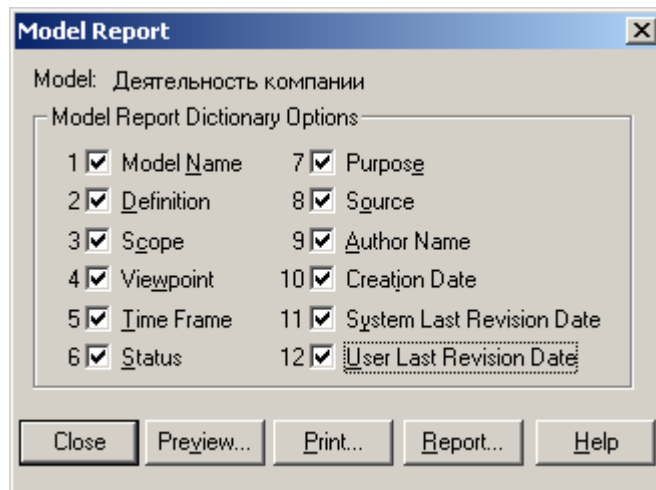


Рис. 17. Задание опций генерирования отчета Model Report

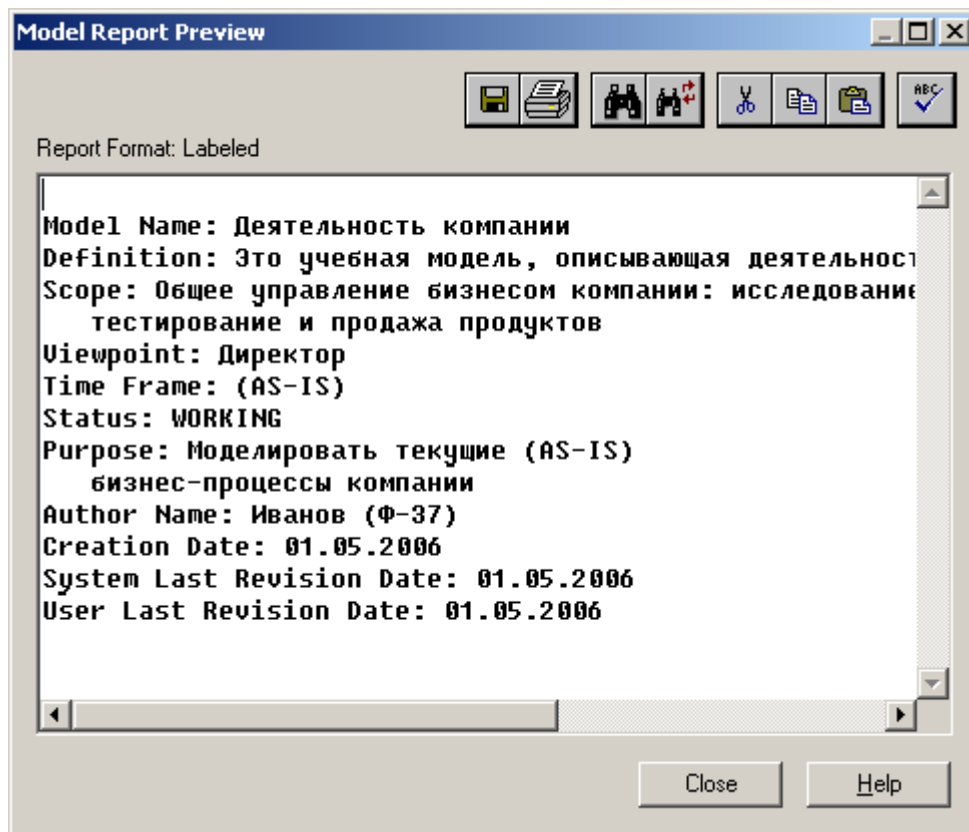


Рис. 18. Предварительный просмотр отчета Model Report

Тема 2. Декомпозиция производственных процессов по методологии IDEF0

Работы (Activity)

Работы обозначают поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Работы изображаются в виде прямоугольников. Все работы должны быть названы и определены. Имя работы должно быть выражено отглагольным существительным, обозначающим действие (например, "Изготовление детали", "Прием заказа" и т.д.). Работа "Изготовление детали" может иметь, например, следующее определение: "Работа относится к полному циклу изготовления изделия от контроля качества сырья до отгрузки готового упакованного изделия". При создании новой модели (меню **File/New**) автоматически создается контекстная диаграмма с единственной работой, изображающей систему в целом (рис. 1).

Для внесения имени работы следует щелкнуть по работе правой кнопкой мыши, выбрать в меню **Name Editor** и в появившемся диалоге внести имя работы. Для описания других свойств работы служит диалог **Activity Properties** (рис. 2).

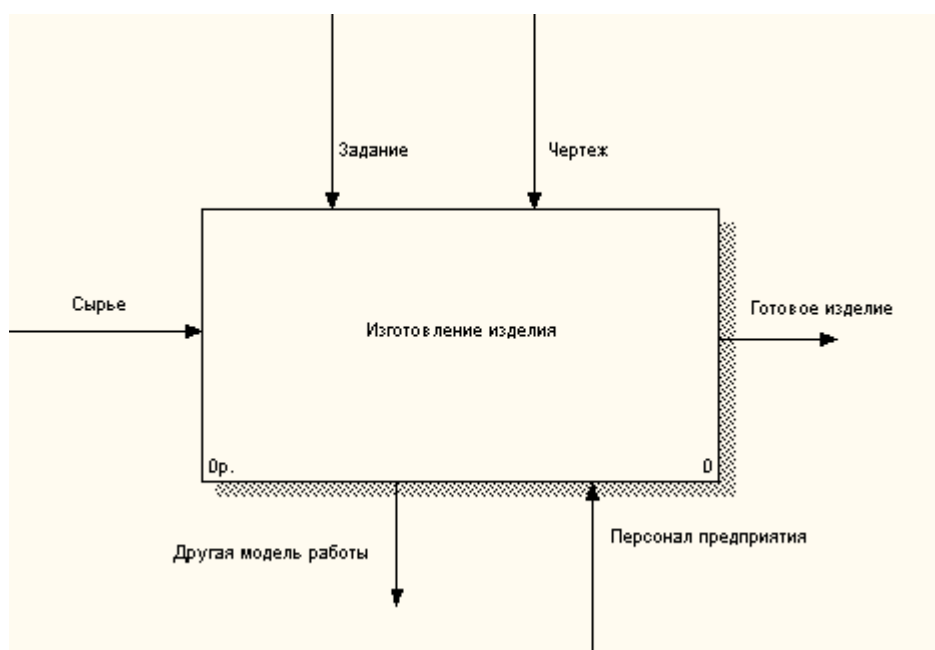


Рис. 1. Пример контекстной диаграммы

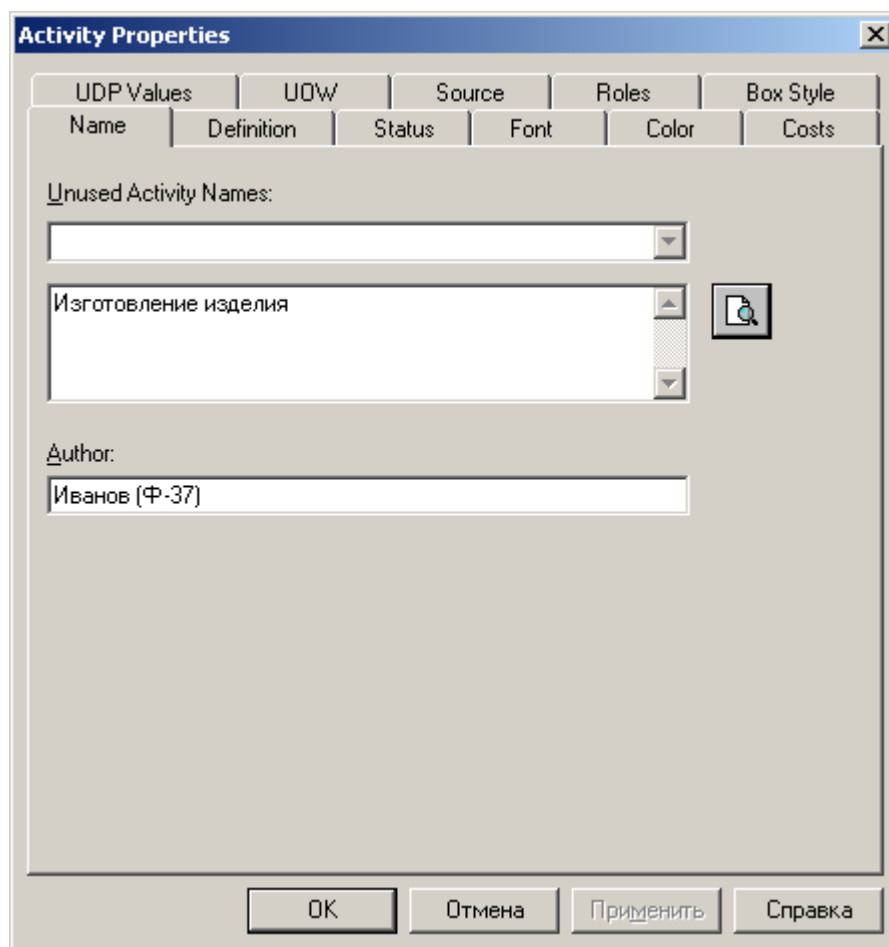


Рис. 2. Редактор задания свойств работы

Диаграммы декомпозиции содержат родственные работы, т.е. дочерние работы, имеющие общую родительскую работу. Для создания диаграммы декомпозиции следует щелкнуть по кнопке ▼.

Возникает диалог **Activity Box Count** (рис. 3), в котором следует указать нотацию новой диаграммы и количество работ на ней. Выберем нотацию **IDEF0** и щелкнем на **OK**. Появляется диаграмма декомпозиции (рис. 4). Допустимый интервал числа работ 2-8. Декомпозировать работу на одну работу не имеет смысла: диаграммы с количеством работ более восьми получаются перенасыщенными и плохо читаются. Для обеспечения наглядности и лучшего понимания моделируемых процессов рекомендуется использовать от трех до шести блоков на одной диаграмме.

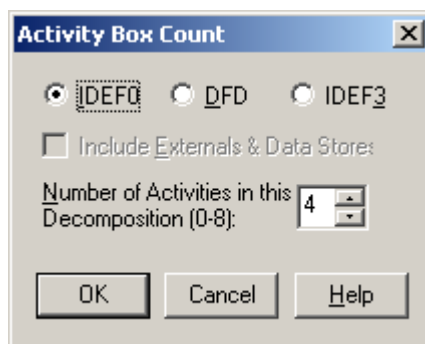


Рис. 3. Диалог Activity Box Count

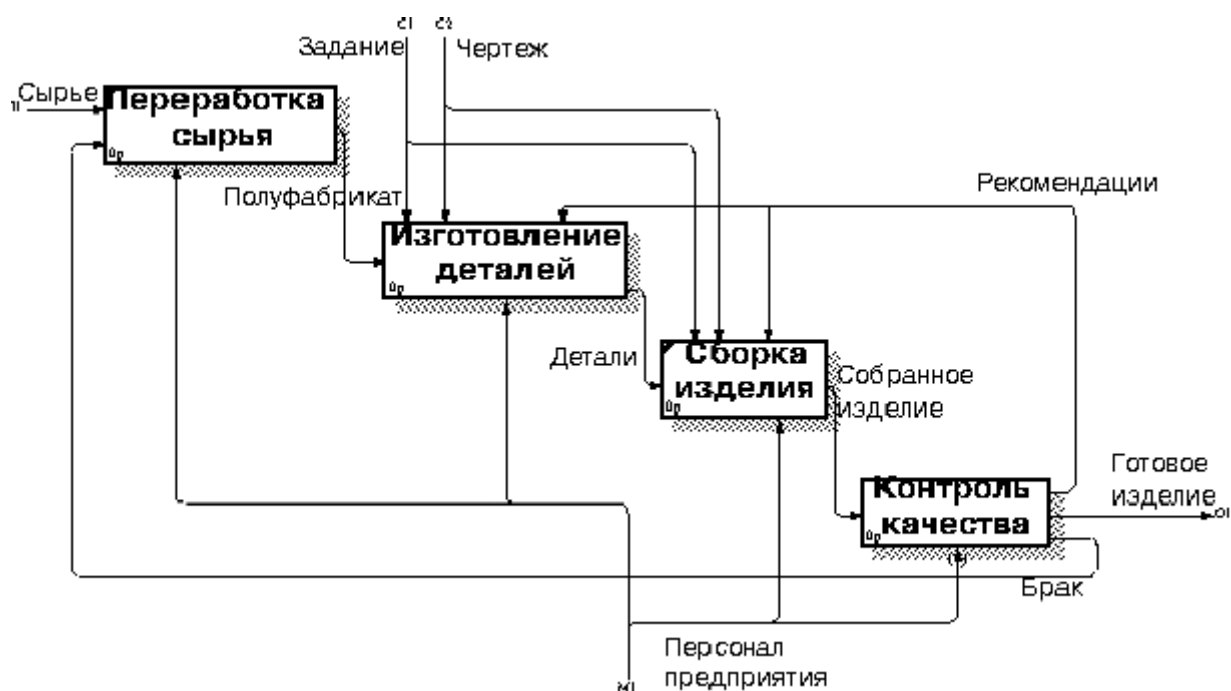



Рис. 4. Пример диаграммы декомпозиции

Если оказывается, что количество работ недостаточно, то работу можно добавить в диаграмму, щелкнув сначала по кнопке  на палитре инструментов, а затем по свободному месту на диаграмме.

Работы на диаграммах декомпозиции обычно располагаются по диагонали от левого верхнего угла к правому нижнему.

Такой порядок называется порядком доминирования. Согласно этому принципу расположения в левом верхнем углу располагается самая важная работа или работа, выполняемая по времени первой. Далее вправо вниз располагаются менее важные или выполняемые позже работы. Такое расположение облегчает

чтение диаграмм, кроме того, на нем основывается понятие взаимосвязей работ.

Каждая из работ на диаграмме декомпозиции может быть в свою очередь декомпозирована. На диаграмме декомпозиции работы нумеруются автоматически слева направо. Номер работы показывается в правом нижнем углу. В левом верхнем углу изображается небольшая диагональная черта, которая показывает, что данная работа не была декомпозирована. Так, например, работа "Сборка изделия" имеет номер 3 и не была еще декомпозирована. Работа "Контроль качества" (номер 4) имеет нижний уровень декомпозиции

Стрелки (Arrow)

Взаимодействие работ с внешним миром и между собой описывается в виде стрелок. Стрелки представляют собой некую информацию и именуется существительными (например, "Заготовка", "Изделие", "Заказ").

В IDEF0 различают пять типов стрелок:

Вход (Input) - материал или информация, которые используются или преобразуется работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Каждый тип стрелок подходит к определенной стороне прямоугольника, изображающего работу, или выходит из нее. Стрелка входа рисуется как входящая в левую грань работы. При описании технологических процессов (для этого и был придуман IDEF0) не возникает проблем определения входов. Действительно, "Сырье" на рис. 1. - это нечто, что перерабатывается в процессе "Изготовление изделия" для получения результата. При моделировании ИС, когда стрелками являются не физические объекты, а данные, не все так очевидно. Например, при "Приеме пациента" карта пациента может быть и на входе и на выходе, между тем качество этих данных меняется. Другими словами, в этом примере для того, чтобы оправдать свое назначение, стрелки входа и выхода должны быть точно определены с тем, чтобы указать на то, что данные действительно были переработаны (например, на

выходе - "Заполненная карта пациента"). Очень часто сложно определить, являются ли данные входом или управлением. В этом случае подсказкой может служить то, перерабатываются/изменяются ли данные в работе или нет. Если изменяются, то скорее всего это вход, если нет - управление.

Управление (Control) - правила, стратегии, процедуры или стандарты, которыми руководствуется работа. Каждая работа должна иметь хотя бы одну стрелку управления. Стрелка управления рисуется как входящая в верхнюю грань работы. На рис. 1 стрелки "Задание" и "Чертеж" - управление для работы "Изготовление изделия". Управление влияет на работу, но не преобразуется работой. Если цель работы - изменить процедуру или стратегию, то такая процедура или стратегия будет для работы входом. В случае возникновения неопределенности в статусе стрелки (управление или вход) рекомендуется рисовать стрелку управления.


Выход (Output) - материал или информация, которые производятся работой. Каждая работа должна иметь хотя бы одну стрелку выхода. Работа без результата не имеет смысла и не должна моделироваться. Стрелка выхода рисуется как исходящая из правой грани работы. На рис. 1 стрелка "Готовое изделие" является выходом для работы "Изготовление изделия".

Механизм (Mechanism) - ресурсы, которые выполняют работу, например персонал предприятия, станки, устройства и т. д. Стрелка механизма рисуется как входящая в нижнюю грань работы. На рис. 1 стрелка "Персонал предприятия" является механизмом для работы "Изготовление изделия". По усмотрению аналитика стрелки механизма могут не изображаться в модели.

Вызов (Call) - специальная стрелка, указывающая на другую модель работы. Стрелка вызова рисуется как исходящая из нижней грани работы. На рис. 1 стрелка "Другая модель работы" является вызовом для работы "Изготовление изделия". Стрелка вызова используется для указания того, что некоторая работа выполняется за пределами моделируемой системы. В VRwin стрелки вызова используются в механизме слияния и разделения моделей.

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

Для внесения граничной стрелки входа следует:

- щелкнуть по кнопке с символом стрелки → ;
- в палитре инструментов перенести курсор к левой стороне экрана, пока не появится начальная штриховая полоска;
- щелкнуть один раз по полоске (откуда выходит стрелка) и еще раз в левой части работы со стороны входа (где заканчивается стрелка);
- вернуться в палитру инструментов и выбрать опцию редактирования стрелки ;
- щелкнуть правой кнопкой мыши на линии стрелки, во всплывающем меню выбрать **Name Editor** и добавить имя стрелки в закладке **Name** диалога **Arrow Properties**(рис. 5).

Стрелки управления, выхода, механизма и выхода изображаются аналогично. Для рисования стрелки выхода, например, следует щелкнуть по кнопке с символом стрелки в палитре инструментов, щелкнуть в правой части работы со стороны выхода (где начинается стрелка), перенести курсор к правой стороне экрана, пока не появится начальная штриховая полоска, и щелкнуть один раз по штриховой полоске.

Имена вновь внесенных стрелок автоматически заносятся в словарь (**Arrow Dictionary**).

ICOM-коды. Диаграмма декомпозиции предназначена для детализации работы. В отличие от моделей, отображающих структуру организации, работа на диаграмме верхнего уровня в IDEF0 - это не элемент управления нижестоящими работами. Работы нижнего уровня - это то же самое, что работы верхнего уровня, но в более детальном изложении. Как следствие этого границы работы верхнего уровня - это то же самое, что границы диаграммы декомпозиции. **ICOM** (аббревиатура от **Input, Control, Output и Mechanism**) - коды, предназначенные для идентификации граничных стрелок. Код **ICOM** содержит

префикс, соответствующий типу стрелки (**I**, **C**, **O** или **M**), и порядковый номер. ВРwin вносит ICOM-коды автоматически. Для отображения ICOM-кодов следует включить опцию Show ICOM codes на закладке **Presentation** диалога **Model Properties**.

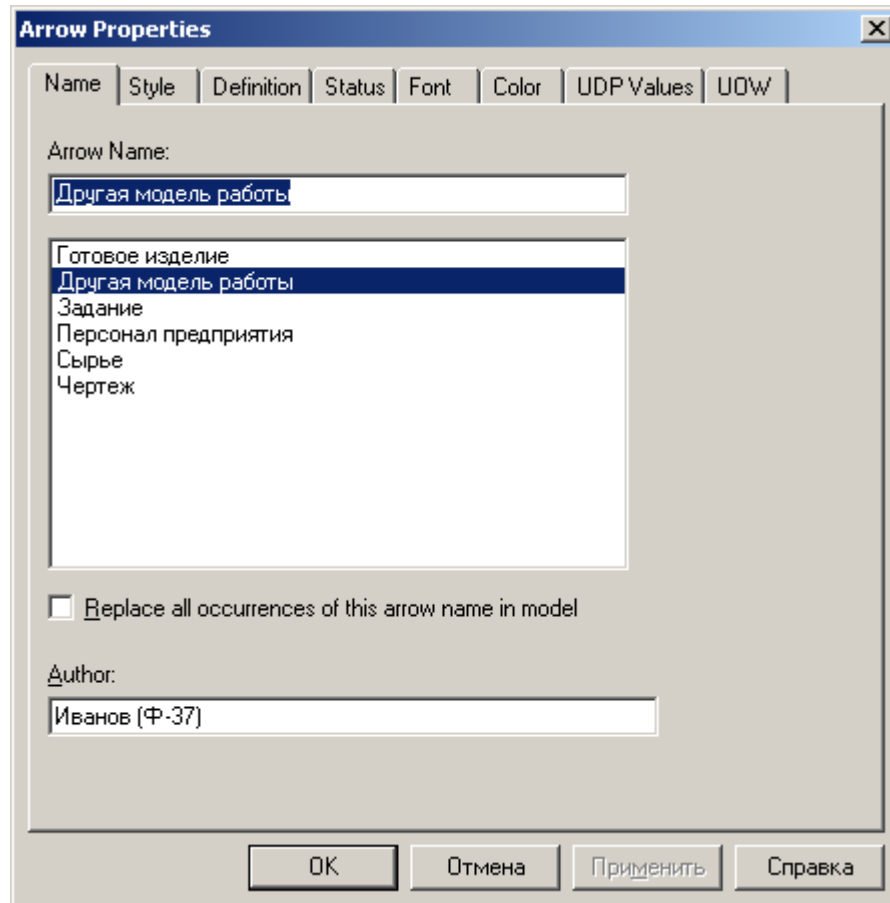


Рис. 5. Диалог Arrow Properties

Словарь стрелок редактируется при помощи специального редактора **Arrow Dictionary Editor**, в котором определяется стрелка и вносится относящийся к ней комментарий (рис. 6). Словарь стрелок решает очень важную задачу. Диаграммы создаются аналитиком для того, чтобы провести сеанс экспертизы, т. е. обсудить диаграмму со специалистом предметной области. В любой предметной области формируется профессиональный жаргон, причем очень часто жаргонные выражения имеют нечеткий смысл и воспринимаются разными специалистами по-разному. В то же время аналитик - автор диаграмм должен употреблять те выражения, которые наиболее понятны экспертам. Поскольку формальные определения часто

сложны для восприятия, аналитик вынужден употреблять профессиональный жаргон, а, чтобы не возникло неоднозначных трактовок, в словаре стрелок каждому понятию можно дать расширенное и, если это необходимо, формальное определение.

Содержимое словаря стрелок можно распечатать в виде отчета (меню **Report/Arrow Report...**) и получить тем самым толковый словарь терминов предметной области, использующихся в модели.

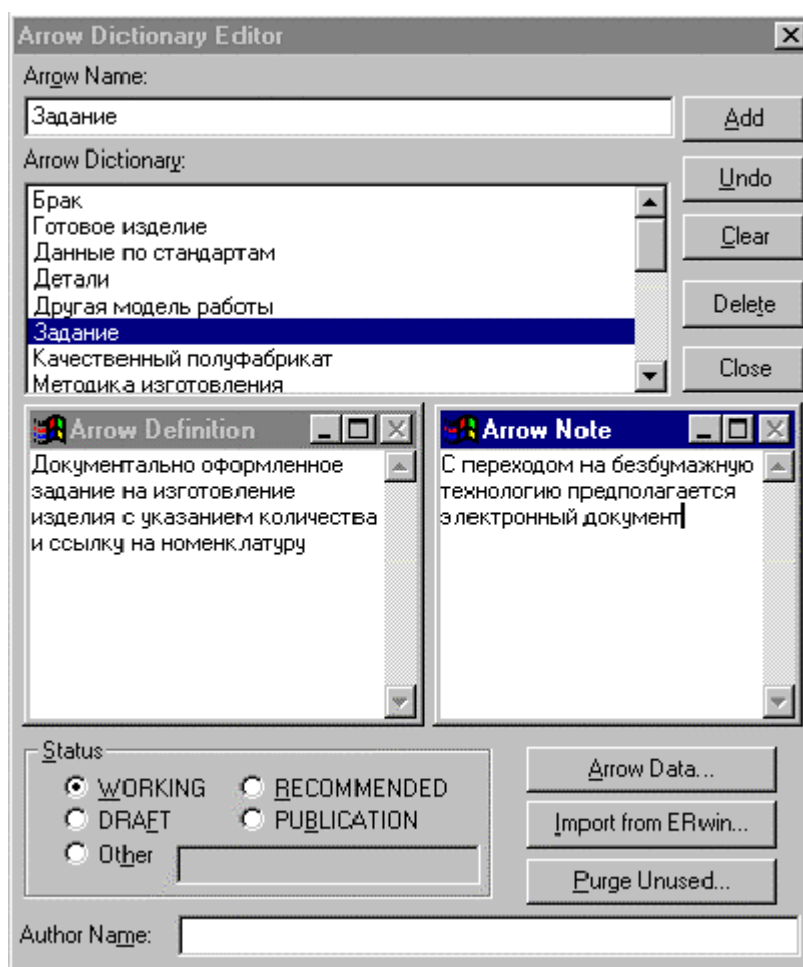


Рис. 6. Словарь стрелок

Несвязанные граничные стрелки (unconnected border arrow). При декомпозиции работы входящие в нее и исходящие из нее стрелки (кроме стрелки вызова) автоматически появляются на диаграмме декомпозиции (миграция стрелок), но при этом не касаются работ. Такие стрелки называются несвязанными и воспринимаются в VPwin как синтаксическая ошибка. Для связывания стрелок входа, управления или механизма

необходимо перейти в режим редактирования стрелок, щелкнуть по кончику стрелки и щелкнуть по соответствующему сегменту работы. Для связывания стрелки выхода необходимо перейти в режим редактирования стрелок, щелкнуть по сегменту выхода работы и затем по стрелке.

Внутренние стрелки. Для связи работ между собой используются внутренние стрелки, т. е. стрелки, которые не касаются границы диаграммы, начинаются у одной и кончаются у другой работы.

Для рисования внутренней стрелки необходимо в режиме рисования стрелок щелкнуть по сегменту (например, выхода) одной работы и затем по сегменту (например, входа) другой. В IDEF0 различают пять типов связей работ.

Связь по входу (output-input), когда стрелка выхода вышестоящей работы (далее - просто выход) направляется на вход нижестоящей.

Связь по управлению (output-control), когда выход вышестоящей работы направляется на управление нижестоящей. Связь по управлению показывает доминирование вышестоящей работы. Данные или объекты выхода вышестоящей работы не меняются в нижестоящей.

Обратная связь по входу (output-input feedback), когда выход нижестоящей работы направляется на вход вышестоящей. Такая связь, как правило, используется для описания циклов.

Обратная связь по управлению (output-control feedback), когда выход нижестоящей работы направляется на управление вышестоящей. Обратная связь по управлению часто свидетельствует об эффективности бизнес - процесса.

Связь выход-механизм (output-mechanism), когда выход одной работы направляется на механизм другой. Эта взаимосвязь используется реже остальных и показывает, что одна работа подготавливает ресурсы, необходимые для проведения другой работы.

Явные стрелки. Явная стрелка имеет источником одну-единственную работу и назначением тоже одну-единственную работу.

Разветвляющиеся и сливающиеся стрелки. Одни и те же данные или объекты, порожденные одной работой, могут использоваться сразу в нескольких других работах. С другой стороны, стрелки, порожденные в разных работах, могут представлять собой одинаковые или однородные данные или объекты, которые в дальнейшем используются или перерабатываются в одном месте. Для моделирования таких ситуаций в IDEF0 используются разветвляющиеся и сливающиеся стрелки. Для разветвления стрелки нужно в режиме редактирования стрелки щелкнуть по фрагменту стрелки и по соответствующему сегменту работы. Для слияния двух стрелок выхода нужно в режиме редактирования стрелки сначала щелкнуть по сегменту выхода работы, а затем по соответствующему фрагменту стрелки.

Смысл разветвляющихся и сливающихся стрелок передается именованим каждой ветви стрелок. Существуют определенные правила именования таких стрелок. Рассмотрим их на примере разветвляющихся стрелок. Если стрелка именована до разветвления, а после разветвления ни одна из ветвей не именована, то подразумевается, что каждая ветвь моделирует те же данные или объекты, что и ветвь до разветвления.


Если стрелка именована до разветвления, а после разветвления какая-либо из ветвей не именована, то подразумевается, что эти ветви соответствуют именованию. Если при этом какая-либо ветвь после разветвления осталась неименованной, то подразумевается, что она моделирует те же данные или объекты, что и ветвь до разветвления.

Недопустима ситуация, когда стрелка до разветвления не именована, а после разветвления не именована какая-либо из ветвей. VPwin определяет такую стрелку как синтаксическую ошибку.

Правила именования сливающихся стрелок полностью аналогичны - ошибкой будет считаться стрелка, которая после слияния не именована, а до слияния не именована какая-либо из ее ветвей. Для именования отдельной ветви разветвляющихся и сливающихся стрелок следует выделить на диаграмме только одну ветвь, после этого вызвать редактор имени и присвоить имя

стрелке. Это имя будет соответствовать только выделенной ветви.

Тоннелирование стрелок. Вновь внесенные граничные стрелки на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня.

Для их "перетаскивания" наверх нужно сначала выбрать кнопку  на палитре инструментов и щелкнуть по квадратным скобкам граничной стрелки. Появляется диалог **Border Arrow Editor** (рис. 7).

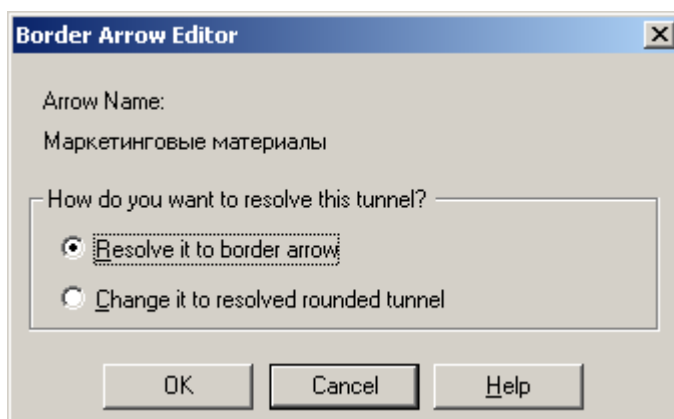


Рис. 7. Диалог Border Arrow Editor

Если щелкнуть по кнопке **Resolve Border Arrow**, стрелка мигрирует на диаграмму верхнего уровня, если по кнопке **Change To Tunnel** - стрелка будет затоннелирована и не попадет на другую диаграмму.

Тоннелирование может быть применено для изображения малозначимых стрелок. Если на какой-либо диаграмме нижнего уровня необходимо изобразить малозначимые данные или объекты, которые не обрабатываются или не используются работами на текущем уровне, то их необходимо направить на вышестоящий уровень (на родительскую диаграмму). Если эти данные не используются на родительской диаграмме, их нужно направить еще выше, и т. д. В результате малозначимая стрелка будет изображена на всех уровнях и затруднит чтение всех диаграмм, на которых она присутствует. Выходом является тоннелирование стрелки на самом нижнем уровне. Такое тоннелирование называется "не-в-родительской-диаграмме".


Лабораторная работа №2

Цель работы: Научиться проводить декомпозицию производственных процессов с использованием методологии IDEF0 в среде VPwin 4.0.

Постановка задачи: Построить диаграмму декомпозиции для производственного процесса согласно выбранному варианту задания. Варианты производственных процессов приведены в задании к лабораторной работе № 1.

Результат: Отчет по лабораторной работе с кратким описанием производственного процесса; описанием процедуры создания диаграммы декомпозиции; распечаткой диаграммы и отчетом по модели.

Пример выполнения работы

1. Выберите кнопку  перехода на нижний уровень в палитре инструментов и в диалоговом окне **Activity Box Count** (рис. 8) установите число работ на диаграмме нижнего уровня - 3 - и нажмите кнопку **ОК**.

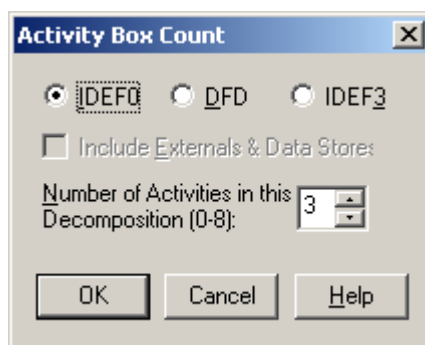


Рис. 8. Диалоговое окно Activity Box Count

2. Автоматически будет создана диаграмма декомпозиции (рис. 9).

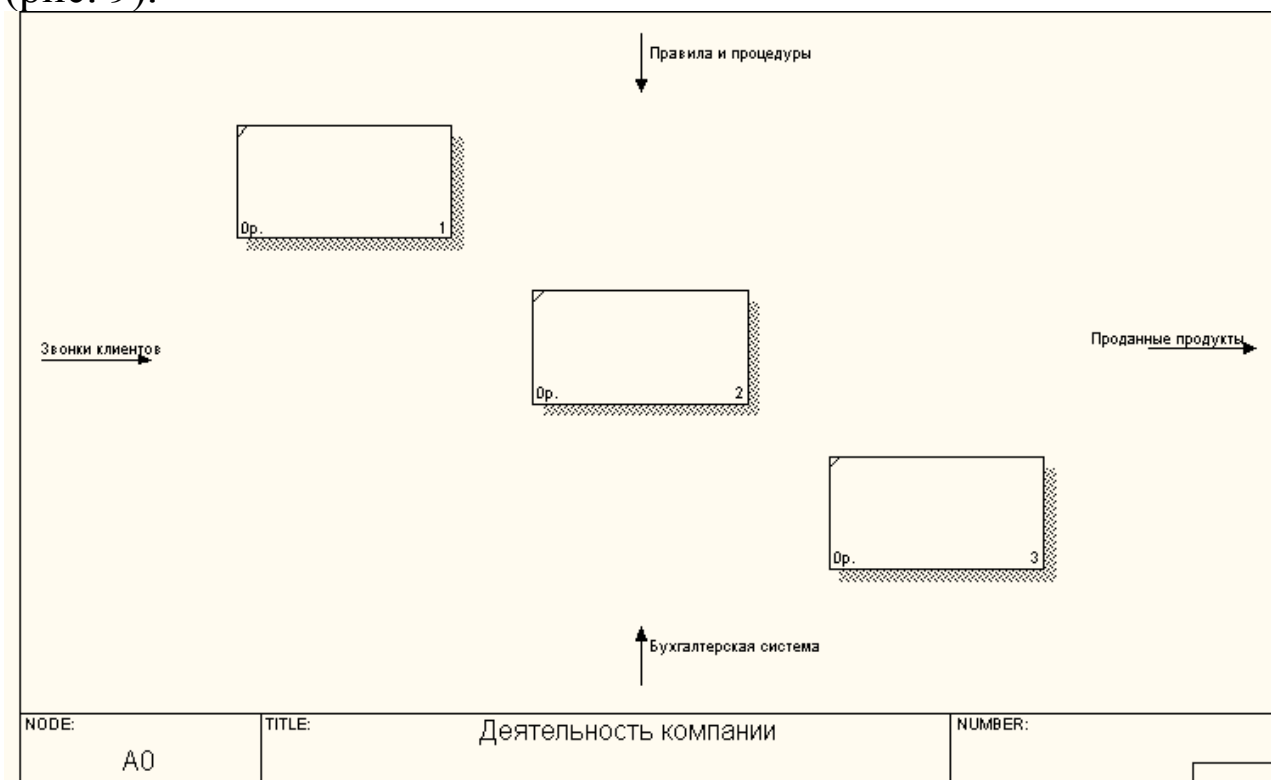


Рис. 9. Диаграмма декомпозиции

Правой кнопкой мыши щелкните по работе расположенной в левом верхнем углу области редактирования модели, выберите в контекстном меню опцию **Name** и внесите имя работы. Повторите операцию для оставшихся двух работ. Затем внесите определение, статус и источник для каждой работы согласно данным таблицы 1.

Таблица 1. Работы диаграммы декомпозиции A0

Название работы (Activity Name)	Определение работы (Activity Definition)
Продажи и маркетинг	Телемаркетинг и презентации, выставки
Сборка и тестирование компьютеров	Сборка и тестирование настольных и портативных компьютеров
Отгрузка и получение	Отгрузка заказов клиентам и получение компонентов от поставщиков

Диаграмма декомпозиции примет вид представленный на рис. 10.

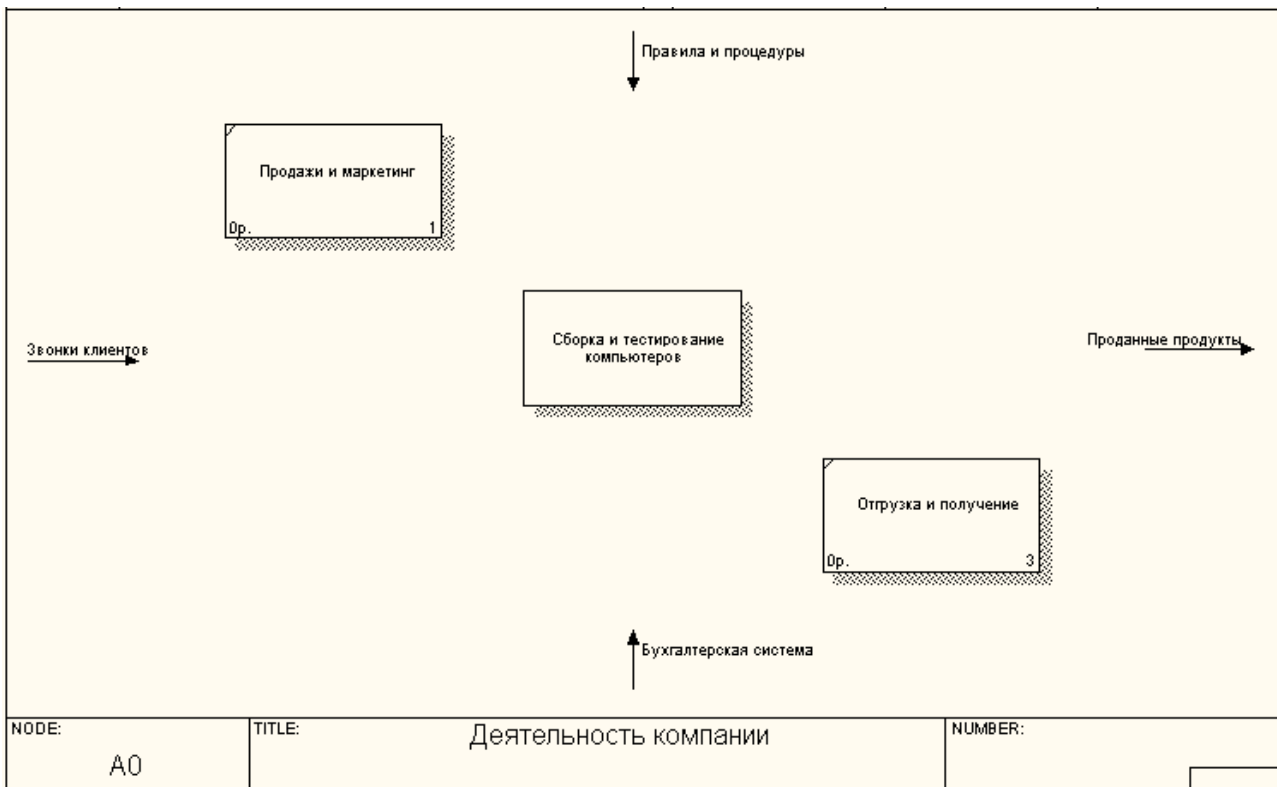



Рис.10 Диаграмма декомпозиции после присвоения работ наименований


3. Для изменения свойств работ после их внесения в диаграмму можно воспользоваться словарем работ (рис. 11). Вызов словаря производится при помощи пункта главного меню **Dictionary /Activity**.

Name	Definition	Author
Деятельность компан	Текущие бизнес-процессы компании	Иванов (Ф-37)
Отгрузка и полчение	Отгрузка заказов клиентам и получение комп	Иванов (Ф-37)
Продажи и маркетинг	Телемаркетинг и презентации, выставки	Иванов (Ф-37)
Сборка и тестировани	Сборка и тестирование настольных и портати	Иванов (Ф-37)

Рис. 11. Словарь Activity Dictionary

Если описать имя и свойства работы в словаре, ее можно будет внести в диаграмму позже с помощью кнопки в палитре инструментов. Невозможно удалить работу из словаря, если она используется на какой-либо диаграмме. Если работа удаляется из

диаграммы, из словаря она не удаляется. Имя и описание такой работы может быть использовано в дальнейшем. Для добавления работы в словарь необходимо перейти в конец списка и щелкнуть правой кнопкой по последней строке. Возникает новая строка, в которой нужно внести имя и свойства работы. Для удаления всех имен работ, не используемых в модели, щелкните по кнопке  (**Purge** (Чистить)).

4. Перейдите в режим рисования стрелок и свяжите граничные стрелки, воспользовавшись кнопкой  на палитре инструментов так, как это показано на рис. 12.

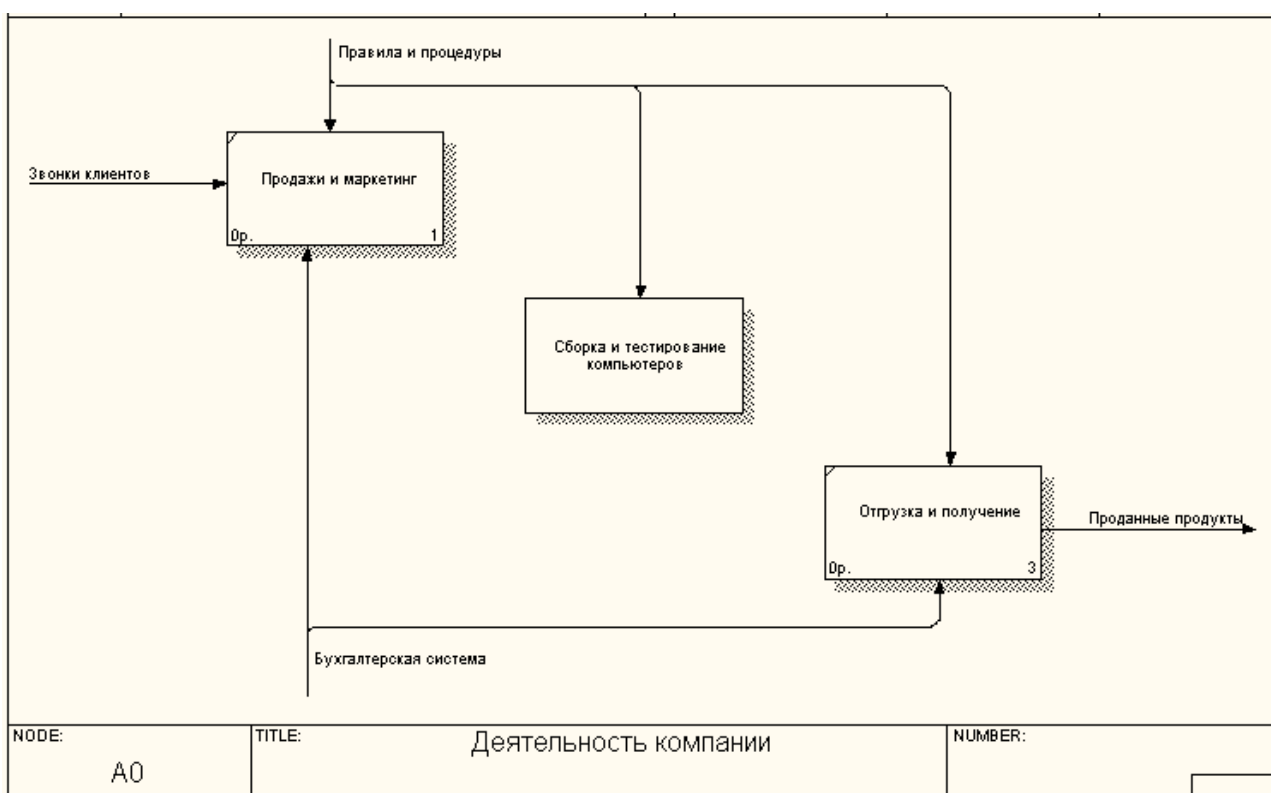


Рис. 12. Связанные граничные стрелки на диаграмме А0

5. Правой кнопкой мыши щелкните по ветви стрелки управления работы "Сборка и тестирование компьютеров" и переименуйте ее в "Правила сборки и тестирования" (рис. 13). Внесите определение для новой ветви: "Инструкции по сборке, процедуры тестирования, критерии производительности и т. д.". Правой кнопкой мыши щелкните по ветви стрелки механизма работы "Продажи и маркетинг" и переименуйте ее как "Система оформления заказов" (рис. 14).

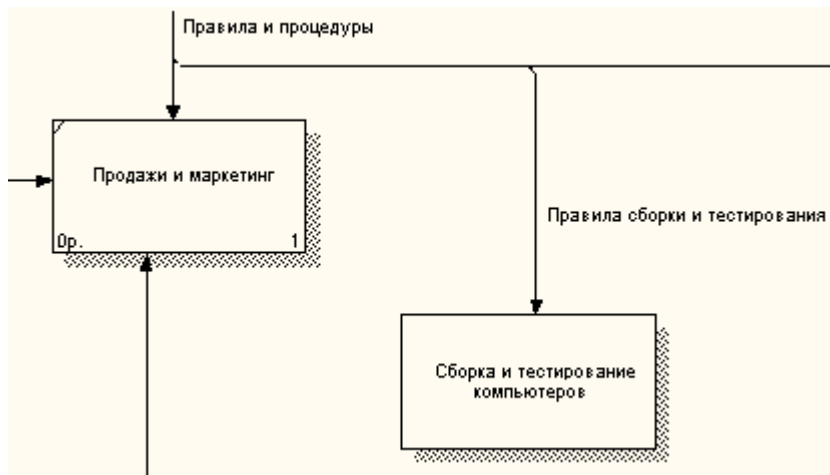


Рис. 13. Стрелка "Правила сборки и тестирования"

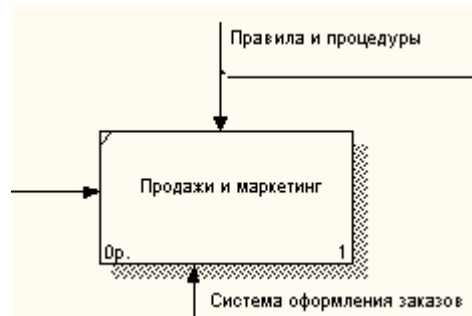


Рис. 14. Стрелка "Система оформления заказов"

6. Альтернативный метод внесения имен и свойств стрелок - использование словаря стрелок (вызов словаря - меню **Dictionary/ Arrow**). Если внести имя и свойства стрелки в словарь (рис. 15), ее можно будет внести в диаграмму позже.

Name	Definition	Author	Status
Бухгалтерская система	Оформление счетов, оплата счетов, р	Иванов (Ф-37)	WORKING
Звонки клиентов	Запросы информации, заказы, техно	Иванов (Ф-37)	WORKING
Правила и процедуры	Правила продаж, инструкции по сбор	Иванов (Ф-37)	WORKING
Правила сборки и тестиров	Инструкции по сборке, процедуры те	Иванов (Ф-37)	WORKING
Проданные продукты	Настольные и портативные компьют	Иванов (Ф-37)	WORKING
Система оформления зака		Иванов (Ф-37)	WORKING
			WORKING

Рис. 15. Словарь стрелок

Стрелку нельзя удалить из словаря, если она используется на какой-либо диаграмме. Если удалить стрелку из диаграммы, из словаря она не удаляется. Имя и описание такой стрелки может быть использовано в дальнейшем. Для добавления стрелки необходимо перейти в конец списка и щелкнуть правой кнопкой по последней строке. Возникает новая строка, в которой нужно внести имя и свойства стрелки.

7. Создайте новые внутренние стрелки так, как показано на рис. 16.

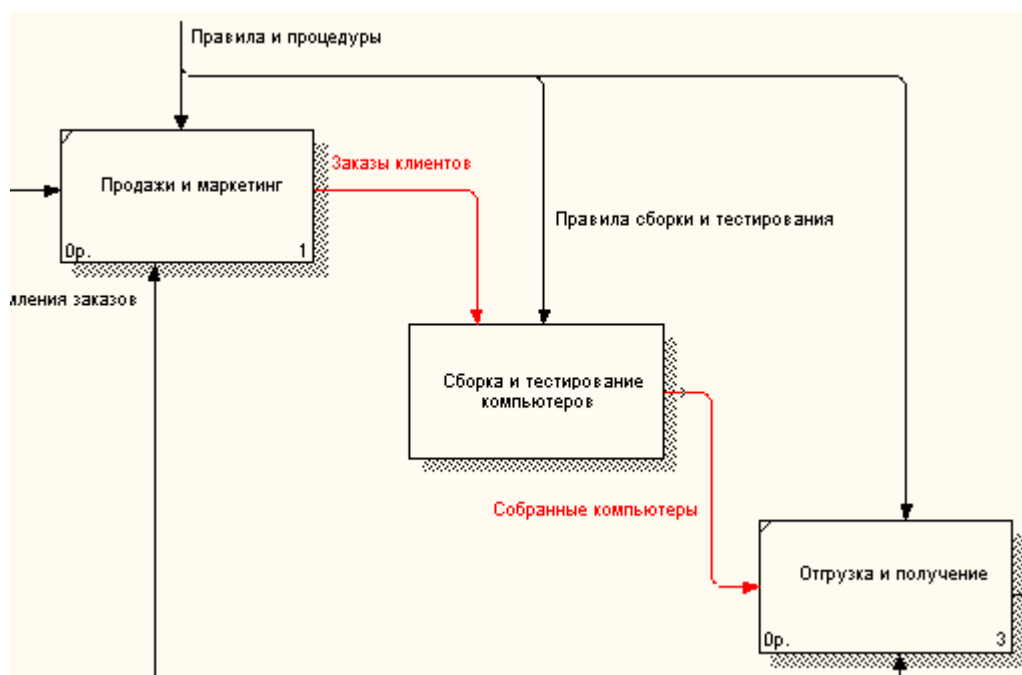


Рис. 16. Внутренние стрелки диаграммы А0

8. Создайте стрелку обратной связи (по управлению) "Результаты сборки и тестирования", идущую от работы "Сборка и тестирование компьютеров" к работе "Продажи и маркетинг". Измените, при необходимости, стиль стрелки (толщина линий) и установите опцию **Extra Arrowhead** (Дополнительный Наконечник стрелы) (из контекстного меню). Методом **drag&drop** перенесите имена стрелок так, чтобы их было удобнее читать. Если необходимо, установите из контекстного меню **Squiggle** (Загогулину). Результат возможных изменений показан на рис. 17.

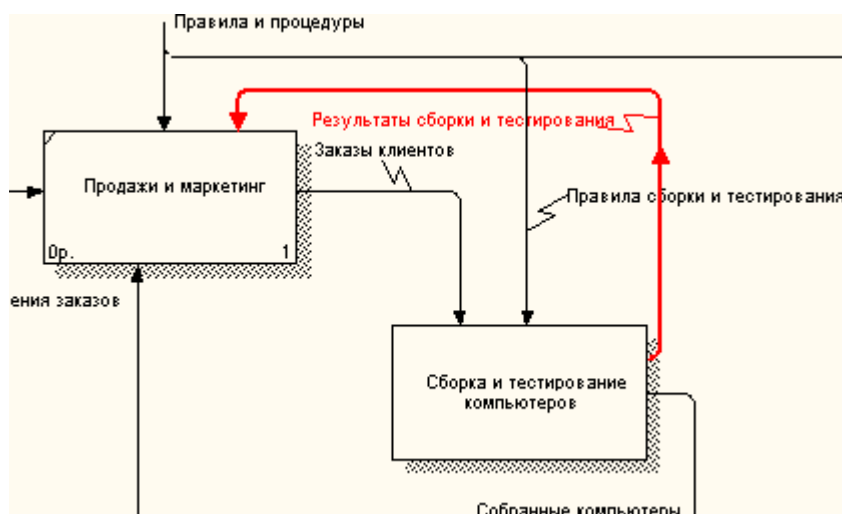



Рис. 17. Результат редактирования стрелок на диаграмме А0

9. Создайте новую граничную стрелку выхода "Маркетинговые материалы", выходящую из работы "Продажи и маркетинг". Эта стрелка автоматически не попадает на диаграмму верхнего уровня и имеет квадратные скобки на наконечнике  (рис. 18).

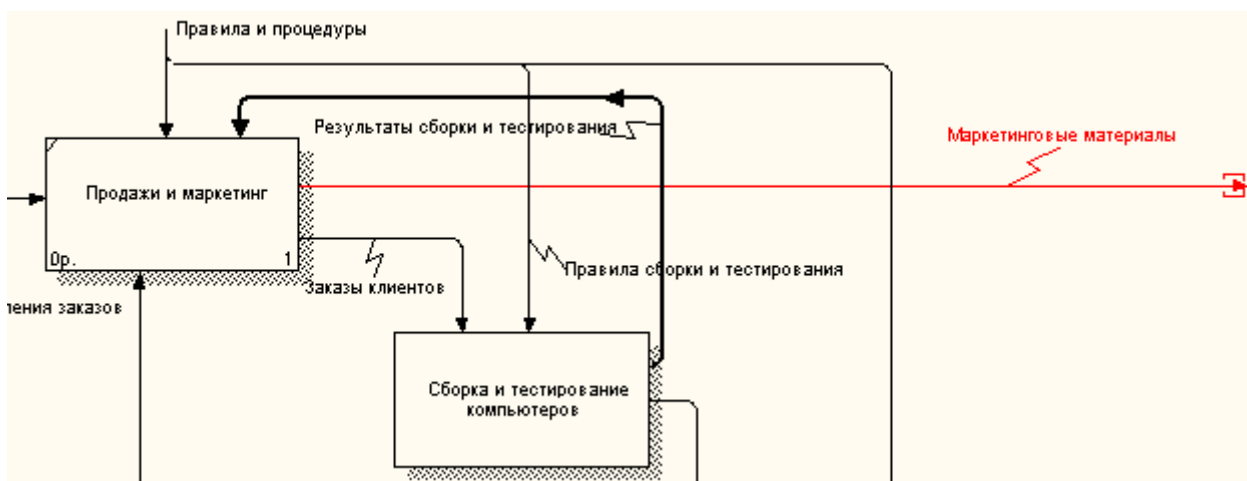


Рис. 18. Стрелка Маркетинговые материалы

10. Щелкните правой кнопкой мыши по квадратным скобкам и выберите пункт меню **Arrow Tunnel** (рис. 19).

В диалоговом окне **Border Arrow Editor** (Редактор Граничных Стрелок) выберите опцию **Resolve it to Border Arrow** (Разрешить как Граничную Стрелку) (рис. 20).

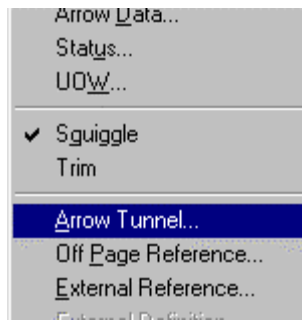


Рис. 19. Пункт меню Arrow Tunnel

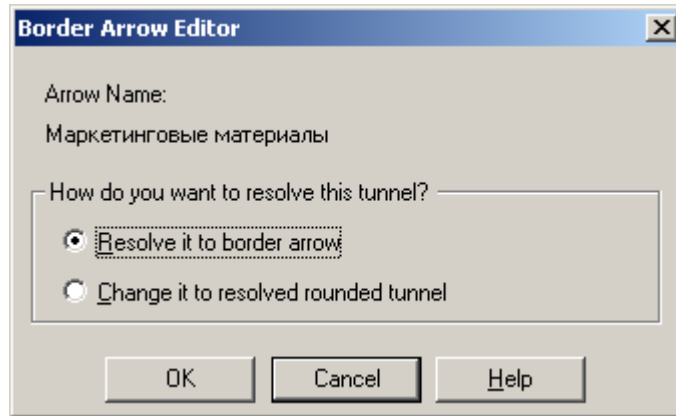


Рис. 20. Диалоговое окно Border Arrow Editor

Для стрелки "Маркетинговые материалы" выберите опцию **Trim** (Упорядочить) из контекстного меню. Результат выполнения лабораторной работы показан на рис. 21.

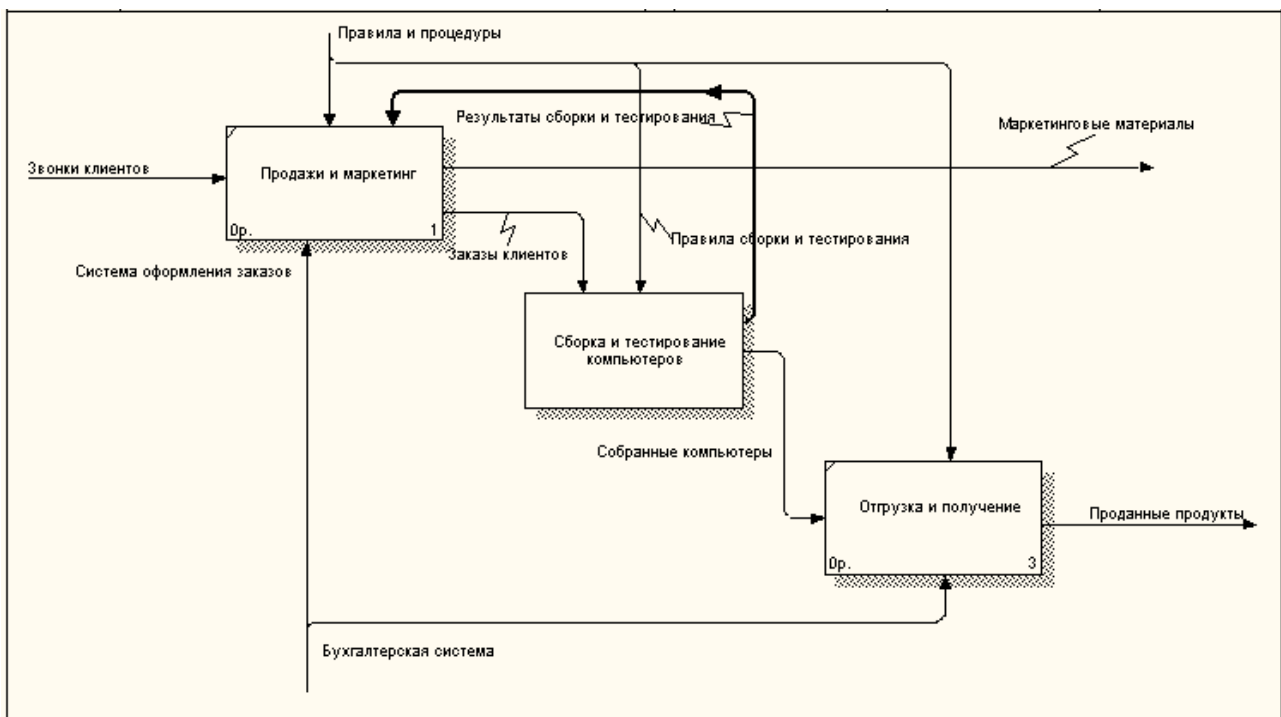


Рис. 21. Результат выполнения декомпозиции

Тема 3. Диаграммы дерева узлов и диаграммы FEO

Диаграмма дерева узлов показывает иерархию работ в модели и позволяет рассмотреть всю модель целиком, но не показывает взаимосвязи между работами (стрелки) (рис. 1). Процесс создания модели работ является итерационным, следовательно, работы могут менять свое расположение в дереве узлов многократно. Чтобы не запутаться и проверить способ декомпозиции, следует после каждого изменения создавать диаграмму дерева узлов. Впрочем, VPwin имеет мощный инструмент навигации по модели -Model Explorer, который позволяет представить иерархию работ и диаграмм в удобном и компактном виде, однако этот инструмент не является составляющей стандарта IDEF0.



Рис. 1. Диаграмма дерева узлов

Для создания диаграммы дерева узлов следует выбрать в меню пункт **Diagram/Add Node Tree**.

В мастере **Node Tree Wizard** следует указать глубину дерева - **Number of Levels** (по умолчанию 3) и корень дерева (по умолчанию - родительская работа текущей диаграммы). По умолчанию нижний уровень декомпозиции показывается в виде списка, остальные работы - в виде прямоугольников. Для отображения всего дерева в виде прямоугольников следует выключить опцию **Bullet Last Level**. При создании дерева узлов следует указать имя диаграммы, поскольку, если в нескольких

диаграммах в качестве корня на дереве узлов использовать одну и ту же работу, все эти диаграммы получают одинаковый номер (номер узла + постфикс N, например AON) и в списке открытых диаграмм (пункт меню **Window**) их можно будет различить только по имени.

Диаграммы "только для экспозиции" (FEO) часто используются в модели для иллюстрации других точек зрения, для отображения отдельных деталей, которые не поддерживаются явно синтаксисом IDEF0. Диаграммы FEO позволяют нарушить любое синтаксическое правило, поскольку по сути являются просто картинками - копиями стандартных диаграмм и не включаются в анализ синтаксиса. Например, работа на диаграмме FEO может не иметь стрелок управления и выхода. С целью обсуждения определенных аспектов модели с экспертом предметной области может быть создана диаграмма только с одной работой и одной стрелкой, поскольку стандартная диаграмма декомпозиции содержит множество деталей, не относящихся к теме обсуждения и дезориентирующих эксперта. Но если FEO используется для иллюстрации альтернативных точек зрения (альтернативный контекст), рекомендуется все-таки придерживаться синтаксиса IDEF0. Для создания диаграммы FEO следует выбрать пункт меню Insert/FEO Diagram. В возникающем диалоге Create New FEO Diagram следует указать имя диаграммы FEO и тип родительской диаграммы (рис. 2).

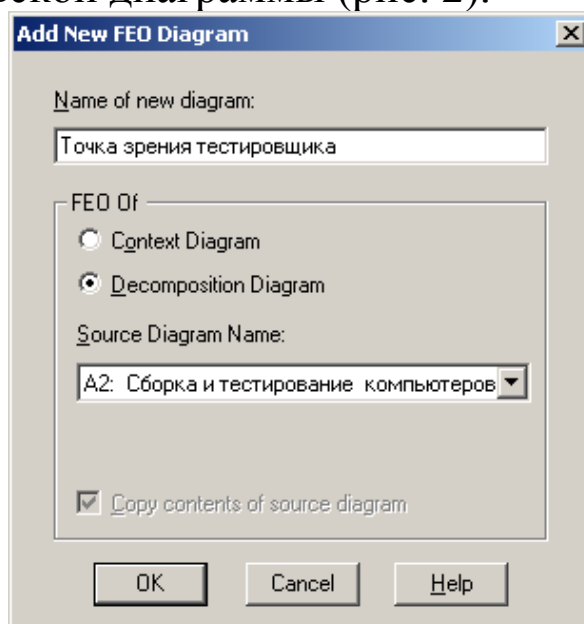


Рис. 2. Диалог создания FEO диаграммы

Лабораторная работа №3

Цель работы: Научиться строить диаграммы дерева узлов и диаграммы FEO (альтернативных точек зрения) для производственных процессов с использованием методологии IDEF0 в среде PRwin 4.0

Постановка задачи: Построить диаграмму декомпозиции второго уровня (A2) для одной из трех активностей первого уровня. Построить диаграмму дерева узлов и диаграмму FEO для производственного процесса согласно выбранному варианту задания. Варианты производственных процессов приведены в задании к лабораторной работе № 1.

Результат: Отчет по лабораторной работе с кратким описанием производственного процесса; описанием процедуры создания диаграмм; распечаткой диаграмм и отчетом по модели.

Пример выполнения работы

Декомпозируем работу "Сборка и тестирование компьютеров".

В результате проведения экспертизы получена следующая информация:

- Производственный отдел получает заказы клиентов от отдела продаж по мере их поступления.
- Диспетчер координирует работу сборщиков, сортирует заказы, группирует их и дает указание на отгрузку компьютеров, когда они готовы.
- Каждые 2 часа диспетчер группирует заказы - отдельно для настольных компьютеров и ноутбуков - и направляет на участок сборки.

- Сотрудники участка сборки собирают компьютеры согласно спецификациям заказа и инструкциям по сборке. Когда группа компьютеров, соответствующая группе заказов, собрана, она направляется на тестирование. Тестировщики тестируют каждый компьютер и в случае необходимости заменяют неисправные компоненты.
- Тестировщики направляют результаты тестирования диспетчеру, который на основании этой информации принимает решение о передаче компьютеров, соответствующих группе заказов, на отгрузку.

Методика выполнения упражнения

1. На основе информации, полученной в результате экспертизы, внесите новые работы и стрелки (таблица 1 и 2).

Таблица 1. Работы диаграммы декомпозиции A2

Название работы (Activity Name)	Определение работы (Activity Definition)
Отслеживание расписания и управление сборкой и тестированием	Просмотр заказов, установка расписания выполнения заказов, просмотр результатов тестирования, формирование групп заказов на сборку и отгрузку
Сборка настольных компьютеров	Сборка настольных компьютеров в соответствии с инструкциями и указаниями диспетчера
Сборка ноутбуков	Сборка ноутбуков в соответствии с инструкциями и указаниями диспетчера
Тестирование компьютеров	Тестирование компьютеров и компонентов. Замена неработающих компонентов

Таблица 2. Стрелки диаграммы декомпозиции A2

Наименование стрелки (Arrow Name)	Источник стрелки (Arrow Source)	Тип стрелки источника (Arrow Src.Type)	Приемник стрелки (Arrow Dest.)	Тип стрелки приемника (Arrow Dest. Type)
Диспетчер	Персонал производственного отдела		Отслеживание расписания и управление сборкой и тестированием	Mechanism
Заказы клиентов	Граница диаграммы	Control	Отслеживание расписания и управление сборкой и тестированием	Control

Продолжение таблицы 2. Стрелки диаграммы декомпозиции A2

Наименование стрелки (Arrow Name)	Источник стрелки (Arrow Source)	Тип стрелки источника (Arrow Src. Type)	Приемник стрелки (Arrow Dest.)	Тип стрелки приемника (Arrow Dest. Type)
Заказы на настольные компьютеры	Отслеживание расписания и управление сборкой и тестированием	Output	Сборка настольных компьютеров	Control
Заказы на ноутбуки	Отслеживание расписания и управление сборкой и тестированием	Output	Сборка ноутбуков	Control
Компоненты	"Tunnel"	Input	Сборка настольных компьютеров	Input
			Сборка ноутбуков	Input
			Тестирование компьютеров	Input
Настольные компьютеры	Сборка настольных компьютеров	Output	Тестирование компьютеров	Input
Ноутбуки	Сборка ноутбуков	Output	Тестирование компьютеров	Input
Персонал производственного отдела	"Tunnel"		Сборка настольных компьютеров	Mechanism
			Сборка ноутбуков	Mechanism
Правила сборки и тестирования	Граница диаграммы		Сборка настольных компьютеров	Control
			Сборка ноутбуков	Control
			Тестирование компьютеров	Control
Результаты сборки и тестирования	Сборка настольных компьютеров	Output	Граница диаграммы	Output
	Сборка ноутбуков	Output		
	Тестирование компьютеров	Output		
Результаты тестирования	Тестирование компьютеров	Output	Отслеживание расписания и управление сборкой и тестированием	Input
Собранные компьютеры	Тестирование компьютеров	Output	Граница диаграммы	Output
Тестировщик	Персонал производственного отдела		Тестирование компьютеров	Mechanism
Указание передать компьютеры на отгрузку	Отслеживание расписания и управление сборкой и тестированием	Output	Тестирование компьютеров	Control

2. Туннелируйте и свяжите на верхнем уровне граничные стрелки, если это необходимо. Результат выполнения декомпозиции A2 показан на рис. 3.

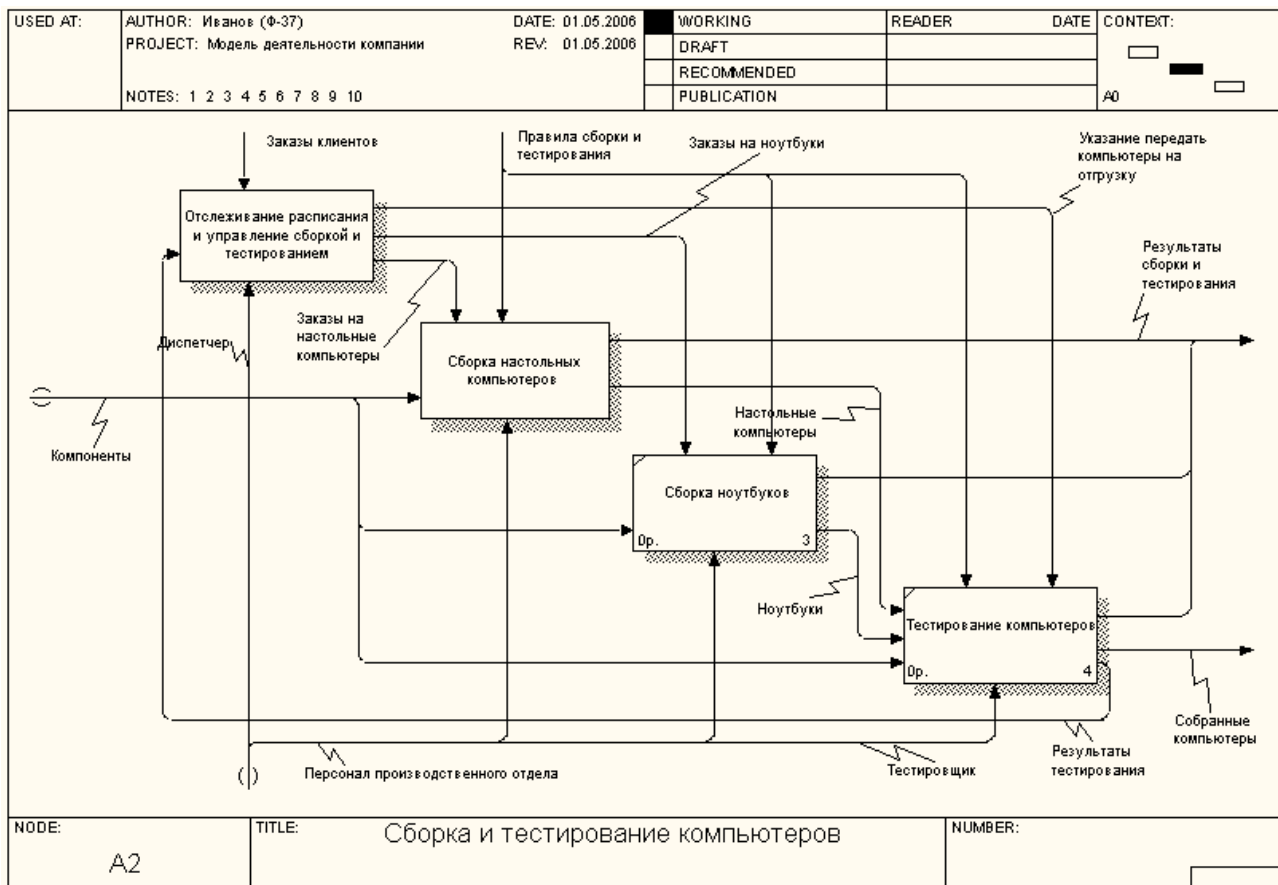


Рис. 3. Результат выполнения декомпозиции A2

3. Выберите пункт главного меню **Diagram/Add Node Tree**.

4. В первом диалоговом окне мастера **Node Tree Wizard** внесите имя диаграммы, укажите диаграмму корня дерева и количество уровней (рис. 4).

5. Во втором диалоговом окне мастера **Node Tree Wizard** установите опции, как показано на рис. 5.

6. Щелкните по кнопке **Finish**. В результате будет создана диаграмма дерева узлов (**Node tree Diagram**) (рис. 6).

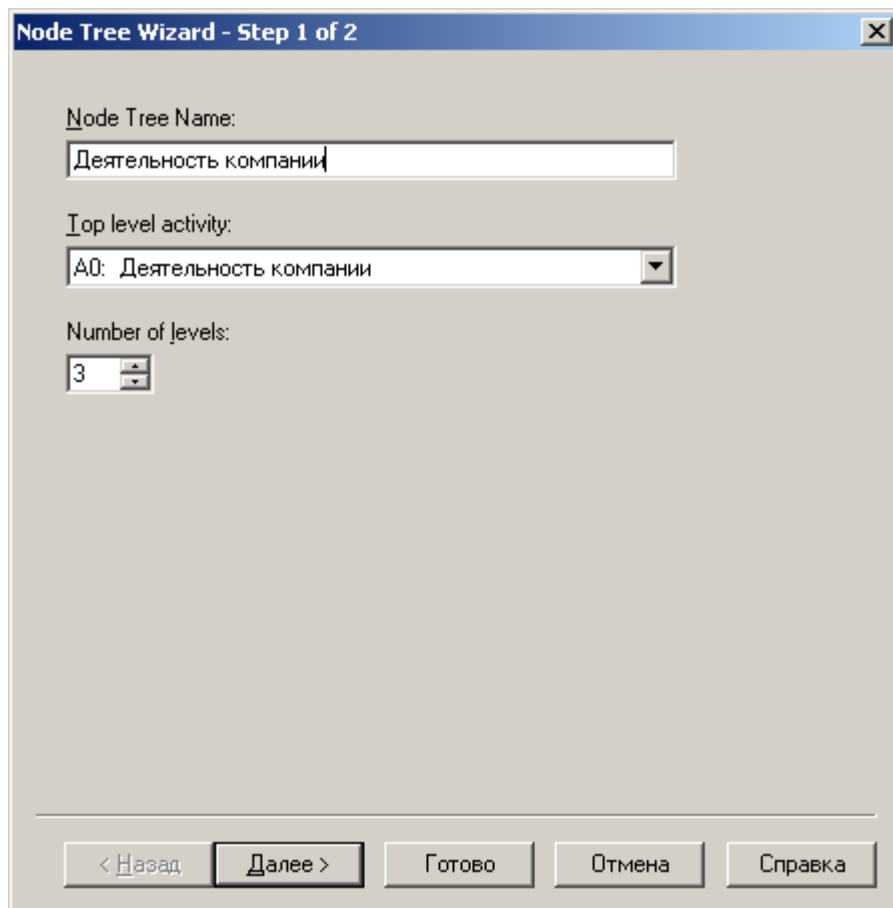


Рис. 4. Первое диалоговое окно мастера Node Tree Wizard

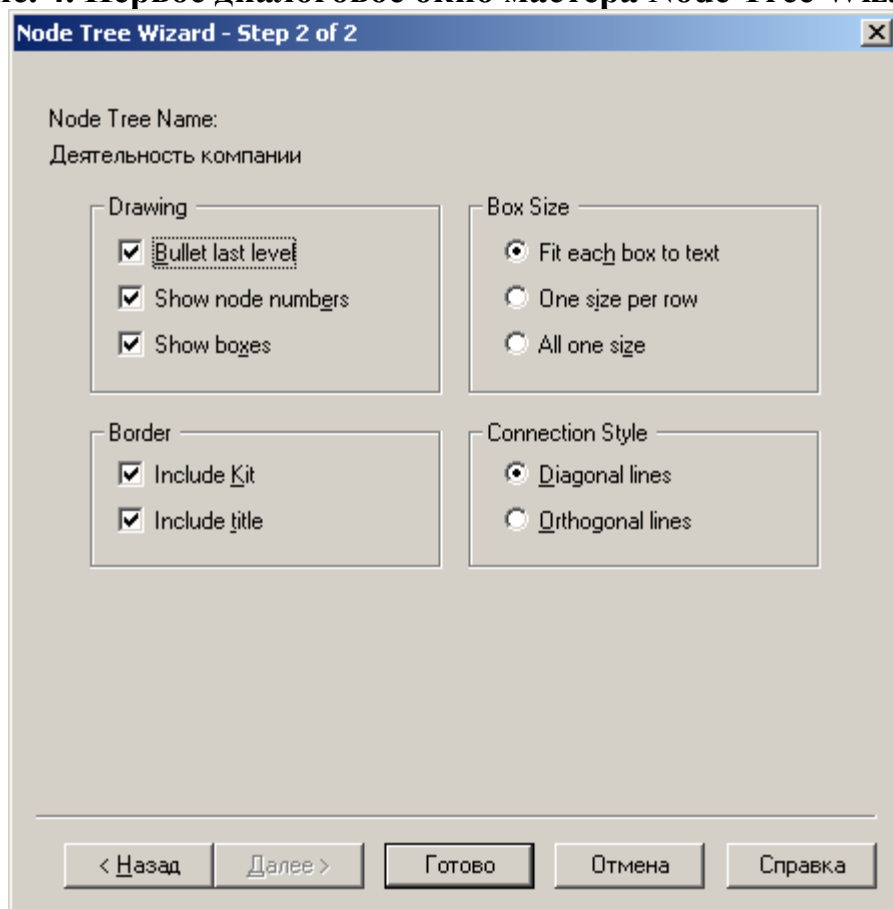


Рис. 5. Второе диалоговое окно мастера Node Tree Wizard

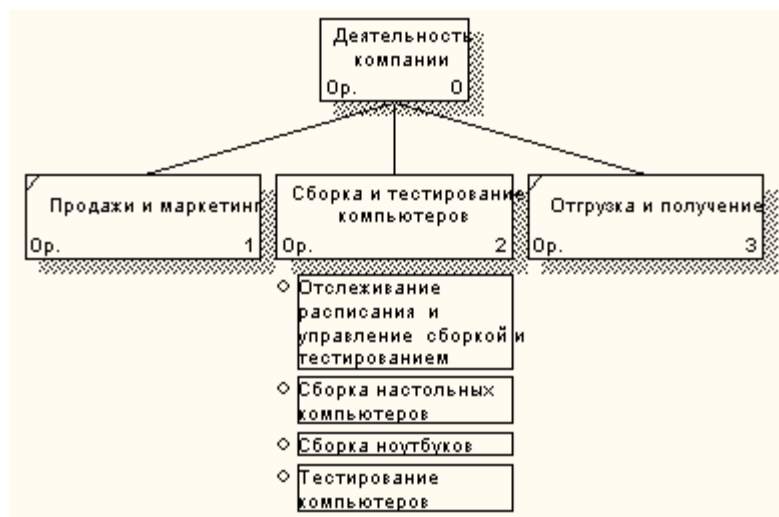


Рис. 6. Диаграмма дерева узлов

7. Диаграмму дерева узлов можно модифицировать. Нижний уровень может быть отображен не в виде списка, а в виде прямоугольников, так же как и верхние уровни. Для модификации диаграммы правой кнопкой мыши щелкните по свободному месту, не занятому объектами, выберите меню **Node tree Diagram Properties** и во вкладке **Style** диалога **Node Tree Properties** отключите опцию **Bullet Last Level** (рис. 7).

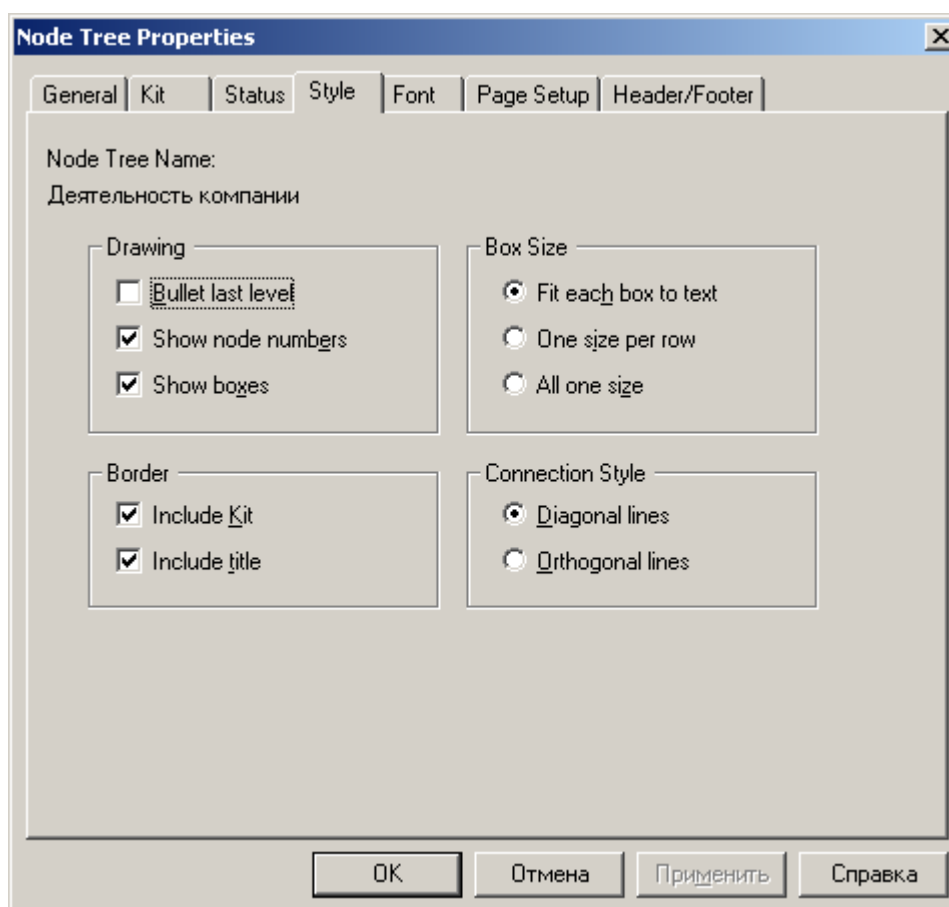


Рис. 7. Отключение опции Bullet Last Level

8. Щелкните по **ОК**. Результат модификации диаграммы дерева узлов показан на рис. 8.

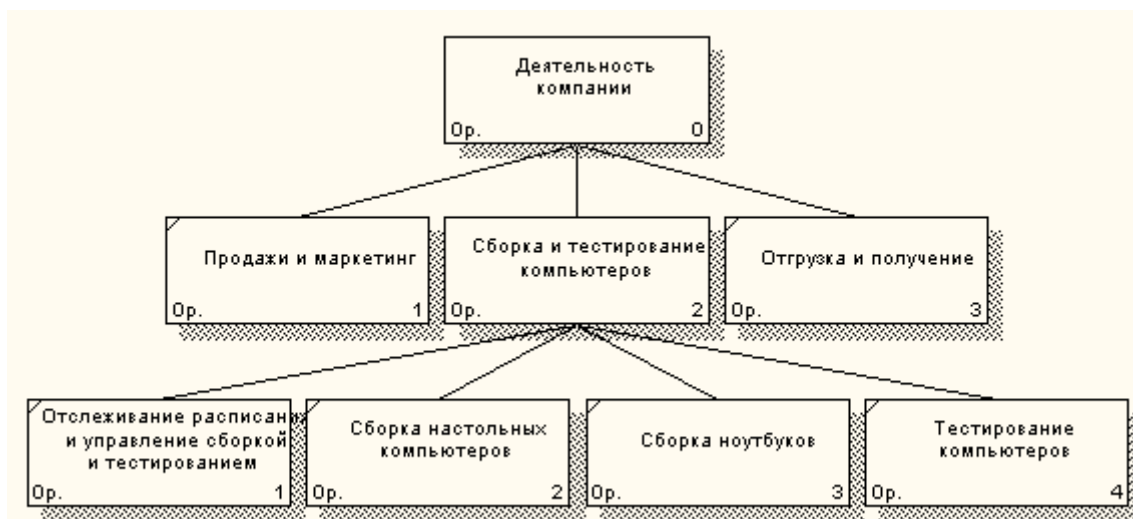


Рис. 8. Диаграмма дерева узлов

Создание FEO диаграммы

Предположим, что при обсуждении бизнес-процессов возникла необходимость детально рассмотреть взаимодействие работы "Сборка и тестирование компьютеров" с другими работами. Чтобы не портить диаграмму декомпозиции, создайте FEO-диаграмму (FEO – расшифровывается как «только для экспозиции»), на которой будут только стрелки работы "Сборка и тестирование компьютеров".

1. Выберите пункт главного меню **Diagram/Add FEO Diagram** (рис. 9).

2. В диалоговом окне **Add New FEO Diagram** выберите тип и внесите имя диаграммы **FEO** как показано на рис. 10. Щелкните по кнопке **ОК**.

3. Для определения содержания диаграммы перейдите в пункт меню **Diagram/Diagram Properties** и во вкладке **Diagram Text** внесите определение (рис. 11).

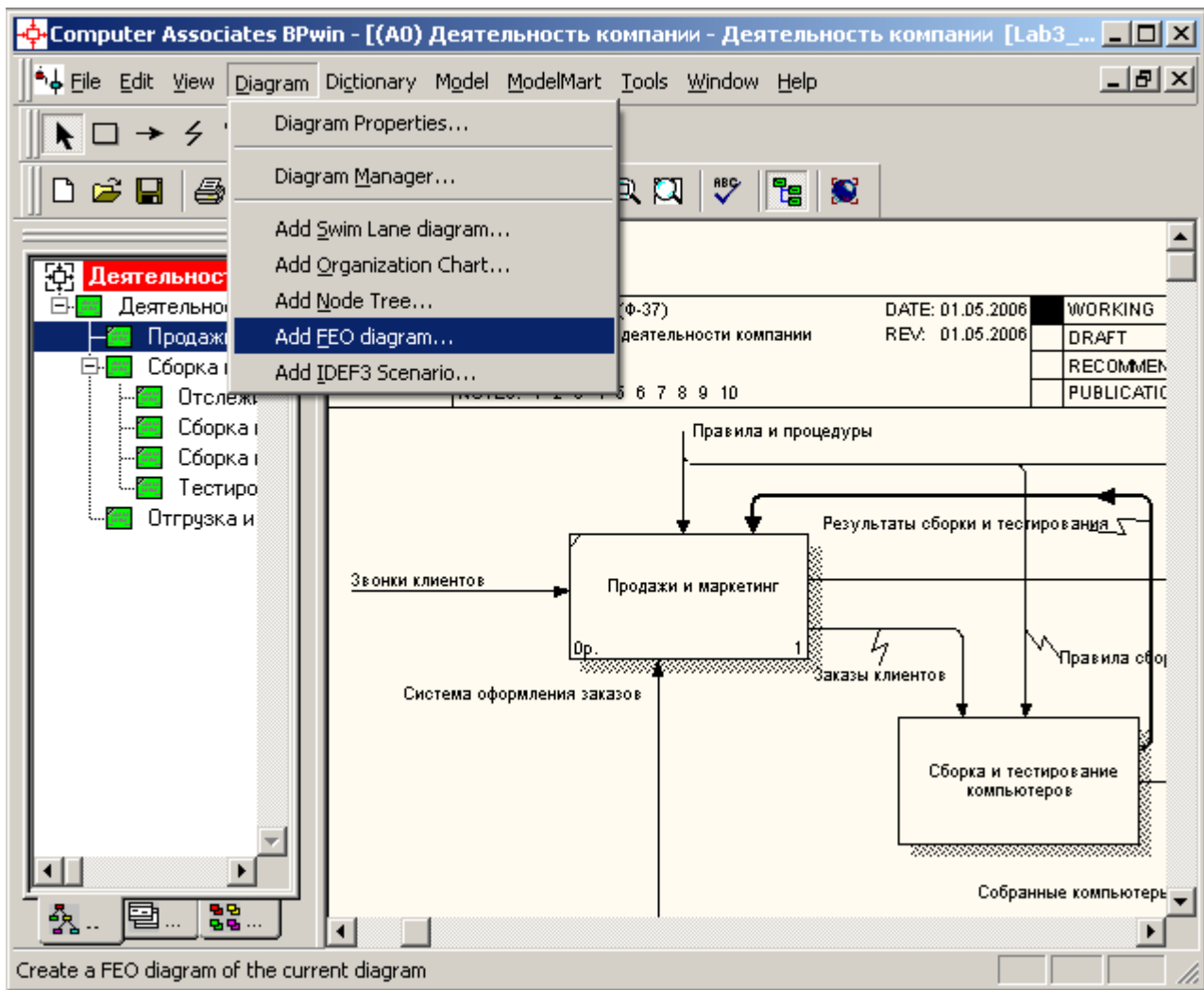


Рис. 9. Диалоговое окно Add New FEO Diagram

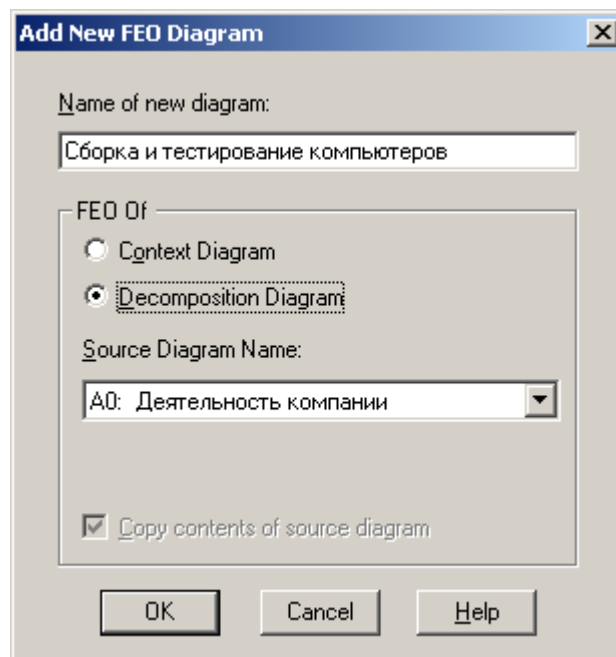


Рис. 10. Диалоговое окно Add New FEO Diagram

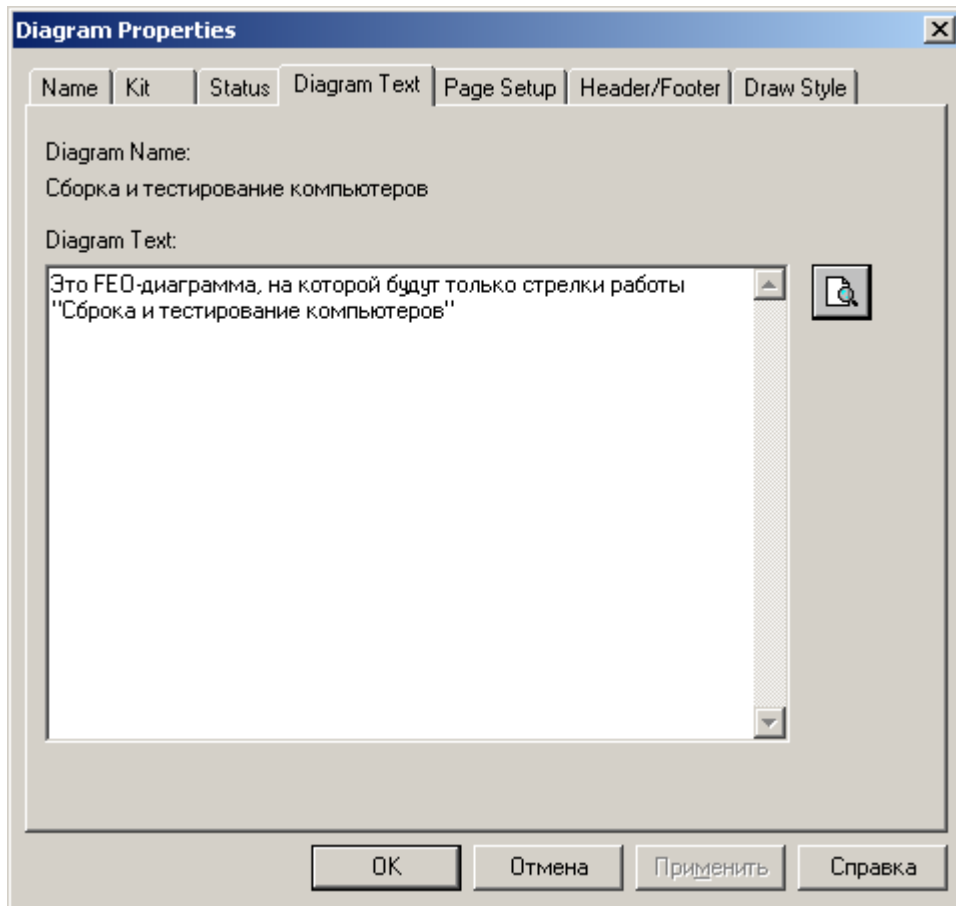


Рис. 11. Вкладка Diagram Text диалогового окна Diagram Properties

4. Удалите лишние стрелки на диаграмме FEO. Результат показан на рис. 12.

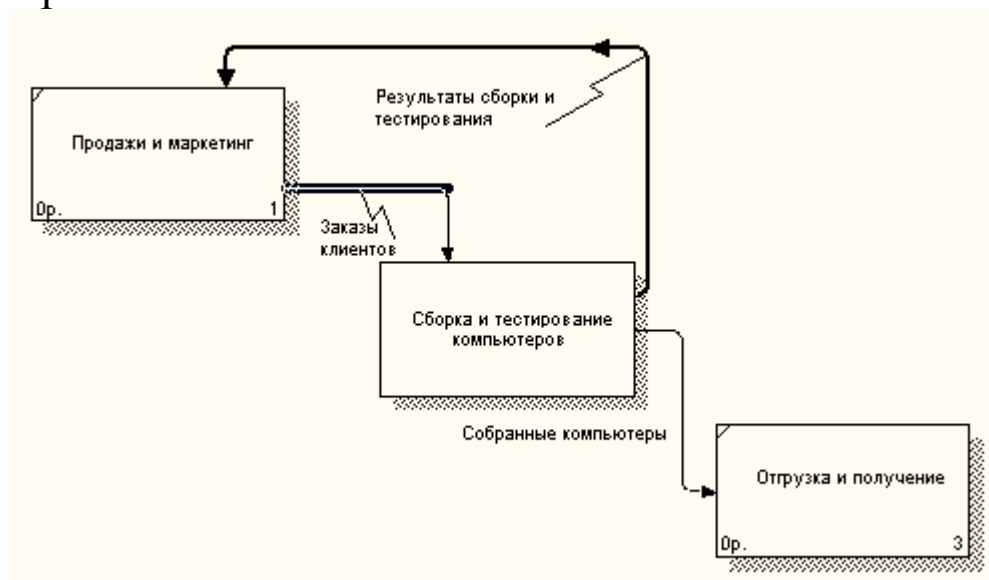



Рис. 12. Диаграмма FEO

Для перехода между стандартной диаграммой, деревом узлов и FEO используйте кнопку  на палитре инструментов.

Тема 4. Стоимостный анализ

Как было указано ранее, обычно сначала строится функциональная модель существующей организации работы - AS-IS (как есть). После построения модели AS-IS проводится анализ бизнес-процессов, потоки данных и объектов перенаправляются и улучшаются, в результате строится модель TO-BE. Как правило, строится несколько моделей TO-BE, из которых по какому-либо критерию выбирается наилучшая. Проблема состоит в том, что таких критериев много и непросто определить важнейший. Для того чтобы определить качество созданной модели с точки зрения эффективности бизнес-процессов, необходима система метрики, т. е. качество следует оценивать количественно.

ВРwin предоставляет аналитику два инструмента для оценки модели - стоимостный анализ, основанный на работах (Activity Based Costing, ABC), и свойства, определяемые пользователем (User Defined Properties, UDP). ABC является широко распространенной методикой, используемой международными корпорациями и государственными организациями (в том числе Департаментом обороны США) для идентификации истинных движителей затрат в организации.

Стоимостный анализ представляет собой соглашение об учете, используемое для сбора затрат, связанных с работами, с целью определить общую стоимость процесса. Стоимостный анализ основан на модели работ, потому что количественная оценка невозможна без детального понимания функциональности предприятия. Обычно ABC применяется для того, чтобы понять происхождение выходных затрат и облегчить выбор нужной модели работ при реорганизации деятельности предприятия (Business Process Reengineering, BPR). С помощью стоимостного анализа можно решить такие задачи, как определение действительной стоимости производства продукта, определение действительной стоимости поддержки клиента, идентификация работ, которые стоят больше всего (те, которые должны быть

улучшены в первую очередь), обеспечение менеджеров финансовой мерой предлагаемых изменений т. д.

ABC может проводиться только тогда, когда модель работы последовательная (следует синтаксическим правилам IDEF0), корректная (отражает бизнес), полная (охватывает всю рассматриваемую область) и стабильная (проходит цикл экспертизы без изменений), другими словами, создание модели работы закончено.

ABC включает следующие основные понятия:

- объект затрат - причина, по которой работа выполняется, обычно, основной выход работы, стоимость работ есть суммарная стоимость объектов затрат;
- движитель затрат - характеристики входов и управлений работы, которые влияют на то, как выполняется и как долго длится работа;
- центры затрат, которые можно трактовать как статьи расхода.

При проведении стоимостного анализа в VPwin сначала задаются единицы измерения времени и денег. Для задания единиц измерения следует вызвать диалог **Model Properties**, закладка **ABC Units**. Если в списке выбора отсутствует необходимая валюта (например, рубль), ее можно добавить. Символ валюты по умолчанию берется из настроек Windows. Диапазон измерения времени в списке **Unit of measurment** достаточен для большинства случаев - от секунд до лет.

Затем описываются центры затрат (cost centers). Для внесения центров затрат необходимо вызвать диалог **Cost Center Editor**. Каждому центру затрат следует дать подробное описание в окне **Definition**. Список центров затрат упорядочен. Порядок в списке можно менять при помощи стрелок, расположенных справа от списка. Задание определенной последовательности центров затрат в списке, во-первых, облегчает последующую работу при присвоении стоимости работам, а во-вторых, имеет значение при использовании единых стандартных отчетов в разных моделях. Хотя VPwin сохраняет информацию о стандартном отчете в файле VPWINRPT.INI, информация о центрах затрат и UDP сохраняется в виде указателей, т. е. хранятся не названия центров

затрат, а их номера. Поэтому, если нужно использовать один и тот же стандартный отчет в разных моделях, списки центров затрат должны быть в них одинаковы.

Для задания стоимости работы (для каждой работы на диаграмме декомпозиции) следует щелкнуть правой кнопкой мыши по работе и на всплывающем меню выбрать **Costs**. В диалоге **Activity Cost** указывается частота проведения данной работы в рамках общего процесса (окно **Frequency**) и продолжительность (**Duration**). Затем следует выбрать в списке один из центров затрат и в окне **Cost** задать его стоимость. Аналогично назначаются суммы по каждому центру затрат, т. е. задается стоимость каждой работы по каждой статье расхода. Если в процессе назначения стоимости возникает необходимость внесения дополнительных центров затрат, диалог **Cost Center Editor** вызывается прямо из диалога **Activity Cost** соответствующей кнопкой.

Общие затраты по работе рассчитываются как сумма по всем центрам затрат. При вычислении затрат вышестоящей (родительской) работы сначала вычисляется произведение затрат дочерней работы на частоту работы (число раз, которое работа выполняется в рамках проведения родительской работы), затем результаты складываются. Если во всех работах модели включен режим **Compute from Decompositions**, подобные вычисления автоматически проводятся по всей иерархии работ снизу вверх.

Для проведения более тонкого анализа можно воспользоваться специализированным средством стоимостного анализа EasyABC (ABC Technology, Inc.). ВРwin имеет двунаправленный интерфейс с EasyABC. Для экспорта данных в EasyABC следует выбрать пункт меню **File/Export/Node Tree**, задать в диалоге **Export Node Tree** необходимые настройки и экспортировать дерево узлов в текстовый файл (.txt). Файл экспорта можно импортировать в EasyABC. После проведения необходимых расчетов результирующие данные можно импортировать из EasyABC в ВРwin. Для импорта нужно выбрать меню **File/Import/Costs** и в диалоге **Import Activity Costs** выбрать необходимые установки.

Лабораторная работа №4

Цель работы: Научиться осуществлять оценку модели производственного процесса с применением стоимостного анализа в среде VPwin 4.0.

Постановка задачи: Выполнить стоимостный анализ для производственного процесса согласно выбранному варианту задания. Варианты производственных процессов приведены в задании к лабораторной работе № 1.

Результат: Отчет по лабораторной работе с описанием процедуры выполнения стоимостного анализа и стоимостным отчетом.

Пример выполнения работы

1. В диалоговом окне **Model Properties** (вызывается из меню **Model/Model Properties**) во вкладке **ABC Units** (рис. 1) установите единицы измерения денег – “у.е.” и времени - часы.

2. Перейдите в меню **Dictionary/Cost Center** (Словарь/Центр Затрат) (рис. 2) и в окне **Cost Center Dictionary** (Словарь Центра Затрат) (рис. 3) внесите название и определение центров затрат (таблица 1). Вид окна **Cost Center Dictionary** после внесения названия и определение центров затрат представлен на рис. 4 (обратите внимание на то, что центры затрат упорядочились по алфавиту).

Таблица 1. Центры затрат ABC

Центр затрат	Определение
Управление	Затраты на управление, связанные с составлением графика работ, формированием партий компьютеров, контролем над сборкой и тестированием
Рабочая сила	Затраты на оплату рабочих, занятых сборкой и тестированием компьютеров
Компоненты	Затраты на закупку компонентов

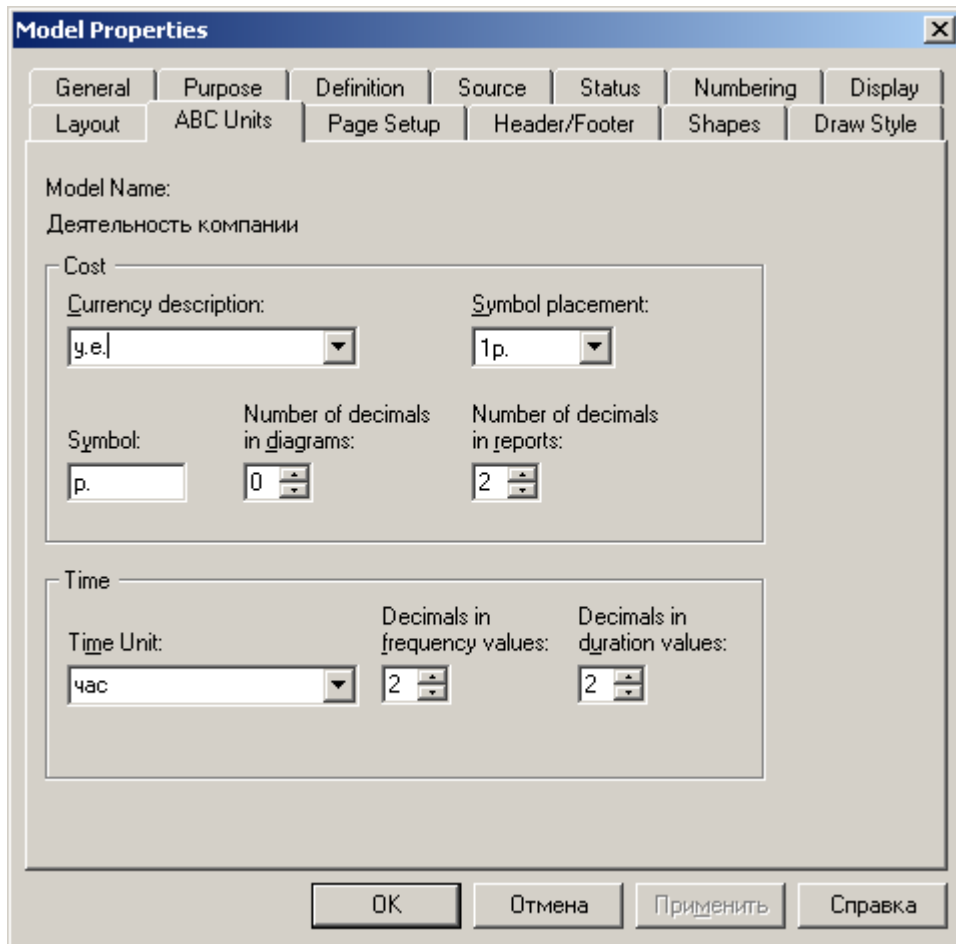


Рис. 1. Вкладка ABC Units диалога Model Properties

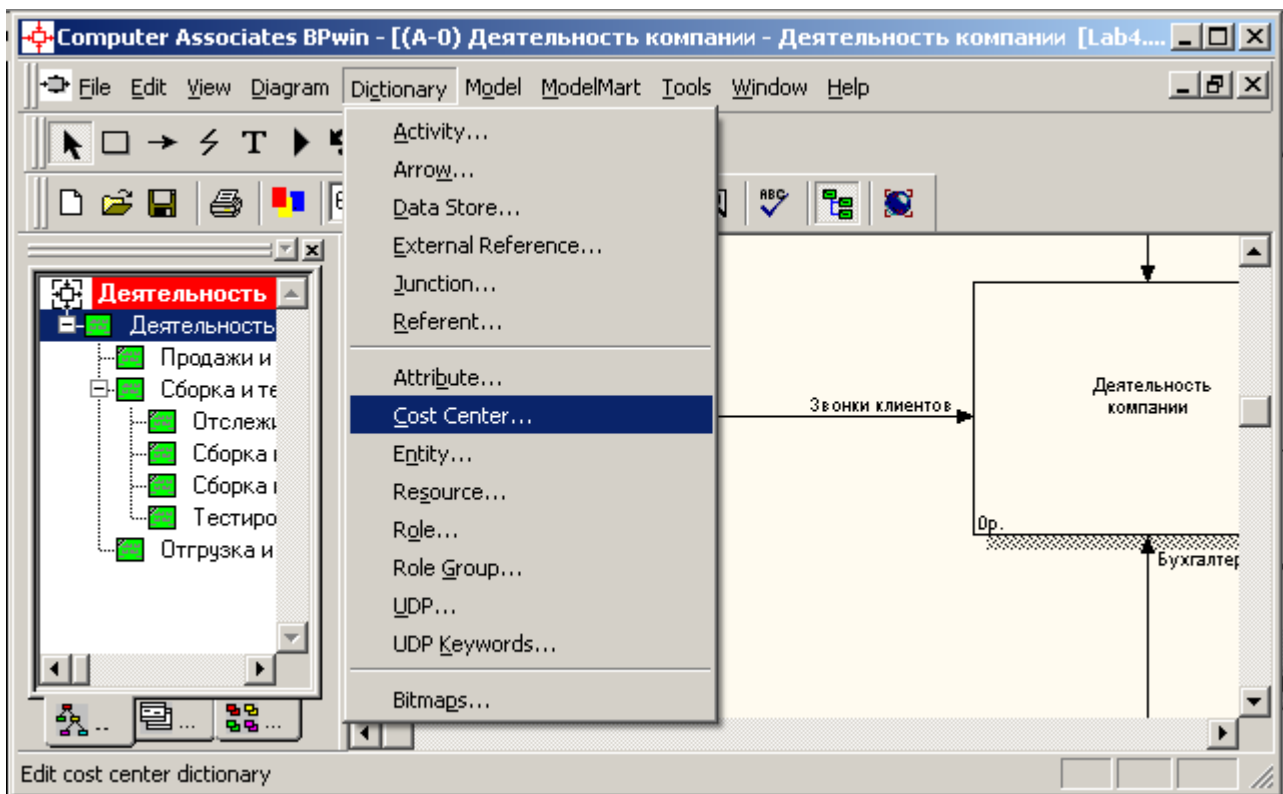


Рис. 2. Выбор меню Dictionary/Cost Center

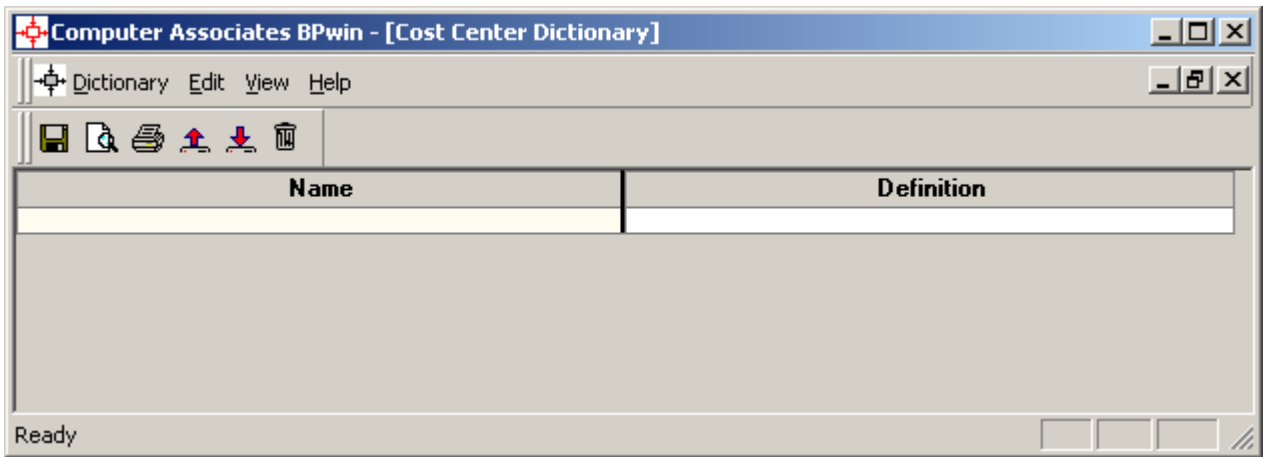


Рис. 3. Незаполненное окно Cost Center Dictionary

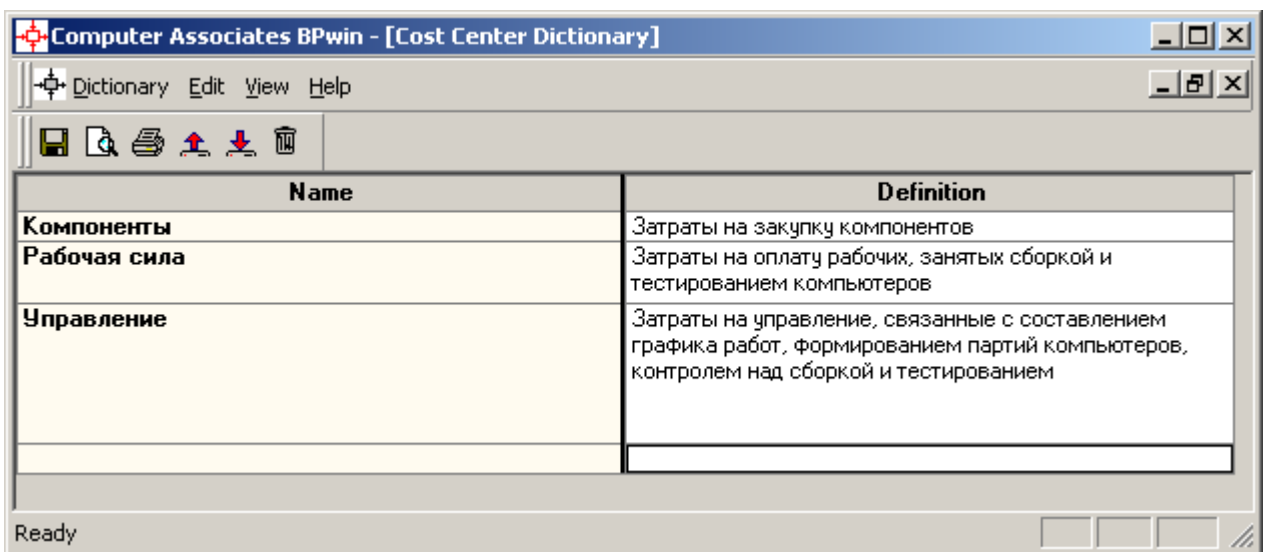


Рис. 4. Заполненное окно Cost Center Dictionary

Для отображения стоимости каждой работы в нижнем левом углу прямоугольника перейдите в меню **Model/Model Properties** и во вкладке **Display** диалога **Model Properties** включите опцию **ABC Data** (рис. 5).

Для отображения частоты или продолжительности работы переключите радио-кнопки в группе **ABC Units**.

Для назначения стоимости работе "Сборка настольных компьютеров" следует на диаграмме A2 (рис. 6) щелкнуть по ней правой кнопкой мыши и выбрать в контекстном меню **Cost** (рис. 7).

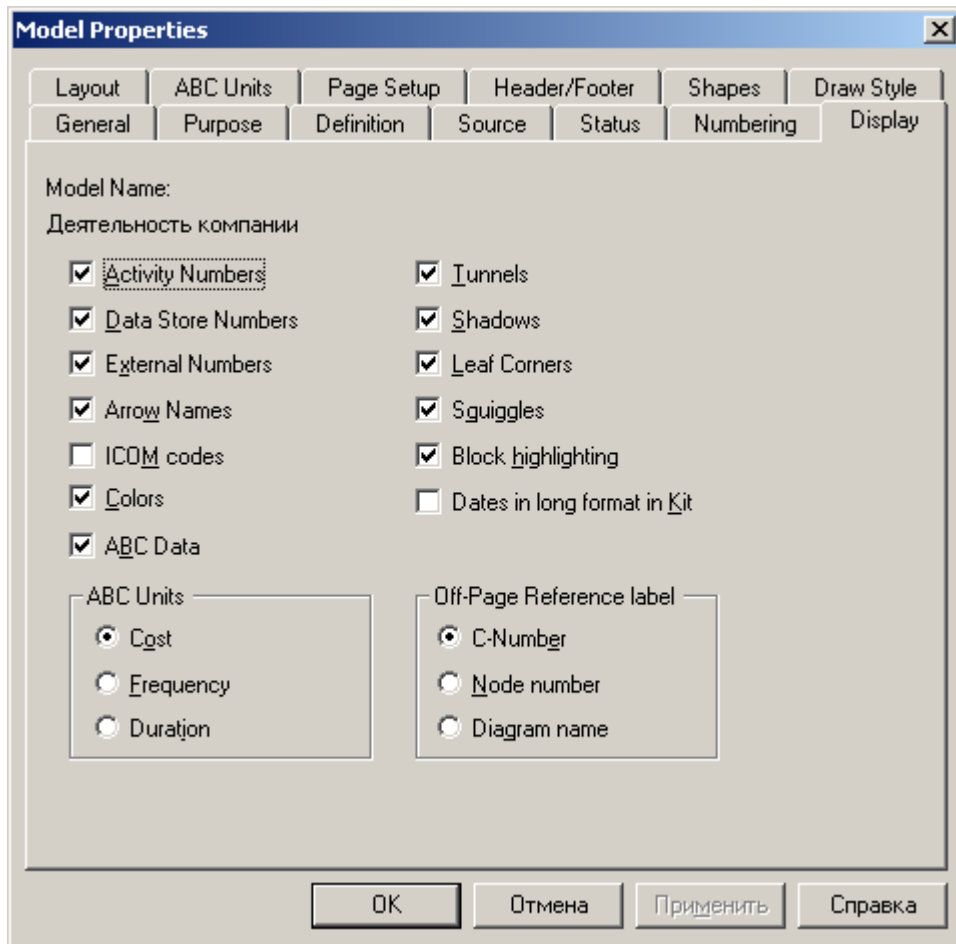


Рис. 5. Вкладка Display диалога Model Properties

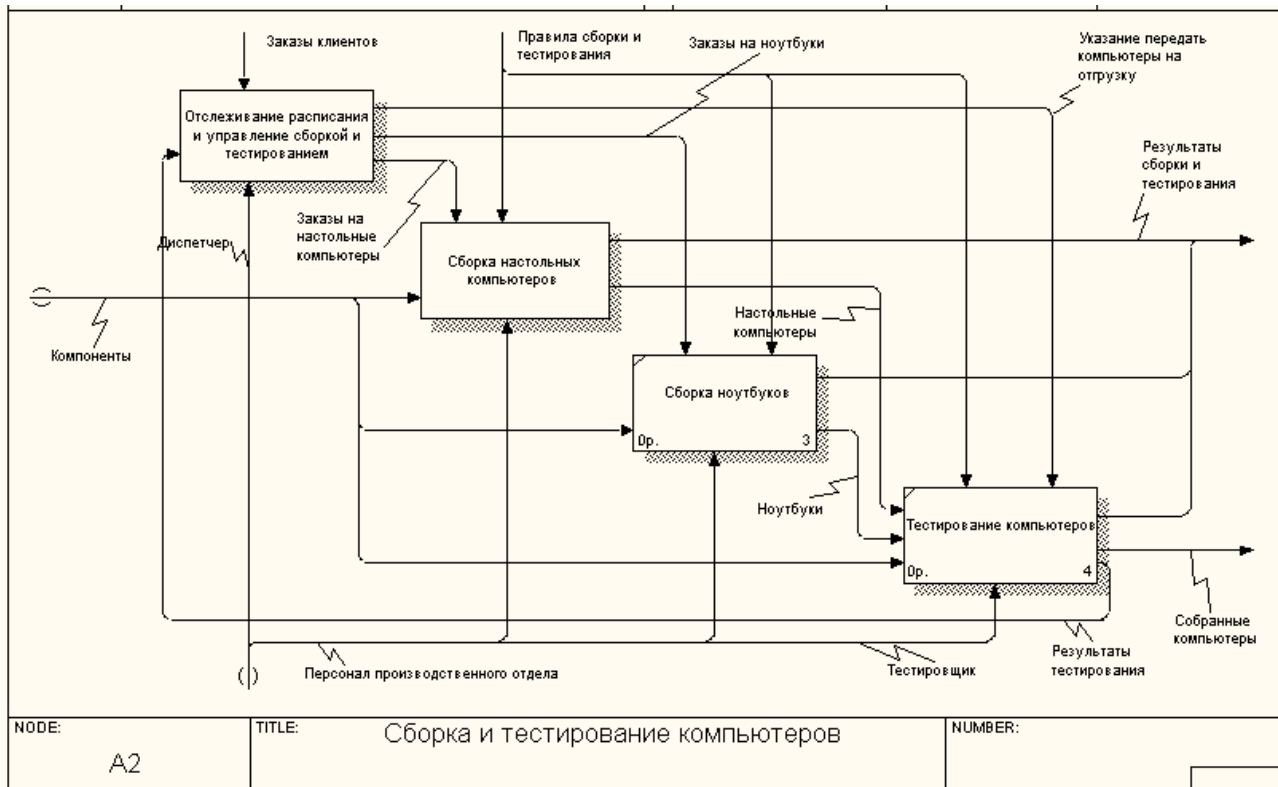


Рис. 6. Диаграмма A2

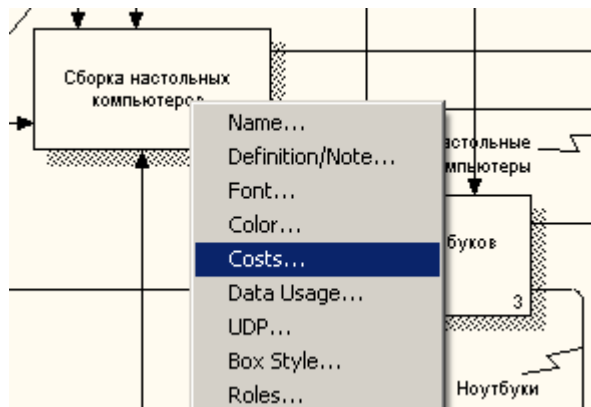


Рис. 7. Выбор в контекстном меню опции Costs

Откроется диалоговое окно **Activity Properties** (рисунок 8) в котором следует указать величины затрат (в у.е.) на компоненты, рабочую силу, управление и временные характеристики работы – **Duration** (Продолжительность) и **Frequency** (Частоту) выполнения (см. таблицу 2).

Cost Center	y.e.
Компоненты	1 000,00
Рабочая сила	5,00
Управление	0,00

This Activity has NO Decomposition. Total cost: 1 005,00

Override decompositions Total cost x Frequency: 8 040,00

Compute from decompositions

Frequency:

Duration: час

Duration x Frequency 16,00 час

Рис. 8. Вкладка Cost диалога Activity Properties

3. Для работ на диаграмме A2 внесите параметры ABC (таблица 2).

Таблица 2 – Показатели стоимости работ на диаграмме A2

Activity Name	Cost Center	Cost Center Cost, y.e.	Duration, час	Frequency
Отслеживание расписания и управление сборкой и тестированием	Управление	20,00	0,50	14,00
Сборка настольных компьютеров	Рабочая сила	10,00	2,00	8,00
	Компоненты	500,00		
Сборка ноутбуков	Рабочая сила	13,00	4,00	6,00
	Компоненты	1000,00		
Тестирование компьютеров	Рабочая сила	8,00	1,00	14,00

Посмотрите результат - стоимость работы верхнего уровня (рис. 9).



Рис. 9. Отображение стоимости в нижнем левом углу прямоугольника

4. Выбрав соответствующие опции меню (рис. 10), сгенерируйте отчет **Activity Cost Report**.

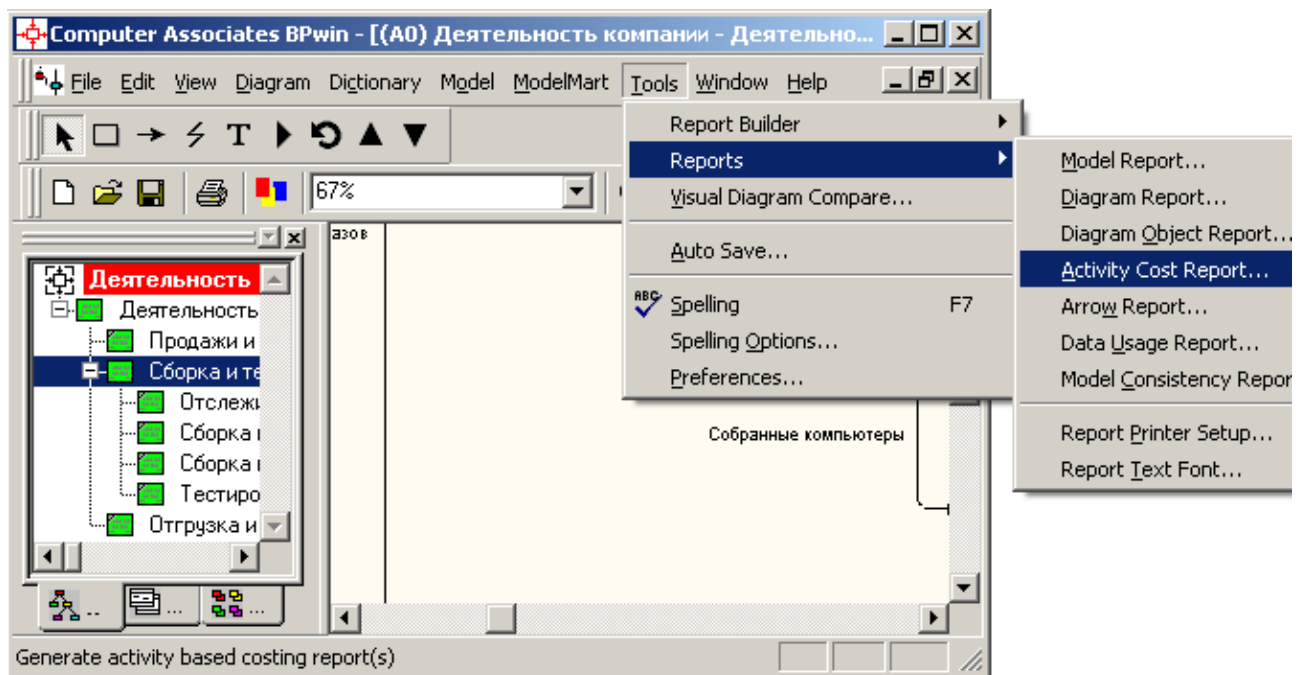


Рис. 10. Выбор опций меню для генерации отчета Activity Cost Report

В открывшемся диалоговом окне **Activity Based Costing Report** задайте параметры генерации отчета (рис. 11).

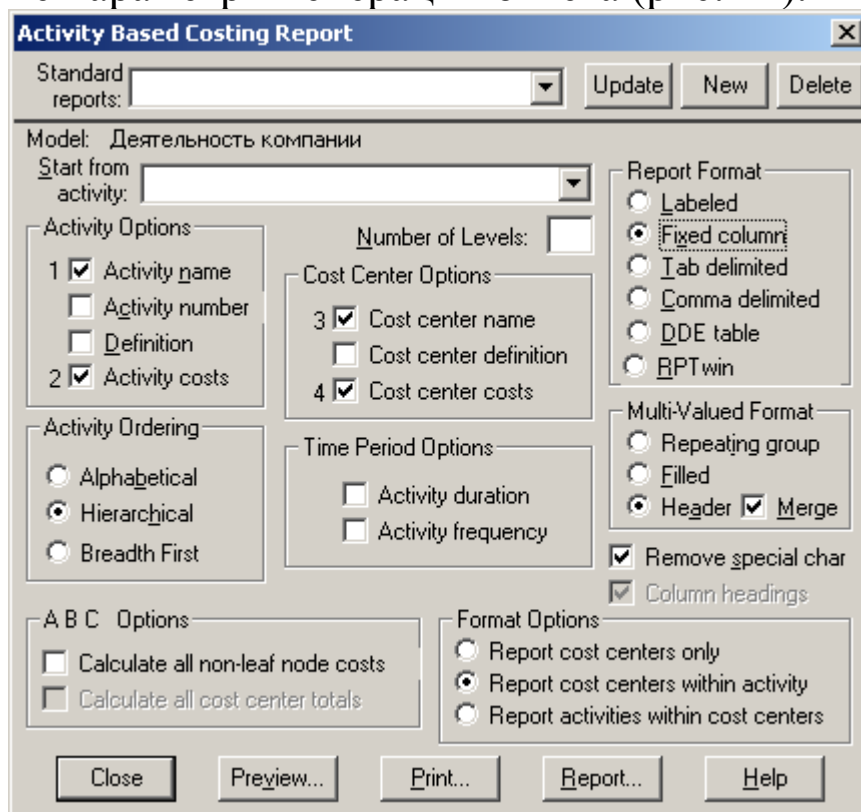


Рис. 11. Задание параметров генерации отчета Activity Cost Report

Activity Name	Activity Cost (y.e.)	Cost Center	Cost Center Cost (y.e.)
Деятельность компании	10 550,00	Компоненты	10 000,00
		Рабочая сила	270,00
		Управление	280,00
Продажи и маркетинг	0,00		
Сборка и тестирование компьютеров	10 550,00	Компоненты	10 000,00
		Рабочая сила	270,00
		Управление	280,00
Отслеживание расписания и управление сборкой и тестированием	20,00	Управление	20,00
Сборка настольных компьютеров	510,00	Компоненты	500,00
		Рабочая сила	10,00
Сборка ноутбуков	1 013,00	Компоненты	1 000,00
		Рабочая сила	13,00
Тестирование компьютеров	8,00	Рабочая сила	8,00
Отгрузка и получение	0,00		

Рис. 12. Фрагмент отчета Activity Cost Report

ЛИТЕРАТУРА

1. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика, 1998.
2. Заикин О.А., Советов Б.Я. Проектирование интегрированных систем обработки информации и управления. Учебное пособие. – М.: МГАП «Мир Книги», 1994.
3. Зиндер Е.З. Бизнес-реинжиниринг и новое системное проектирование. – М.: Син-тег, 1997.
4. Калянов Г.Н. CASE-технологии. – М.: Финансы и статистика, 1998.
5. Липаев В.В. Системное проектирование сложных программных средств для информационных систем. – М.: Синтег, 1999.
6. Мамиконов А.Г. Основы построения АСУ. – М.: Высшая школа, 1981.
7. Мамиконов А.Г. Проектирование АСУ. Учебник. – М.: Высшая школа, 1987.
8. Основы построения АСУ / Под ред. Б.Я. Советова. – Ташкент.: Укитувчи, 1984.
9. Советов Б.Я. АСУ. Введение в специальность. – М.: Высшая школа, 1989.
10. Советов Б.Я., Цехановский В.В. Автоматизированное управление современным производством. Серия «ЭВМ в производстве». – Л.: Машиностроение, 1988.

Для заметок

УЧЕБНОЕ ИЗДАНИЕ

НИКИШАЕВ Василий Александрович

ОСНОВЫ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ

**Практическое пособие по выполнению лабораторных работ
для студентов специальности I-53 01 02 – «АСОИ»**

В авторской редакции

Подписано в печать . (). Формат 60x84 1/16. Бумага писчая №1.
Печать на ризографе. Гарнитура Times New Roman. Усл.печ.л.
4,5. Уч-изд.л. 3,0. Тираж 35 экз.

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»,
246019, г.Гомель, ул.Советская, 104

Отпечатано в учреждении образования
«Гомельский государственный университет
имени Франциска Скорины»
246019, г.Гомель, ул.Советская, 104

