Пензенский государственный университет кафедра "Информационно-вычислительные системы"

В. И. Горбаченко, Г. Ф. Убиенных, Г. В. Бобрышева

СОЗДАНИЕ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ИНФОРМАЦИОННОЙ СИСТЕМЫ С ПОМОЩЬЮ CASE-cpeдства CA ERwin Process Modeler 7.3

УЧЕБНОЕ ПОСОБИЕ

Γ67

Горбаченко В. И., Убиенных Г. Ф., Бобрышева Г. В.

Г67 Создание функциональной модели информационной системы с помощью CASE-средства CA ERwin Process Modeler 7.3. – Пенза: ПГУ, 2010. – 66 с.

Учебное пособие посвящено созданию моделей предметной области с помощью CASE-средства CA ERwin Process Modeler 7.3. Приводятся краткие сведения о методологиях IDEF0, DFD и IDEF3. На конкретных примерах рассматривается построение функциональных моделей информационной системы в стандартах IDEF0, DFD и IDEF3.

Пособие предназначено для подготовки специалистов, бакалавров и магистров по направлениям "Прикладная информатика" и "Информатика и вычислительная техника". Пособие может также использоваться студентами других направлений, осваивающих проектирование информационных систем.

СОДЕРЖАНИЕ

1. МЕТОДОЛОГИИ МОДЕЛИРОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ	4
1.1. Функциональная методология IDEF0	4
1.2. Методология DFD	
1.3. Методология IDEF3	
2. СОЗДАНИЕ МОДЕЛИ В СТАНДАРТЕ IDEF0	
2.1. Создание контекстной диаграммы	
2.2. Создание диаграмм декомпозиции	
2.3. Создание диаграммы дерева узлов	
2.4. Создание FEO-диаграммы	
2.5. Расщепление и слияние моделей	
2.6. Задание для самостоятельной работы	55
3. СОЗДАНИЕ МОДЕЛИ В СТАНДАРТЕ DFD	
3.1. Создание контекстной диаграммы	57
3.2. Создание диаграммы декомпозиции	58
3.3. Задание для самостоятельной работы	60
4. СОЗДАНИЕ МОДЕЛИ В СТАНДАРТЕ IDEF3	61
4.1. Создание диаграммы декомпозиции	61
4.2. Задание для самостоятельной работы	64
ЛИТЕРАТУРА	65

1. МЕТОДОЛОГИИ МОДЕЛИРОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Функциональная методология IDEF0

Наиболее популярной методологией IDEF является методология IDEF0 [1 – 4]. Методологию IDEF0 можно считать следующим этапом развития хорошо известного графического языка описания функциональных систем SADT (Structured Analysis and Design Technique – Техника структурного анализа и дизайна) [5]. Исторически IDEF0 как стандарт был разработан в 1981 году в рамках обширной программы автоматизации промышленных предприятий, которая носила обозначение ICAM (Integrated Computer Aided Manufacturing). Семейство стандартов IDEF унаследовало свое обозначение от названия этой программы (IDEF=Icam DEFinition), и последняя его редакция была выпущена в декабре 1993 года Национальным Институтом по Стандартам и Технологиям США (NIST). В России приняты официальные рекомендации по применению методологии IDEEF0 [6].

Целью методологии является построение функциональной схемы исследуемой системы, описывающей все необходимые процессы с точностью, достаточной для однозначного моделирования деятельности системы. Другими словами, в IDEF0 моделируемая система представляется как совокупность взаимосвязанных работ (функций, активностей). Методология IDEF0 получила столь широкое распространение в бизнес—моделировании потому, что эта методология легко представляет такие системные характеристики, как управление, обратная связь, исполнители. Кроме того, методология IDEF0 имеет развитые процедуры поддержки коллективной работы.

В основе методологии лежат четыре основных понятия: функциональный блок, дуга (стрелка), декомпозиция, глоссарий.

Функциональный блок, или работа (Activity Box) представляет собой некоторую конкретную функцию (работу) в рамках рассматриваемой системы. Блок должен иметь название в глагольном наклонении (например, "Проверить документ" или "Проверка документа"). На диаграмме функциональный блок изображается прямоугольником (рис. 1.1). Каждая из четырех сторон функционального блока имеет свое определенное значение (роль) и определяет тип интерфейса, т. е. способ взаимодействия дуги с блоком:

- верхняя сторона имеет значение "Управление" (Control);
- левая сторона имеет значение "Вход" (Input);
- правая сторона имеет значение "Выход (Output);
- нижняя сторона имеет значение "Mexaнизм" (Mechanism).



Рис. 1.1. Функциональный блок

Дуга (Arrow) отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, представленную данным функциональным блоком. Интерфейсные дуги часто называют потоками или стрелками.

С помощью дуг отображают различные объекты, которые передаются между блоками, обрабатываются блоками, определяют правила обработки и механизмы обработки. Такими объектами могут быть элементы реального мира (детали, вагоны, сотрудники и т.д.) или потоки данных и информации (документы, данные, инструкции и т.д.). Стрелки снабжаются надписями – названиями.

В зависимости от того, к какой из сторон функционального блока подходит данная интерфейсная дуга, она носит название "входящей", "исходящей", "управляющей", "механизмом" или вызовом.

В IDEF0 различают *пять типов стрелок*.

Bxod (Input) — материальные объекты или информация, которые используются или преобразуются pafomou для получения результата (выхода). Допускается, что блок может не иметь ни одной стрелки входа.

При описании технологических процессов не возникает проблем определения входов. Вход — это нечто, что перерабатывается в блоке для получения результата. При моделировании информационной системы, когда стрелками являются не физические объекты, а данные, определение входа может вызвать трудности. Например, чтобы показать переработку данных блоком, целесообразно на входе указать "Документ", а на выходе — "Заполненный документ". Например, не может быть входом блока "Прием экзамена" стрелка "Студент", а выходом — стрелка "Экзаменационная ведомость", т. к. студент не перерабатывается в ведомость. В данном примере можно использовать входную стрелку "Не аттестованный студент" и выходную стрелку "Аттестованный студент". Очень часто сложно определить, являются ли данные входом или управлением. В этом случае подсказкой может служить информация о том, перерабатываются/изменяются ли данные в блоке или нет. Если изменяются, то, скорее всего, это вход, если нет — управление. Например, задание на курсовой проект является управлением, а не входом.

Управление (Control) — правила, стратегии, процедуры, стандарты, ограничения на бюджет и время, которыми руководствуется работа. Каждая работа должна иметь хотя бы одну стрелку управления. Управление влияет на работу, но не преобразуется работой. В случае возникновения неопреде-

ленности в статусе *стрелки* (управление или вход) рекомендуется рисовать *стрелку* управления.

Выход (Output) — материальный объект или информация, которые производятся работой. Каждая работа должна иметь хотя бы одну стрелку выхода. Работа без результата не имеет смысла и не должна моделироваться.

Механизм (Mechanism) – ресурсы, которые выполняют работу, например персонал предприятия, станки, устройства и т. д. *По усмотрению аналитика* стрелки *механизма могут не изображаться в модели*.

Вызов (Call) — специальная стрелка, указывающая на другую модель работы. Стрелка вызова используется при расщеплении модели и указывает, что некоторая работа представлена отдельной моделью. Расщепление моделей необходимо для коллективной работы над моделью. Руководитель проекта может создать декомпозицию верхнего уровня и провести расщепление модели на отдельные модели. Аналитики работают над отдельными моделями, а затем сливают отдельные модели в единую модель. Отдельная ветвь модели может быть отщеплена для использования в качестве независимой модели.

Таким образом, любой функциональный блок по требованиям стандарта должен иметь, по крайней мере, одну управляющую дугу и одну исходящую. То есть каждый процесс должен происходить по каким-то правилам (отображаемым управляющей дугой) и должен выдавать некоторый результат (выходящая дуга), иначе его рассмотрение не имеет никакого смысла.

Декомпозиция (Decomposition) является основным понятием стандарта IDEF0. Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции. Декомпозиция позволяет постепенно и структурировано представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко усваиваемой.

Глоссарий (Glossary) – это набор соответствующих определений, ключевых слов, повествовательных изложений и т.д., которые характеризуют объект, отображенный данным элементом. Этот набор является описанием сущности данного элемента. Глоссарий дополняет наглядный графический язык, снабжая диаграммы необходимой дополнительной информацией.

Модель в методологии IDEF0 — это совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе. Модель может содержать четыре типа диаграмм:

- 1. Контекстная диаграмма, которая представляет всю систему как один блок и показывает контекст системы, т. е. связь системы с внешним миром. Модель может иметь только одну контекстную диаграмму.
- 2. Диаграммы декомпозиции, которые получаются в результате разбиения контекстной диаграммы на отдельные активности. Такой процесс называется функциональной декомпозицией, а диаграммы, получившиеся в результате декомпозиции, называются диаграммами декомпозиции. После де-

композиции контекстной диаграммы производится декомпозиция каждой получившейся диаграммы и т. д.

Декомпозиция продолжается до достижения нужного уровня подробности описания. После каждого сеанса декомпозиции проводятся *сеансы экспертизы* — эксперты предметной области проверяют 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 каждом уровне модели. Синтаксис описания системы в целом и каждого ее фрагмента одинаков во всей модели.

- 3. Диаграммы дерева узлов показывают иерархическую зависимость работ. То есть, в виде дерева показывается, какие активности получились в результате декомпозиции каждой активности. Диаграмм деревьев узлов может быть в модели несколько, поскольку дерево может быть построено на различную глубину и начиная с любой диаграммы (не обязательно с контекстной).
- 4. Диаграммы только для экспозиции (FEO for exposition only) строятся для иллюстрации альтернативной точки зрения, для хранения старых версий. FEO это просто картинка. Дело в том, что методология не поддерживает альтернативные варианты декомпозиции. Если необходимо хоть както сохранить альтернативный вариант декомпозиции, то применяют диаграмму только для экспозиции.

Рассмотрим подробнее различные виды диаграмм.

Модель IDEF0 всегда начинается с представления системы как единого целого — одной активности с дугами, простирающимися за пределы рассматриваемой области. Такая диаграмма с одной активностью называется контекстной диаграммой диаграммой. Дуги контекстной диаграммы должны описывать все основные связи моделируемой системы с внешним миром.

Методология IDEF0 подразумевает, что модель является не просто совокупностью диаграмм, а содержит всю необходимую информацию о моделируемой области. Информация о модели задается в свойствах модели. В CA ERwin Process Modeler (для простоты далее будем использовать старое название: BPwin) информация задается в диалоге свойств модели (Model Properties) [19].

В общих свойствах (General) указываются имя модели, название проекта, автор модели, временные рамки модели (Time Frame) — AS-IS и ТО-ВЕ. Обычно сначала строится модель существующей организации работы — AS-IS (как есть). 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. Найденные в модели

AS-IS недостатки можно исправить при создании модели **TO-BE** (как будет) — модели новой организации бизнес—процессов. 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 бизнеспроцесс.

Цель моделирования (**Purpose**) определяется из ответов на следующие вопросы:

- Почему этот процесс должен быть смоделирован?
- Что должна показывать модель?
- Что может получить клиент?

Точка зрения (Viewpoint) — это перспектива, с которой наблюдалась система при построении модели. Хотя при построении модели учитываются мнения различных людей, все они должны придерживаться единой точки зрения на модель. Точка зрения должна соответствовать цели и границам моделирования. Как правило, выбирается точка зрения человека, ответственного за моделируемую работу в целом.

Даются также определение модели (**Definition**) и описание области действия модели (**Scope**).

Указываются источники получения данных о модели (**Source**), например, "Опрос экспертов предметной области и анализ документации".

Статус модели (**Status**) — это рабочая версия (новая модель, не прошедшая экспертиз) — **WORKING**, черновой вариант (модель прошла первичную экспертизу) — **DRAFT**, рекомендовано (прошла экспертизу) — **RECOM- MENDED**, публикация (окончательный вариант) — **PUBLICATION**.

Каждая активность может быть подвергнута декомпозиции на другой — "дочерней" диаграмме (Child Diagram). Каждая диаграмма нижнего уровня показывает "внутреннее" строение активности на родительской диаграмме (Parent Diagram). Каждая из активностей дочерней диаграммы может быть далее детализирована путем аналогичной декомпозиции. В каждом случае декомпозиции функционального блока все интерфейсные дуги, входящие в данный блок или исходящие из него, фиксируются на дочерней диаграмме. Этим достигается структурная целостность IDEF0-модели.

Чтобы сделать диаграммы удобочитаемыми, в стандарте IDEF0 приняты *ограничения сложности*: на диаграмме может быть от трех до шести активностей (в BPwin – от 2 до 8), при этом количество подходящих к одной активности и выходящих из одной активности дуг предполагается не более четырех.

Работы на диаграммах декомпозиции обычно располагаются в так называемом *порядке доминирования* — по диагонали от левого верхнего угла к правому нижнему. 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.

Все активности модели нумеруются. *Номер* состоит из префикса и числа. Может быть использован префикс любой длины, но обычно используют префикс **A**. Контекстная активность имеет номер **A0**. Активности, полученные в результате декомпозиции контекстной активности номера **A1**, **A2**, **A3** и т. д. Работы декомпозиции нижнего уровня имеют номер родительской активности и очередной порядковый номер, например активности декомпозиции **A3** будут иметь номера **A31**, **A32**, **A33**, **A34** и т. д. Активности образуют иерархию, где каждая активность может иметь одну родительскую и несколько дочерних работ, образуя дерево. Такое дерево называют деревом узлов, а вышеописанную нумерацию – нумерацией по узлам.

Диаграммы IDEF0 имеют двойную нумерацию. Во-первых, диаграммы имеют номера по узлу. Контекстная диаграмма всегда имеет номер **A-0**, декомпозиция контекстной диаграммы — номер **A0**, остальные диаграммы декомпозиции — номера по соответствующему узлу (например, **A1**, **A2**, **A213** и т. д.). ВРwin автоматически поддерживает нумерацию по узлам, т. е. при проведении декомпозиции создается новая диаграмма и ей автоматически присваивается соответствующий номер.

Чтобы отличать различные версии одной и той же диаграммы, используется специальный номер — **C-number**, который должен присваиваться автором модели вручную. **C-number** — это произвольная строка, но рекомендуется придерживаться стандарта, когда номер состоит из буквенного префикса и порядкового номера, причем, в качестве префикса используются инициалы автора диаграммы, а порядковый номер отслеживается автором вручную, например **GVI021**.

Если активность не повергалась декомпозиции, то левый верхний угол прямоугольника активности автоматически перечеркивается.

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. Граничные стрелки мигрируют (переносятся) из родительской диаграммы в дочернюю диаграмму. Границы дочерней диаграммы соответствуют границам декомпозируемой активности. Поэтому входные стрелки располагаются на левой границе диаграммы декомпозиции и т. п. Для большего удобства граничные стрелки могут снабжаться так называемыми ICOM-кодами. 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. Граничные стрелки необходимо соединить с соответствующими входами, выходами и т. п. активностей диаграммы декомпозиции.

Стрелки, соединяющие активности на диаграмме, называются *внутренними*. В IDEF0 различают *пять типов связей работ*.

Связь по входу (output-input), когда стрелка выхода вышестоящей работы (далее — просто выход) направляется на вход нижестоящей работы (рис. 1.2). На рис. 1.5-1.6 в основном показаны только рассматриваемые связи.

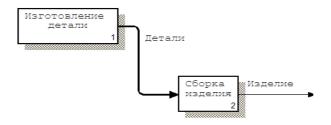


Рис. 1.2. Связь по входу

Связь по управлению (output-control), когда выход вышестоящей работы направляется на управление нижестоящей (рис. 1.3). Связь по управлению показывает доминирование вышестоящей работы.

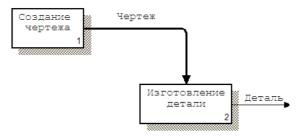


Рис. 1.3. Связь по управлению

Обратная связь по входу (output-input feedback), когда выход нижестоящей работы направляется на вход вышестоящей (рис. 1.4). Такая связь, как правило, используется для описания циклов.

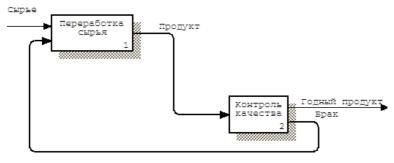


Рис. 1.4. Обратная связь по входу

Обратная связь по управлению (output-control feedback), когда выход нижестоящей работы направляется на управление вышестоящей (рис. 1.5). Обратная связь по управлению часто свидетельствует об эффективном управлении бизнес — процессами.

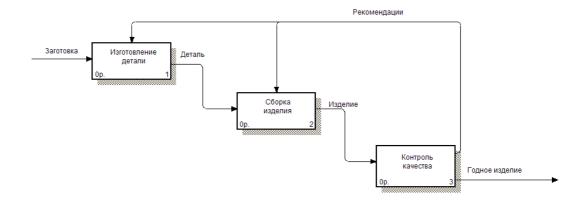


Рис. 1.5. Обратная связь по управлению

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 (рис. 1.6).

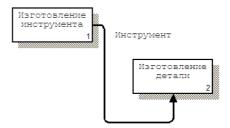


Рис. 1.6. Связь выход-механизм

Простейшим и наиболее распространенным видом стрелок является явная стрелка, которая имеет источником одну-единственную активность и назначением тоже одну-единственную активность. Одни и те же данные или объекты, порожденные одной активностью, могут использоваться сразу в нескольких других активностях. С другой стороны, стрелки, порожденные в разных активностях, 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. Смысл разветвляющихся и сливающихся стрелок передается именованием 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 тоже 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. 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 чего вызвать редактор имени и присвоить имя стрелке. Это имя будет соответствовать только выделенной ветви.

Иногда отдельные интерфейсные дуги высшего уровня не имеет смысла продолжать рассматривать на диаграммах нижнего уровня, или наоборот – отдельные дуги нижнего уровня отражать на диаграммах более высоких уровней — это будет только перегружать диаграммы и делать их сложными для восприятия. Для решения подобных задач в стандарте IDEF0 предусмотрено понятие *туннелирования*.

Вновь созданные на диаграмме декомпозиции граничные стрелки изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня (рис. 1.7).

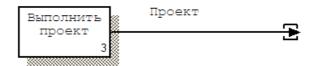


Рис. 1.7. Неразрешенная (unresolved) стрелка

Можно разрешить миграцию новой стрелки на диаграмму верхнего уровня или не разрешить такую миграцию. В последнем случае говорят, что стрелка будет туннелирована. В ВРwin для этого нужно щелкнуть правой кнопкой мыши по квадратным скобкам граничной стрелки и в контекстном меню выбрать команду Arrow Tunnel. Появляется 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 туннелирована и не попадет на другую диаграмму. Туннельная стрелка изображается с круглыми скобками на конце.

Туннелирование может быть применено для изображения малозначимых стрелок. Если на какой-либо диаграмме нижнего уровня необходимо изобразить малозначимые данные или объекты, которые нецелесообразно отображать на диаграммах вышестоящего уровня, то следует туннелировать стрелки на самом нижнем уровне. Такое туннелирование называется туннель "нев-родительской-диаграмме". Другим примером туннелирования может быть ситуация, когда стрелка механизма мигрирует с верхнего уровня на нижний, причем на нижнем уровне этот механизм используется одинаково во всех работах без исключения. В этом случае стрелка механизма на нижнем уровне может быть удалена, после чего на родительской диаграмме она может быть туннелирована, острие стрелки на родительской диаграмме будет изображено в круглых скобках. В комментарии к стрелке или в словаре можно указать, что механизм будет использоваться во всех работах дочерней диаграммы де-

композиции. Такое туннелирование называется туннель "не-в-дочерней-диаграмме".

Стандарт IDEF0 содержит набор процедур, позволяющих разрабатывать и согласовывать модель большой *группой людей*, принадлежащих к разным областям деятельности моделируемой системы. Обычно процесс разработки является итеративным и состоит из следующих условных этапов:

- 1. Создание модели группой специалистов, относящихся к различным сферам деятельности предприятия. Эта группа в терминах IDEF0 называется авторами (**Authors**). Построение первоначальной модели является динамическим процессом, в течение которого авторы опрашивают компетентных лиц о структуре различных процессов, создавая модели деятельности подразделений. При этом их интересуют ответы на следующие вопросы:
 - Что поступает в подразделение "на входе"?
 - Какие функции и в какой последовательности выполняются в рамках подразделения?
 - Кто является ответственным за выполнение каждой из функций?
 - Чем руководствуется исполнитель при выполнении каждой из функций?
 - Что является результатом работы подразделения (на выходе)?
- 2. На основе имеющихся положений, документов и результатов опросов создается черновик (Model Draft) модели.
- 3. Распространение черновика для рассмотрения, согласований и комментариев. На этой стадии происходит обсуждение черновика модели с широким кругом компетентных лиц (в терминах IDEF0 читателей) на предприятии. При этом каждая из диаграмм черновой модели письменно критикуется и комментируется, а затем передается автору. Автор, в свою очередь, также письменно соглашается с критикой или отвергает ее с изложением логики принятия решения и вновь возвращает откорректированный черновик для дальнейшего рассмотрения. Этот цикл продолжается до тех пор, пока авторы и читатели не придут к единому мнению.
- 4. Официальное утверждение модели. Утверждение согласованной модели происходит руководителем рабочей группы в том случае, если у авторов модели и читателей отсутствуют разногласия по поводу ее адекватности. Окончательная модель представляет собой согласованное представление о предприятии (системе) с заданной точки зрения и для заданной цели.

Наглядность графического языка IDEF0 делает модель вполне читаемой и для лиц, которые не принимали участия в проекте ее создания, а также эффективной для проведения показов и презентаций. В дальнейшем на базе построенной модели могут быть организованы новые проекты, нацеленные на производство изменений в модели.

1.2. Методология DFD

Целью методологии является построение модели рассматриваемой системы в виде диаграммы потоков данных (Data Flow Diagram – DFD). Диаграммы потоков данных предназначены прежде всего для описания документооборота и обработки информации, хотя допускают и представление других объектов.

При создании диаграммы потоков данных используются *четыре основных понятия*:

- потоки данных,
- процессы (работы) преобразования входных потоков данных в выходные,
 - внешние сущности,
 - накопители данных (хранилища).

Потоки данных являются абстракциями, использующимися для моделирования передачи информации (или физических компонент) из одной части системы в другую. Потоки на диаграммах изображаются именованными стрелками, ориентация которых указывает направление движения информации.

Процессы (работы) служат для преобразования входных потоков данных в выходные. Имя процесса должно содержать глагол в неопределенной форме с последующим дополнением (например, «получить документы по отгрузке продукции»). Каждый процесс имеет уникальный номер для ссылок на него внутри диаграммы, который может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

Хранилище (накопитель) данных моделирует данные, которые будут сохраняться в памяти между процессами. Информация, которую содержит хранилище, может использоваться в любое время после ее получения, при этом данные могут выбираться в любом порядке. Имя хранилища должно определять его содержимое и быть существительным.

Внешняя сущность представляет собой материальный объект вне контекста системы, являющейся источником или приемником данных. Ее имя должно содержать существительное, например, «склад товаров». Предполагается, что объекты, представленные как внешние сущности, не должны участвовать ни в какой обработке.

Кроме основных элементов, в состав DFD входят словари данных и миниспецификации.

Словари данных являются каталогами всех элементов данных, присутствующих в DFD, включая потоки данных, хранилища и процессы, а также все их атрибуты. Миниспецификации обработки — описывают DFD-процессы нижнего уровня. Фактически миниспецификации представляют собой алгоритмы описания задач, выполняемых процессами: множество всех миниспецификаций является полной спецификацией системы.

Диаграммы потоков данных строятся в виде иерархии. Сначала строится контекстная диаграмма. При проектировании относительно простых систем строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы. Перед построением контекстной DFD необходимо проанализировать внешние события (внешние сущности), оказывающие влияние на функционирование системы. Количество потоков на контекстной диаграмме должно быть по возможности небольшим, поскольку каждый из них может быть в дальнейшем разбит на несколько потоков на следующих уровнях диаграммы.

Для сложных систем (признаками сложности могут быть наличие большого количества внешних сущностей (десять и более), распределенная природа системы или ее многофункциональность) строится *иерархия контекстных диаграмм*. При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор *подсистем*, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

Для проверки контекстной диаграммы можно составить *список собы- тий*. Список событий должен состоять из описаний действий внешних сущностей (событий) и соответствующих реакций системы на события. Каждое событие должно соответствовать одному или более потокам данных: входные потоки интерпретируются как воздействия, а выходные потоки – как реакции системы на входные потоки.

Каждый процесс на DFD, в свою очередь, может быть детализирован при помощи DFD или (если процесс элементарный) спецификации. Спецификация процесса должна формулировать его основные функции таким образом, чтобы в дальнейшем специалист, выполняющий реализацию проекта, смог выполнить их или разработать соответствующую программу. Спецификация является конечной вершиной иерархии DFD. Решение о завершении детализации процесса и использовании спецификации принимается аналитиком исходя из следующих критериев:

- наличия у процесса относительно небольшого количества входных и выходных потоков данных (2-3 потока);
- возможности описания преобразования данных процессов в виде последовательного алгоритма;
- выполнения процессом единственной логической функции преобразования входной информации в выходную;
- возможности описания логики процесса при помощи спецификации небольшого объема (не более 20-30 строк).
- В качестве языка спецификаций обычно используются структурированный естественный язык или псевдокод.
- В методологии DFD используются две *нотации*: Йодана-Де Марко (Yourdan) и Гейна-Сарсона (Gane-Sarson) табл. 1.1.

Таблица 1.1. Элементы методологии DFD в нотациях Гейна-Сарсона и Йодана-Де Марко

Элемент	Описание	Нотация Йордона- Де Марко	Нотация Гейна- Сарсона
Функция	Работа.	Имя функции Номер	Имя функции Номер
Поток данных	Объект, над которым выполняется работа. Может быть логическим ил и управляющим. (Управляющие потоки обозначаются пунктирной линией со стренкой).	Имя объекта	Имя объекта (Понятие управляющего потока отсутствует)
Хранилище данных	Структура для хранения информационных объектов.	Имя объекта	Имя объекта
Внешняя сущность	Внешний по отношению к системе объект, обмениваю щийся с нею потоками.	Имя внешнего объекта	Имя внешнего объекта

Следует отметить, что в BPwin формально используется нотация Гейна-Сарсона, но с рядом отступлений: отсутствуют миниспецификации, отличается изображение функций, контекстная диаграмма не может содержать подсистемы. Сравнивая методологии DFD и IDEF0, можно отметить, что в методологии DFD, кроме расширения изобразительных возможностей диаграмм (за счет хранилищ данных), изменяются правила интерфейсов для стрелок: стрелки могут входить и выходить с любой стороны блока.

1.3. Методология IDEF3

IDEF3 является стандартом документирования *технологических процессов*, происходящих на предприятии, и предоставляет инструментарий для наглядного исследования и моделирования их сценариев [7 – 9]. *Сценарием (Scenario)* называется описание последовательности изменений свойств объекта в рамках рассматриваемого процесса (например, описание последовательности этапов обработки детали в цехе и изменение её свойств после прохождения каждого этапа). Исполнение каждого сценария сопровождается соответствующим документооборотом, который состоит из двух основных потоков: документов, определяющих структуру и последовательность процесса (технологические карты, стандарты и т.д.), и документов, отображающих ход его выполнения (результаты тестов и экспертиз, отчеты о браке, и т.д.). Для эффективного управления любым процессом, необходимо иметь детальное представление об его сценарии и структуре сопутствующего документооборота.

Средства документирования и моделирования IDEF3 позволяют выполнять следующие задачи:

- 1. Документировать данные о технологии процесса.
- 2. Определять и анализировать точки влияния потоков сопутствующего документооборота на сценарий технологических процессов.

- 3. Определять ситуации, в которых требуется принятие решения, влияющего на жизненный цикл процесса, например, изменение конструктивных, технологических или эксплуатационных свойств конечного продукта.
- 4. Содействовать принятию оптимальных решений при реорганизации технологических процессов.
- 5. Разрабатывать имитационные модели технологических процессов, по принципу "КАК БУДЕТ, ЕСЛИ..." Такая возможность существует при использовании, например, системы имитационного моделирования ARENA.

Существуют два типа диаграмм в стандарте IDEF3, представляющих описание одного и того же сценария технологического процесса в разных ракурсах. Диаграммы, относящиеся к первому типу, называются диаграммами потокового описания процесса (Process Flow Description Diagrams, PFDD), а ко второму – диаграммами сети изменения состояний объектов (Object State Transition Network, OSTN).

Рассмотрим *пример* [9, 10]. Предположим, требуется описать процесс окраски детали в производственном цехе на предприятии. С помощью диаграмм PFDD документируется последовательность и описание стадий обработки детали в рамках исследуемого технологического процесса. Диаграммы OSTN используются для иллюстрации трансформаций детали, которые происходят на каждой стадии обработки. На следующем примере опишем, как графические средства IDEF3 позволяют документировать вышеуказанный производственный процесс окраски детали. В целом, этот процесс состоит непосредственно из самой окраски, производимой на специальном оборудовании и этапа контроля ее качества, который определяет, нужно ли деталь окрасить заново (в случае несоответствия стандартам и выявления брака) или отправить ее в дальнейшую обработку.

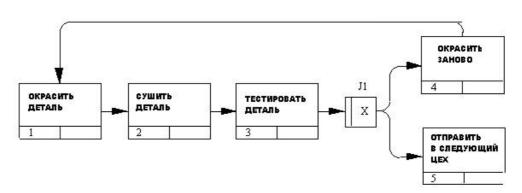


Рисунок 1.8. Пример PFDD диаграммы

На рис. 1.8 изображена диаграмма PFDD, являющаяся графическим отображением сценария обработки детали. Прямоугольники на диаграмме PFDD называются функциональными элементами или элементами поведения (Unit of Behavior, UOB) и обозначают событие, стадию процесса или принятие решения. Каждый UOB имеет свое имя, отображаемое в глагольном наклоне-

_

 $^{^{1}}$ В BPwin используется термин "Единица работы" (Unit of Work – UOW).

нии, и уникальный номер. Этот номер не используется вновь даже в том случае, если в процессе построения модели действие удаляется. В диаграммах IDEF3 номер действия, полученного в результате декомпозиции, обычно предваряется номером его родителя (рис. 1.9).



Рис. 1.9. Изображение и нумерация действия в диаграмме IDEF3

Существенные взаимоотношения между действиями изображаются с помощью *связей*. Все связи в IDEF3 являются однонаправленными, и хотя стрелка может начинаться или заканчиваться на любой стороне блока, обозначающего действие, диаграммы IDEF3 обычно организуются слева направо таким образом, что стрелки начинаются на правой и заканчиваются на левой стороне блоков. В табл. 1.2 приведены три возможных типа связей. Стандарт предусматривает и другие типы стрелок [8, 11], но они малоприменимы и не поддерживаются CASE-системами.

Таблица 1.2. Типы связей IDEF3

Изображение	Название	Назначение
	*	Исходное действие должно завершиться, прежде чем конечное действие сможет начаться
	Объектный поток (Object flow)	Выход исходного действия является входом конечного действия (исходное действие должно завершиться, прежде чем конечное действие сможет начаться)
	Нечеткое отношение (Relationship)	Вид взаимодействия между исходным и конечным действиями задается аналитиком отдельно для каждого случая использования такого отношения

Связь типа "временное предшествование" показывает, что исходное действие должно полностью завершиться, прежде чем начнется выполнение конечного действия.

Связь типа "объектный поток" используется в том случае, когда некоторый объект, являющийся результатом выполнения исходного действия, необходим для выполнения конечного действия. Обозначение такой связи отличается от связи временного предшествования двойной стрелкой. Наименования потоковых связей должны четко идентифицировать объект, который передается с их помощью. Временная семантика объектных связей аналогична связям предшествования, это означает, что порождающее объектную связь

исходное действие должно завершиться, прежде чем конечное действие может начать выполняться.

Связь типа "нечеткое отношение" используется для выделения отношений между действиями, которые невозможно описать с использованием связей предшествования или объектных связей. Обычно эти связи указывают, что между объектам существуют некоторые отношения, но на момент описания процесса они не определены.

Объект, обозначенный на рис. 1.8 как J1, называется перекрестиком (Junction). Перекрестки используются для отображения логики взаимодействия стрелок (потоков) при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы. Различают перекрестки для слияния (Fan-in Junction) и перекрестки для разветвления (Fan-out Junction) стрелок. Перекресток не может использоваться одновременно для слияния и для разветвления. При внесении перекрестка в диаграмму необходимо указать тип перекрестка. Классификация возможных типов перекрестков приведена в табл. 1.3.

Таблица 1.3. Типы перекрестков

Обозначение	Наименование	Смысл в случае слияния стрелок (Fan-in Junction)	Смысл в случае разветвления стрелок (Fan-out Junction)
&	Asynchronous AND	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
&	Synchronous AND	Все предшествующие процессы должны быть завершены одновременно	Все следующие процессы запускаются одновременно
0	Asynchronous OR	Один или несколько пред- шествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
0	Synchronous OR	Один или несколько пред- шествующих процессов за- вершаются одновременно	Один или несколько следующих процессов запускаются одновременно
X	XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Все перекрестки в PFDD диаграмме нумеруются, каждый номер имеет префикс "J".

Сценарий, отображаемый на диаграмме, можно описать в следующем виде. Деталь поступает в окрасочный цех, подготовленной к окраске. В процессе окраски наносится один слой эмали при высокой температуре. После этого производится сушка детали, после которой начинается этап проверки качества нанесенного слоя. Если тест подтверждает недостаточное качество нанесенного слоя (недостаточную толщину, неоднородность и т.д.), то деталь заново пропускается через цех окраски. Если деталь успешно проходит кон-

троль качества, то она отправляется в следующий цех для дальнейшей обработки.

Описания процессов могут состоять из нескольких сценариев и содержать как диаграммы PFDD, так и OSTN. Для обозначения отношений и связей между UOB различных уровней PFDD и OSTN диаграмм и разных сценариев в IDEF3 используются специальные ccылкu (Referents).

Ссылки могут использоваться:

- для обращения к ранее определенному функциональному модулю *UOB* без повторения его описания;
- для передачи управления или индикации наличия циклических действий при выполнении процесса;
- организации связи между диаграммами описания процесса PFDD и OSTN диаграммами.

Соответственно, выделяют следующие типы ссылок:

GOTO — циклический переход (в повторяющейся последовательности UOW), возможно, на текущей диаграмме, но не обязательно. Если все UOW цикла присутствуют на текущей диаграмме, цикл может также изображаться стрелкой, возвращающейся на стартовую UOW. GOTO может ссылаться на перекресток.

UOB — экземпляр другого, ранее определенного UOB, выполняется в определенной точке. Например, UOB "Контроль качества" может быть использован в процессе "Изготовление редуктора" несколько раз, после каждой единичной операции.

SCENARIO – название сценария. Эта ссылка означает, что должна быть произведена активизация всех декомпозиций указанного сценария.

TS (*Transition Schematic*) – переход на схему. Это ссылка на соответствующую диаграмму, т. е. процесс, на который ссылаются, должен быть инициализирован.

NOTE (примечание) используется для документирования информации, относящейся к каким-либо графическим объектам на диаграмме. Элемент «примечание» может использоваться как в диаграммах описания процесса, так и объектных диаграммах OSTN. Этот элемент может быть применен к функциональному элементу UOW, перекрестку, связи, объекту или ссылке.

Отметим, что в BPwin используются немного другие ссылки [12].

Методология IDEF3 определяет два вида ссылок по способу запуска. Ссылка "Вызвать и продолжить" (Call and Continue Referent) указывает, что элемент, указанный в ссылке, должен быть активизирован до завершения выполнения действия модулем, к которому относится ссылка. Ссылка "Вызвать и ждать" (Call and Wait Referent), указывает, что элемент, указанный в ссылке, должен начать и закончить выполнение действия до завершения действия модулем, к которому относится ссылка.

Графические обозначения ссылок приведены на рис. 1.10.

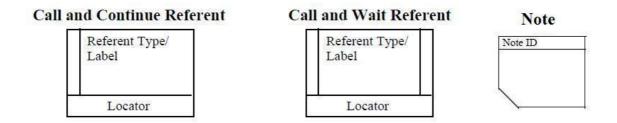


Рис. 1.10. Графическое обозначение ссылок

В основном поле символа ссылки указывается её тип (*Referent Type*) "*UOB*", "*SCENARIO*", "*TS*" или "*GOTO*" и через дробь "*Label*" – уникальное наименование блока, сценария, схемы или функции узла, на который указывает ссылка. В поле "Locator" указывается уникальный идентификатор элемента, указанного в ссылке. Пример использования ссылок показан на рис. 1.11 [8].

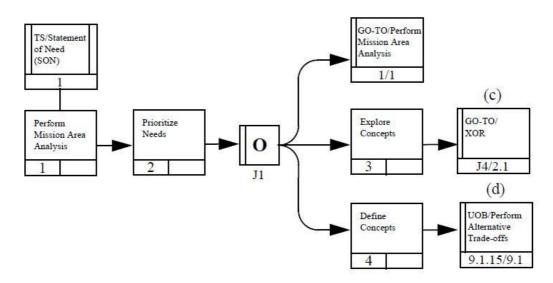


Рис. 1.11. Примеры использования ссылок

Каждый функциональный блок UOB может иметь последовательность декомпозиций, и, следовательно, может быть детализирован с любой необходимой точностью. Под декомпозицией мы понимаем представление каждого UOB с помощью отдельной IDEF3 диаграммы. Например, мы можем декомпозировать UOB "Окрасить Деталь", представив его отдельным процессом и построив для него свою PFDD диаграмму. При этом эта диаграмма будет называться дочерней, по отношению к изображенной на рис. 1.8, а та, соответственно, родительской. Номера UOB дочерних диаграмм имеют сквозную нумерацию, т.е., если родительский UOB имеет номер "1", то блоки UOB на его декомпозиции будут соответственно иметь номера "1.1", "1.2" и т.д. Применение принципа декомпозиции в IDEF3 позволяет структурировано описывать процессы с любым требуемым уровнем детализации. На рис. 1.12 приведен пример декомпозиции модулей (UOB) и принцип формирования их

номеров. Для наглядности все модули представлены на одном рисунке, но в IDEF3 они отображаются в трех диаграммах.

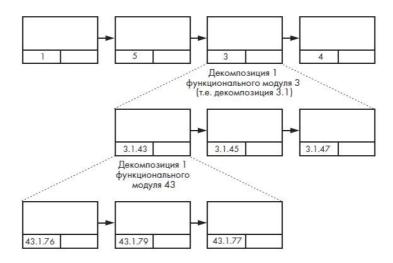


Рис. 1.12. Декомпозиция функциональных блоков

Методология IDEF3 позволяет декомпозировать работу многократно, т. е. работа может иметь множество дочерних работ. Возможность множественной декомпозиции отражается в нумерации работ: номер работы состоит из номера родительской работы, номера декомпозиции и номера работы на текущей диаграмме. На рис. 1.13 представлен пример двух вариантов декомпозиции родительского модуля [8].

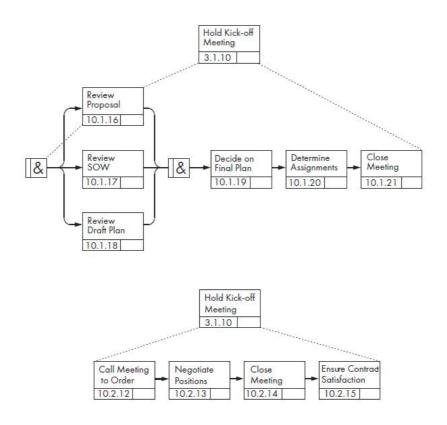


Рис. 1.13. Пример двух вариантов декомпозиции модуля

Если диаграммы PFDD описывают технологический процесс "с точки зрения наблюдателя", то другой класс диаграмм *OSTN* – диаграммы сети изменения состояний объектов (не поддерживаются в BPwin) позволяет рассматривать тот же самый процесс "с точки зрения объекта". С ее помощью можно графически представить, как одни виды объектов преобразуются в другие или изменяют свое состояние в ходе выполнения рассматриваемого процесса.

На OSTN состояния объектов изображаются окружностями с именем объекта внутри, а изменения состояний — соединительными линиями. Состояние объекта описывается фактами и ограничениями, которые должны выполняться, чтобы объект находился в данном состоянии. Требования для перехода объекта в заданное состояние определяются условиями входа. Условия выхода говорят о ситуации, в которой объект выходит из заданного состояния. Эти ограничения описываются в списке свойств. Связи переходов состояний задают возможные способы изменения состояний объектов.

Для изображения последовательностей переходов объектов из одного вида в другой и изображения перехода одного и того же объекта из одного состояния в другое в диаграммах OSTN используются связи переходов (Transition Links), которые бывают слабыми (Weak Transition Link) и сильными (Strong Transition Link). Слабые связи переходов изображаются сплошными одинарными стрелками (рис. 1.14) и показывают, что объекту вида В предшествует объект вида А или что состоянию В некоторого объекта предшествует его состояние А.

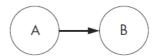


Рис. 1.14. Пример слабой связи переходов

Сильные связи переходов изображаются двойными однонаправленными стрелками (рис. 1.15) и подчеркивают, что объекту вида В должен предшествовать объект вида А или что состояние В объекта достижимо только из состояния А.

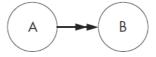


Рис. 1.15. Пример сильной связи переходов

В диаграммах *OSTN* используются те же виды *ссылок*, что и в диаграммах *PFDD*. Исключение составляет лишь ссылка типа *GOTO*, которая используется только в диаграммах потоковых процессов *PFDD*. Ссылки могут относиться как к символу объекта, так и к связи перехода. Соответственно, они интерпретируются как действия, которые необходимо осуществлять для поддержания объекта в данном виде или состоянии, или как действия, которые необходимы для преобразования вида или состояния объекта. Так как

процессы поддержания объекта в определенном состоянии и его преобразования могут быть сложными, то допускается использование нескольких ссылок к любому элементу OSTN диаграммы.

На диаграммах *OSTN* могут использоваться перекрестки. Перекресток изображается кружком, внутри которого содержится условное обозначение логической функции, реализуемой перекрестком В качестве логических функций могут использоваться И (&), ИЛИ (O) и ИСКЛЮЧАЮЩЕЕ ИЛИ (X). Как и на диаграммах *PFDD*, узлы перехода могут означать слияние и разветвление. Но на диаграммах *OSTN* перекрестки не делятся на асинхронные и синхронные. На рис. 1.16 показан пример использования узла разветвления с логической функцией ИЛИ.

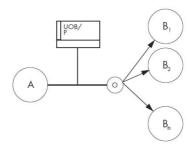


Рис. 1.16. Пример перекрестка с логической функцией ИЛИ

Диаграмма рис. 1.16 означает, что под действиями UOB с именем P объект из состояния A может перейти в одно или сразу несколько состояний из множества возможных: B_1 , B_2 , ..., B_n . Если бы в качестве логической функции использовалась функция ИСКЛЮЧАЮЩЕЕ ИЛИ, то это говорило бы, что возможен переход только в одно из возможных состояний B_1 , B_2 , ..., B_n . Использование же функции И в перекрестке отображало бы переход объекта из состояния A сразу во все состояния B_1 , B_2 , ..., B_n .

На рис.1.17 представлено отображение процесса окраски с точки зрения OSTN диаграммы.

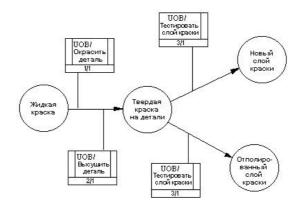


Рис. 1.17. Пример OSTN диаграммы

BPwin имеет возможность преобразования диаграмм IDEF3 в имитационную модель популярной системы моделирования Arena [13 – 14]. Идея преобразования описана в [2, 3], подробное описание дано в фирменной документации [12].

2. СОЗДАНИЕ МОДЕЛИ В СТАНДАРТЕ IDEF0

2.1. Создание контекстной диаграммы

Создание модели рассмотрим на следующем примере [18]. Компания занимается сборкой и продажей настольных компьютеров и ноутбуков. Основными процедурами в компании являются следующие:

- прием продавцами заказов клиентов;
- группировка операторами заказов по типам компьютеров;
- сборка и тестирование операторами компьютеров (под операторами понимаем технический персонал);
 - упаковка операторами компьютеров;
 - отгрузка кладовщиком заказов клиентам.

Компания использует систему компьютерной бухгалтерии, которая позволяет оформить заказ, счет и отследить платежи по счетам.

Запускаем программу: заходим в Пуск>Все программы>CA>ERwin>ERwin Process Modeler r7.3>Process Modeler.

Начинаем создание модели [18, 19]. Описание и скриншоты приведены для ERwin Process Modeler Release 7.3.3.1773. После запуска появляется диалог **I would like to** – Я хочу...(рис. 2.1). Если диалог не появился (он может быть отключен), выберите пункт **File>New** или кнопку создания нового файла.

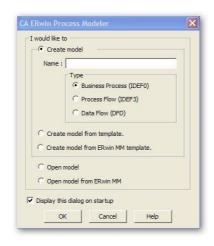


Рис. 2.1. Диалог I would like to

В этом окне нам предлагается выбрать, с чего мы хотим начать работу:

- -Create model from template создать новую модель по шаблону;
 - − Open model открыть существующую модель
- -Open model from ERwin MM; Create model from ERwin template открыть или создать модель из среды групповой разработки CA ERwin Model Manager.

Выберем создание новой модели по шаблону в методологии IDEF0. Введем имя модели: "Деятельность компании" и нажмем кнопку **ОК**.

Далее выводится диалог свойств модели — Properties for New Models (рис. 2.2).

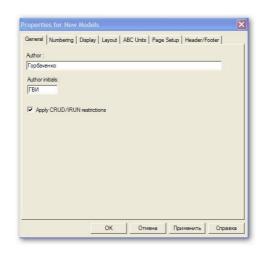


Рис. 2.2. Диалог свойств модели

На вкладке General вводим имя и инициалы автора. Включим опцию Ap-Restrictions _ ply CRUD/IRUN Применить ограничения **CRUD/IRUN**, хотя для нашей задачи эта опция не важна, так как мы не создаем модели данных для нашего процесса. Эти ограничения применяются для связывания модели процессов и модели данных [18]. Данные не могут использоваться работами произвольно. Например, входные данные, представленные стрелкой входа, могут только преобразовываться в выход или потребляться. Рассматриваемые ограничения контролируют использование Аббревиатура CRUD означает Create, Retrieve, Update, Delete и применяется для сущностей, a IRUN означает Insert, Retrieve, Update, Nullify (сделать неопределенным) и применяется для атрибутов сущностей.

На вкладке **Numbering** задаются опции нумерации элементов модели (рис. 2.3):

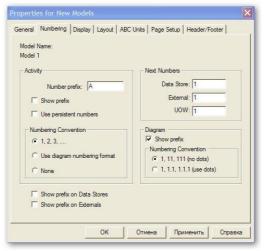


Рис. 2.3. Опции нумерации

Activity — группа параметров, отвечающая за нумерацию функциональных блоков:

- **Number prefix** символ, предшествующий номеру блока (по умолчанию "A");
 - **Show prefix** отображение префикса;
- Use persistent numbers использование постоянной нумерации для диаграмм IDEF0 (если опция включена, то в случае перемещения блока нумерация будет сохранена, если же опция отключена, то нумерация будет автоматически изменена в соответствии со стандартом IDEF0). На диаграммах IDEF3 и DFD используется только постоянная нумерация;

Numbering convention — задание параметров нумерации:

- -1, 2, 3, ... задание последовательной нумерации;
- Use diagram numbering format нумерация каждой функции содержит в себе уровень декомпозиции.
 - **None** скрытие нумерации функциональных блоков;

Next Numbers — зона, содержащая опции, отвечающие за значения, с которых начнется нумерация:

- **Data Store** задание первого значения для хранилищ данных на DFD-диаграммах;
- **External** задание первого значения для внешней ссылки на DFD-диаграммах;
 - **UOW** задание первого значения для UOW на IDEF3-диаграммах.

Оставим на вкладке все без изменения.

На вкладке **Display** (рис. 2.4) определяется, что будет отображаться на диаграммах.

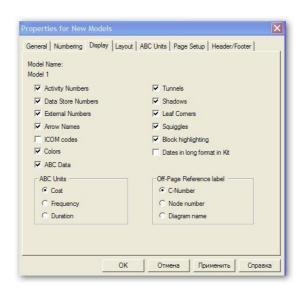


Рис. 2.4. Вкладка Display

Activity Numbers — отображение нумерации функциональных блоков; Data Store Numbers — отображение нумерации хранилищ данных в нотации DFD; External Numbers — отображение нумерации внешних сущностей;

Arrow Names – отображение названий стрелок;

ICOM codes – отображение ICOM-кодов. 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;

Colors – отображение цветов;

ABC Data – отображение данных стоимостного анализа;

Tunnels – отображение квадратных и круглых скобок на стрелках при обозначении незатуннелированных и затуннелированных стрелок соответственно;

Shadows – отображение теней объектов;

Leaf Corners — отображение риски на левом верхнем углу блока, говорящей об отсутствии декомпозиции соответствующей функции.

Squiggles – отображение сноски на названия стрелок;

Block highlighting – отображение подсветки выделенного блока;

Dates in long format in Kit – отображение даты в длинном формате (короткий формат – 10/1/04, длинный формат – October 1, 2004).

Группа переключателей **ABC Units** отвечает за содержимое поля стоимостного анализа, а именно: **Cost** — стоимость, **Frequency** — частота, **Duration** — длительность.

В группе Off-Page Reference label указывается маркер межстраничной ссылки. В качестве маркера может выступать пользовательский номер С-number-диаграммы (C-number), номер диаграммы по узлу (Node number) и имя диаграммы (Diagram name).

И на этой вкладке оставим все по умолчанию.

Параметры вкладки **Layout – схема размещения** (рис. 2.5) отвечают за размещение объектов на диаграмме.

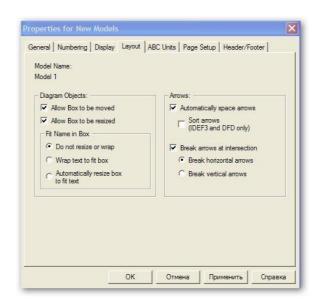


Рис. 2.5. Вкладка **Layout**

Группа **Diagram Objects** отвечает за объекты диаграмм:

- Allow Box to be moved возможность перемещения объектов диаграмм вручную;
- -Allow Box to be resized возможность изменения размеров объектов диаграмм вручную;

Подгруппа **Fit Name in Box** позволяет задать опции размещения текста в блоке, а именно:

- Do not resize or wrap текст будет вписан в блок без учета размеров;
 - Wrap text to fit box текст будет подогнан по размеру блока;
- Automatically resize box to fit text— размер блока будет подогнан по тексту;

Подгруппа **Arrows** отвечает за размещение стрелок:

- -Automatically space arrows автоматическое размещение стрелок на диаграмме (например, при задании новой стрелки или изменении размера функционального блока, стрелки на границах блока будут автоматически размещены на одинаковом расстоянии друг от друга);
- Sort arrows минимизация пересечений стрелок на диаграммах IDEF3 и DFD;
- -Break arrows at intersection эта опция позволяет создать разрывы стрелок, что облегчает восприятие диаграмм. Можно задать разрыв горизонтальных (Break horizontal arrow) и вертикальных (Break vertical arrow) стрелок.

Зададим опции, как показано на рис. 2.5.

Вкладка **ABC Units** (рис. 2.6) отвечает за содержание единиц стоимостного анализа.

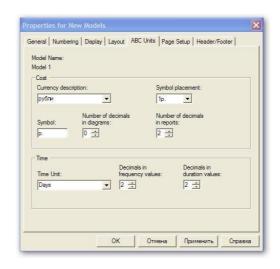


Рис. 2.6. Вкладка **ABC** Units

Cost – зона задания параметров отображения стоимости:

- **Currency description** - определение валюты. Рублей в выпадающем списке нет, но можно просто вписать нужную валюту;

- Symbol placement определение положения знака валюты относительно числа;
- **Symbol** определение знака валюты, по умолчанию берется из настроек Windows, можно вписать новое обозначение;
- Number of decimals in diagrams определение числа десятичных знаков при отображении на диаграмме;
- Number of decimals in reports определение числа десятичных знаков при составлении отчетов.

Time – зона задания параметров отображения времени:

- **Time Unit** определение единиц измерения времени;
- Decimals in frequency values определение количества знаков при задании частоты;
- Decimals in duration values определение количества знаков при задании длительности;

Зададим значения, как показано на рис. 2.6.

На вкладке **Page Setup** (рис. 2.7) задаются опции просмотра и печати страницы.

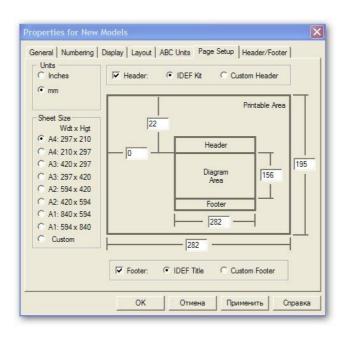


Рис. 2.7. Вкладка **Page** Setup

Units – определение единиц измерения (дюймы или миллиметры);

Sheet size – определение размеров листа;

Header — определения способа заполнения заголовка диаграммы (**IDEF Kit** — в соответствии с методологией IDEF, **Custom Header** — заданный пользователем на вкладке **Header/Footer**);

Footer — определение способа заполнения нижнего колонтитула (IDEF Kit — в соответствии с методологией IDEF, Custom Header — заданный пользователем на вкладке Header/Footer).

Примем значения по умолчанию.

На вкладке **Header/Footer** (рис. 2.8) определяется пользовательский вид заголовка и нижнего колонтитула.

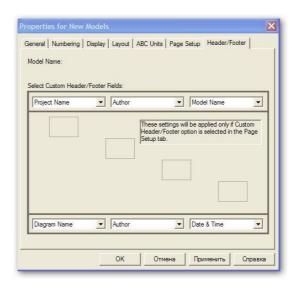


Рис. 2.8. Вкладка Header/Footer

Доступна следующая информация для заполнения полей:

- Author фамилия автора,
- Author Initials инициалы автора,
- **C-Number C-**HOMEP,
- Creation Date дата создания,
- Date and Time дата и время,
- Date (long) дата в полном формате записи,
- Date (short) дата в кратком формате записи,
- **Diagram Name** имя диаграммы,
- Diagram Number номер диаграммы,
- Diagram Status статус диаграммы,
- Model Name имя модели,
- Model Status статус модели,
- Page Number номер страницы,
- **Project Name** имя проекта,
- Revision Date дата пересмотра.

Заполним вкладку как показано на рис. 2.8.

Подтвердим сделанные изменения – нажмем кнопку **ОК**. Сразу создается заготовка для контекстной диаграммы (рис. 2.9).

Граничные рамки диаграммы называются каркасом.

Каркас содержит заголовок (верхняя часть рамки) и подвал (нижняя часть) — табл. 2.1-2.2.

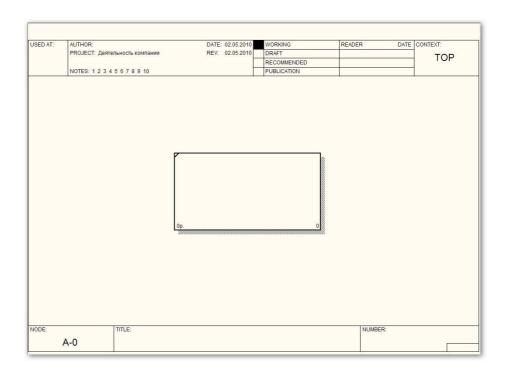


Рис. 2.9. Заготовка контекстной диаграммы

Таблица 2.1. Поля заголовка каркаса (слева направо)

Поле	Смысл		
Used At	Используется для указания на родительскую работу в случае, если на текущую диаграмму ссылались посредством стрелки вызова		
Author, Date, Rev, Project	Имя автора диаграммы, дата создания и имя проекта, в рамках которого была создана диаграмма, последнего редактирования диаграммы		
Notes 123456789 10	Число замечаний. Используется при проведении сеанса экспертизы. Эксперт на бумажной копии диаграммы указывает число замечаний, вычеркивая цифру из списка каждый раз при внесении нового замечания		
Status	Статус отображает стадию создания диаграммы, отображая все этапы публикации Working — новая диаграмма, кардинально обновленная диаграмма или новый автор диаграммы. Draft — диаграмма прошла первичную экспертизу и готова к дальнейшему обсуждению. Recommended — диаграмма и все ее сопровождающие документы прошли экспертизу. Новых изменений не ожидается. Publication — диаграмма готова к окончательной печати и публикации.		
Reader	Имя читателя (эксперта)		
Date	Дата прочтения (экспертизы)		
Context	Схема расположения диаграммы в иерархии диаграмм. Работа, являющаяся родительской, показана темным прямоугольником, остальные — светлым. На контекстной диаграмме (A-0) показана надпись ТОР. В левом нижнем углу показывается номер по узлу родительской диаграммы:		
	A0		

Таблица 2.2. Поля подвала каркаса (слева направо)

Поле	Смысл
Node	Номер узла диаграммы (номер родительской работы)
Title	Имя диаграммы. По умолчанию – имя контекстной работы
Number	C-Number – задаваемый автором уникальный номер версии диаграммы
Page	Номер страницы

Построение модели включает не только построение диаграмм, но и задание свойств модели и ее объектов. Имя заготовке контекстной диаграммы можно задать, выбрав свойства модели (меню Model>Model Properties...), свойства диаграммы — двойной клик мыши на свободном поле диаграммы, или пункт меню Diagram Properties..., или контекстное меню на свободном поле диаграммы. Свойства активности можно задать двойным кликом мыши на активности или из контекстного меню на этой активности. Задание свойств других объектов рассмотрим позднее.

Зададим свойства модели. На вкладке **General** (рис. 2.10) зададим информацию о модели. Временные рамки **Time Frame** примем **AS-IS**. Это означает, что рассматриваются существующие процессы.

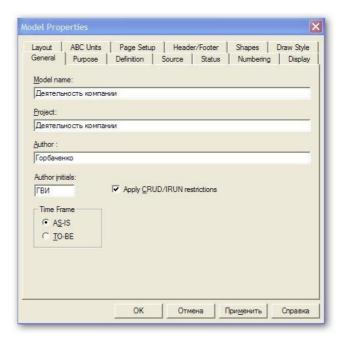


Рис. 2.10. Вкладка General диалога Model Properties

На вкладке **Purpose** (**Цель**) внесем цель моделирования **Purpose**: "Моделировать текущие бизнес—процессы компании" и точку зрения, с которой строится модель **Viewpoint**: "Директор".

На вкладке **Definition (Определение)** задаем определение модели **Definition:** "Учебная модель, описывающая деятельность компании" и границы (рамки) модели **Scope**: "Общее управление бизнесом компании".

На вкладке **Source** (**Источник**) признаемся, что данные взяты из книги [18]. На вкладке **Status** зададим статус всей модели: **Working** — рабочий вариант. На вкладке **Shapes** (**Формы**) задается отображение объектов диаграммы.

На вкладке **Draw Style** — **Стиль рисования** (рис. 2.11) задаются параметры графического отображения.



Рис. 2.11. Вкладка Draw Style

Группа **Style for Diagrams** – задает опции отображения объектов диаграммы:

Methodology Specific – в соответствии с методологией (IDEF0, IDEF3, DFD);

Bitmap – изображения в формате BMP;

Shape – в определенном виде;

Shape and Bitmap - в определенном виде и изображения в формате BMP;

Defer To Diagram — в соответствии с параметрами, заданными на вкладке Draw Style диалогового окна Diagram Properties;

Show Name - отображать имя;

Show Number – отображать номер;

Show ABC Data – отображать данные по стоимостному анализу.

Группа Style for Organization charts — задает опции отображения объектов организационной диаграммы:

Standard – с стандартном виде;

Bitmap – изображения в формате BMP;

Shape – κ a κ форму;

Defer To Diagram — в соответствии с параметрами, заданными на вкладке Draw Style диалогового окна Organization Chart Properties. Остальные вкладки уже были рассмотрены.

Теперь в диалоге **Activity Properties** (рис. 2.12) зададим свойства активности контекстной диаграммы. Вкладки **Name**, **Definition**, **Status**, **Source**, **Box Style** нам уже знакомы. На вкладке **Font** указываются настройки шрифта для отображения надписи на блоке. На вкладке **Color** задаются цветовые настройки.

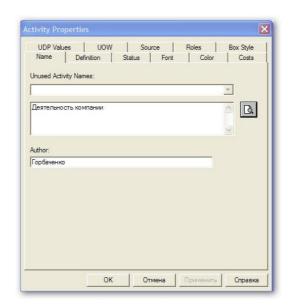


Рис. 2.12. Диалог Activity Properties

На вкладке Cost (рис. 2.13) задаются статьи затрат (Cost Center) для расчета стоимости выполнения данной активности. В верхней части вкладки расположено поле ввода стоимости функции по центрам затрат. В левом столбце представлены доступные центры затрат, их можно определить с помощью кнопки Cost Center Editor или в библиотеке центров затрат. В правом столбце можно ввести стоимость выполнения функции в соответствии с определенным центром затрат. Ниже имеется два переключателя: Override decompositions — не учитывать данные, введенные ниже по декомпозиции (стоимость определяется вручную); Compute from decompositions — вычислить на основе данных, введенных ниже по декомпозиции. Поле Frequency определяет кратность выполнения данной функции в цикле. Поле Duration определяет длительность выполнения функции.

Оставим все значения по умолчанию.

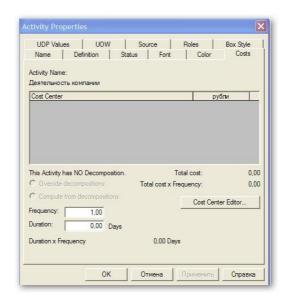


Рис. 2.13. Вкладка Cost

Вкладка **UDP Values** позволяет задавать значения свойств, определенных пользователем. Если тех свойств, которые поддерживает BPwin, недостаточно, то можно воспользоваться возможностью задания произвольных свойств с помощью словаря свойств, определенных пользователем.

Вкладка **UOW** позволяет задать дополнительные свойства активностей. Как правило, эти данные заполняются для функций IDEF3 диаграммы, но могут быть заданы для любого объекта.

На вкладке **Roles** (**Роли**) можно задать роли, которые будут выполнять эту активность. Например, определенную активность выполняет менеджер. Роли нужно задать в словаре ролей, но учтите, что добавить роли можно только в определенную группу ролей, так что сначала нужно задать группы ролей в соответствующем словаре.

Подтвердим сделанные изменения — нажмем кнопку **ОК**. Теперь на контекстной диаграмме изображена активность с заданным именем (рис. 2.14). Кликнув по активности, можно увидеть и редактировать свойства активности.

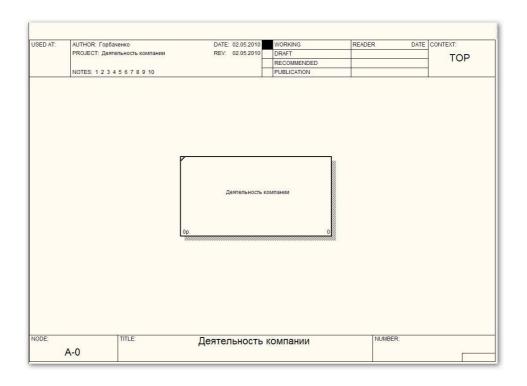


Рис. 2.14. Контекстная диаграмма после задания свойств активности

Дополним контекстную диаграмму стрелками. Чтобы нарисовать стрелку, необходимо выбрать соответствующую кнопку в палитре (рис. 2.15) и первым кликом левой клавиши мыши обозначить начало стрелки, будь-то граница области диаграммы или граница функционального блока, а вторым кликом левой клавиши мыши — конец.



Рис. 2.15. Кнопка рисования стрелок

Нарисуем стрелки в соответствии с рис. 2.16. Чтобы добавить имя стрелке, необходимо дважды кликнуть левой клавишей мыши на стрелке или выбрать пункт **Name** в контекстном меню. В открывшемся окне (рис. 2.17) надо ввести имя стрелки. (Внимание! Никогда не подписывайте стрелки с помощью инструмента **Text**, так как в этом случае информация не попадет в словарь стрелок). После нажатия на кнопку **Применить** появляется окно полного диалога свойств стрелки (рис. 2.18).

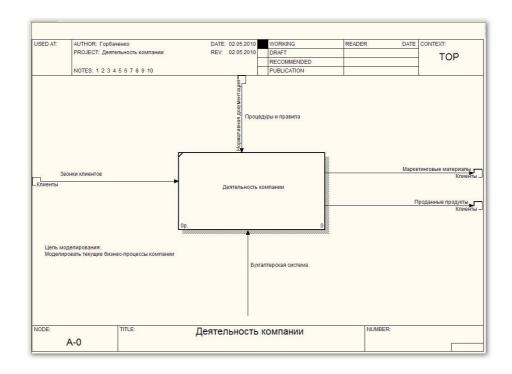


Рис. 2.16. Заполненная контекстная диаграмма

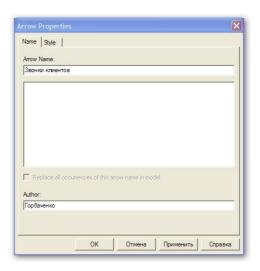


Рис. 2.17. Диалог задания свойств стрелки



Рис. 2.18. Полный диалог задания свойств стрелки

На рис. 2.16 на концах стрелок у границ диаграммы присутствуют квадратики с подписями — это внешние ссылки (External Reference). Для того чтобы добавить внешнюю ссылку, нужно кликнуть правой клавишей мыши на конце стрелки у границы диаграммы и выбрать пункт **External Reference**. Открывается диалоговое окно (рис. 2.19), в котором можно ввести свое название внешней ссылки (оно автоматически будет добавлено в библиотеку), выбрать из уже добавленных или дать ей связь с соответствующей стрелкой.

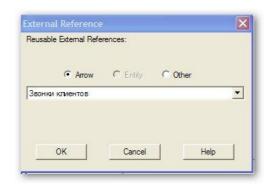


Рис. 2.19. Окно External Reference

Свойства активностей и стрелок автоматически помещаются в соответствующие словари. Для просмотра словаря активностей выберем пункт меню **Dictionary>Activity...** Результат представлен на рис. 2.20.



Рис. 2.20. Словарь активностей

Словарь стрелок можно вывести, выбрав пункт меню **Dictionary>Arrow...** (рис. 2.21).



Рис. 2.21. Словарь стрелок

По модели можно создать отчет, содержащий выбранные пользователем свойства модели. В пункте меню **Tools>Reports>Model Report** вызывается диалог **Model report** (рис. 2.22), в котором отмечаются интересующие свойства (на рис. 2.22 отмечены все возможные свойства). Отчет может быть предварительно просмотрен – **Preview...** (рис. 2.23), выведен на печать (**Print...**) или сохранен как текстовый файл (**Report...**).

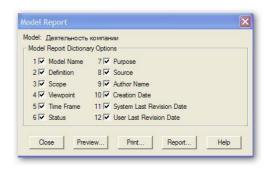


Рис. 2.22. Диалог Model report

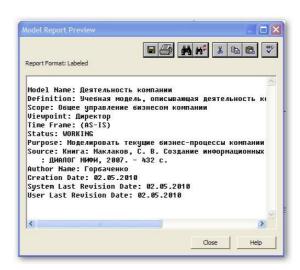


Рис. 2.23. Предварительный просмотр отчета по модели

В левом нижнем углу контекстной диаграммы (рис. 2.16) поместим надпись: «Цель моделирования: Моделировать текущие бизнес — процессы компании». Это необязательный элемент, но надписи помогают сделать модель более наглядной (как и использование цвета). Надписи добавляются с помощью кнопки **Текст** палитры инструментов (рис. 2.24).



Рис. 2.24. Кнопка Текст

Выбираем эту кнопку и кликаем левой клавишей мыши на свободном пространстве диаграммы, куда мы хотим добавить текст. Открывается диалоговое окно добавления текста (рис. 2.25).



Рис. 2.25. Окно добавления текста

С помощью закладок **Font** и **Color** можно задать настройки шрифта и цвета. Внизу есть три варианта для задания заполнения для текстового блока:

Normal – текст, заданный пользователем;

Purpose — цель, которая была задана в окне свойств модели (цель моделирования была добавлена данным методом);

Viewpoint — точка зрения, которая была задана в окне свойств модели. После нажатия кнопки **ОК** элемент будет добавлен.

2.2. Создание диаграмм декомпозиции

Допустим, что в результате анализа бизнес-процессов выделены три активности, составляющие процесс "Деятельность компании":

- Продажи и маркетинг.
- Сборка и тестирование компьютеров.
- Отгрузка и получение.

Проведем декомпозицию контекстной диаграммы на три перечисленных активности. Декомпозицию можно произвести двумя способами.

Первый способ — выделить декомпозируемую активность, кликнув на ней мышкой (или выделив блок в навигаторе модели на вкладке **Activities**), нажать кнопку **Go to Child Diagram** (рис. 2.26) на панели инструментов.



Рис. 2.26. Кнопка Go to Child Diagram

Второй способ – кликнуть правой клавишей мыши в навигаторе модели на вкладке **Activities** на блоке, который необходимо декомпозировать и из выпадающего списка выбрать пункт **Decompose**.

Появляется диалог **Activity Box Count** (рис. 2.27), в котором выбираем тип диаграммы (декомпозицию можно провести в другой методологии) и число активностей на диаграмме декомпозиции.



Рис. 2.27. Диалог Activity Box Count

Выберем методологию IDEF0, зададим число активностей, равное 3 и нажмем кнопку **ОК**. Появляется диаграмма декомпозиции с несвязанными граничными стрелками (рис. 2.28). Обратите внимание на нумерацию диаграмм и активностей. Контекстная диаграмма имеет номер **A0**, а ее активность имеет номер 0. Диаграмма декомпозиции имеет номер декомпозируемой активности, в нашем случае **A0**, а ее активности имеют номера 1, 2, 3. У не подвергшейся декомпозиции активности перечеркнут левый верхний угол. В левом нижнем углу показывается стоимость работы. Так как мы не задавали стоимость работ, то в активностях показано **0**р.

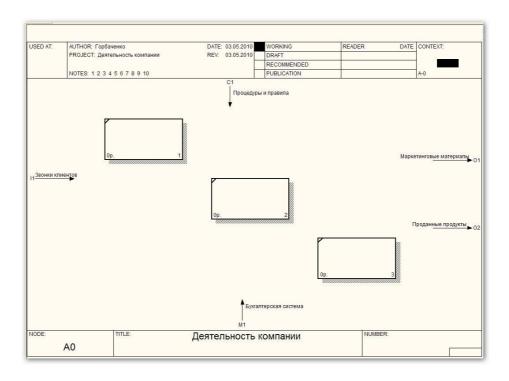


Рис. 2.28. Диаграмма декомпозиции с несвязанными граничными стрелками

Граничные стрелки мигрировали на диаграмму декомпозиции, но не касаются активностей. Кроме того, на диаграмме появились ІСОМ-коды: сокраще-

ния от Input, Control, Output, Mechanism. Для отображения ICOM-кодов в свойствах модели **Model Properties** на вкладке **Display** необходимо включить отображение ICOM-кодов. Несвязанные граничные стрелки необходимо связать с активностями, но предварительно надо задать имена и свойства активностей. Имена и свойства активностей можно задать из контекстного меню выбранной активности, дважды кликнув на активности.

Для связывания стрелок входа, управления и механизма необходимо пе-

рейти в режим редактирования стрелок (кнопка на панели инструментов), щелкнуть по наконечнику стрелки и щелкнуть по соответствующей стороне прямоугольника активности. Для связывания стрелки выхода необходимо в режиме редактирования стрелок щелкнуть по правой стороне активности, а затем – по стрелке. Щелкая по стрелке и соответствующей стороне блока активности, можно построить разветвление стрелки, например, стрелки правил. Для выхода из режима редактирования стрелок необходимо

нажать кнопку указателя на панели инструментов. В результате получаем диаграмму, показанную на рис. 2.29.

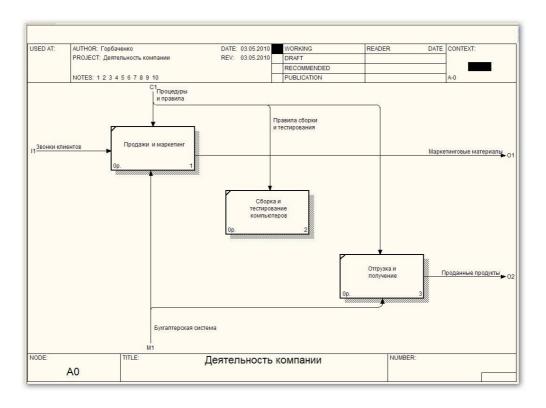


Рис. 2.29. Диаграмма со связанными граничными стрелками

Далее создадим внутренние стрелки, как показано на рис. 2.30. Для рисования внутренней стрелки необходимо в режиме рисования стрелок щелкнуть на стороне блока активности, откуда выходит стрелка, затем по стороне блока активности, куда входит стрелка.

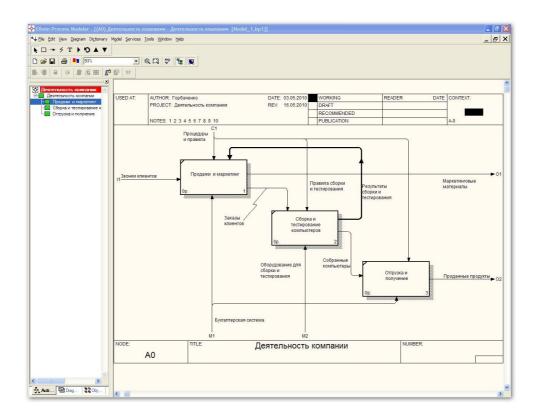


Рис. 2.30. Диаграмма с внутренними стрелками

Необходимо задать имена и свойства стрелок. Установив указатель мыши на имя стрелки (при этом выделяется и сама стрелка), можно перемещать имя, изменять размер поля имени. Указатель в виде молнии (Squiggle) включается с помощью контекстного меню. Кликните правой клавишей мыши на той стрелке, где нужно добавить этот элемент и выберите из выпадающего меню Squiggle. Можно использовать соответствующую кнопку на панели инструментов.

На диаграмме рис. 2.30 введена обратная связь по управлению между активностями "Сборка и тестирование компьютеров" и "Продажи и маркетинг". В контекстном меню этой стрелки задана увеличенная толщина и дополнительная стрелка (Extra Arrowhead).

Добавлена новая граничная стрелка "Оборудование для сборки и тестирования". Эта стрелка автоматически не попадает на диаграмму верхнего уровня, а ее начало заключено в квадратные скобки. Надо щелкнуть правой кнопкой мыши по квадратным скобкам и выбрать в контекстном меню пункт **Arrow Tunnel**. Появляется окно диалога **Border Arrow Editor** (рис. 2.31).

В этом окне выбираем Resolve it border arrow (Разрешить граничную стрелку). В этом случае граничная стрелка будет видна на всех диаграммах верхних уровней. Если выбрать пункт Change it resolved rounded tunnel, то стрелка будет помещена в туннель. Стрелка не будет видна на диаграммах верхних уровней, ее начало будет заключено в круглые скобки. На диаграммах нижних уровней стрелка будет видна в любом случае.



Рис. 2.31. Диалог Border Arrow Editor

Туннелирование может быть применено для изображения малозначимых стрелок. Если на какой-либо диаграмме нижнего уровня необходимо изобразить малозначимые данные или объекты, которые нецелесообразно отображать на диаграммах вышестоящего уровня, то следует туннелировать стрелки на самом нижнем уровне. Такое туннелирование называется туннель "нев-родительской-диаграмме". Другим примером туннелирования может быть ситуация, когда стрелка механизма мигрирует с верхнего уровня на нижний, причем на нижнем уровне этот механизм используется одинаково во всех работах без исключения. В этом случае стрелка механизма на нижнем уровне может быть удалена, после чего на родительской диаграмме она может быть туннелирована, острие стрелки на родительской диаграмме будет изображено в круглых скобках. В комментарии к стрелке или в словаре можно указать, что механизм будет использоваться во всех работах дочерней диаграммы декомпозиции. Такое туннелирование называется туннель "не-в-дочерней-диаграмме".

Для перемещения между диаграммами разных уровней используйте кнопку с изображением треугольника острием вверх на панели инструментов или навигатор модели (рис. 2.32), расположенный слева от диаграммы.



Рис. 2.32. Навигатор

Декомпозируем активность "Сборка и тестирование компьютеров". Эта активность описывается следующим образом [18]. 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.

На основе этой информации на диаграмме декомпозиции создадим 4 активности (табл. 2.3) и стрелки (табл. 2.4).

Таблица 2.3. Активности диаграммы декомпозиции А2

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

Таблица 2.4. Стрелки диаграммы декомпозиции А2

Имя стрелки	Источник стрелки	Tun	Назначение	Тип назначения
(Arrow Name)	(Arrow Source)	источника	стрелки (Arrow	стрелки
,		стрелки	Dest.)	(Ārrow
		(Ārrow	,	Dest.
		Source		Type)
		Type)		71 /
Диспетчер	Персонал производст-		Отслеживание рас-	Mechanism
, ,	венного отдела		писания и управле-	
			ние сборкой и тес-	
			тированием	
Заказы клиен-	Граница диаграммы	Control	Отслеживание рас-	Control
TOB			писания и управле-	
			ние сборкой и тес-	
			тированием	
Заказы на на-	Отслеживание распи-	Output	Сборка	Control
стольные	сания и управление	1	настольных	
компьютеры	сборкой и тести-		компьютеров	
	рованием			
Заказы	Отслеживание распи-	Output	Сборка ноутбуков	Control
на ноутбуки	сания и управление	1		
110 110 1 10 1 1111	сборкой и тести-			
	рованием			
Компоненты	"Tunnel"	Input	Сборка	Input
		1	настольных	1
			компьютеров	
			Сборка ноутбуков	Input
			Тестирование ком-	Input
			пьютеров	1

Имя стрелки (Arrow Name)	Источник стрелки (Arrow Source)	Tun источника стрелки (Arrow Source Type)	Назначение стрелки (Arrow Dest.)	Tun назначения стрелки (Arrow Dest. Туре)
Настольные компьютеры	Сборка настольных компьютеров	Output	Тестирование ком- пьютеров	Input
Ноутбуки	Сборка ноутбуков	Output	Тестирование ком-пьютеров	Input
Персонал производст- венного отде- ла	"Tunnel"	Mechanism	Сборка настольных компьютеров	Mechanism
			Сборка ноутбуков	Mechanism
Правила сборки и тестирования	Граница, диаграммы		Сборка настольных компьютеров	Control
1			Сборка ноутбуков	Control
			Тестирование ком- пьютеров	Control
Результаты сборки и тес- тирования	Сборка настольных компьютеров	Output	Граница диаграммы	Output
1	Сборка ноутбуков	Output		
	Тестирование компь-ютеров	Output		
Результаты тестирования	Тестирование компь- ютеров	Output	Отслеживание расписания и управление сборкой и тестированием	Input
Собранные компьютеры	Тестирование компь-ютеров	Output	Граница диаграммы	Output
Тестировщик	Персонал производственного отдела		Тестирование ком- пьютеров	Mechanism
Указание передать компьютеры на отгрузку	Отслеживание распи- сания и управление сборкой и тестирова- нием	Output	Тестирование ком- пьютеров	Control

Результат декомпозиции показан на рис. 2.33.

На диаграмме рис. 2.33 отключено отображение теней: в меню **Model>Model Properties>Display** отключено отображение теней **(Shadows)**. На диаграмме декомпозиции A2 введены и затуннелированы (не попадают на диаграмму верхнего уровня) входная стрелка "Компоненты" и стрелка механизма "Персонал производственного отдела". Эта стрелка имеет разные имена после разветвления. Для более наглядного представления диаграммы использован разный цвет стрелок.

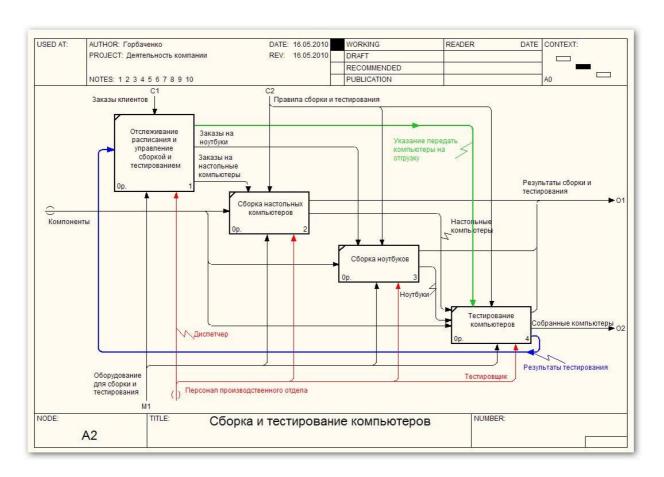


Рис. 2.33. Диаграмма декомпозиции А2

Для перемещения по модели целесообразно использовать навигатор, (Model Explorer), который после декомпозиции имеет вид рис. 2.34.



Рис. 2.34. Навигатор после декомпозиции

На рис. 2.34 отображена вкладка **Activities** (**Активности**), на которой видны иерархически упорядоченные активности модели. На вкладке **Diagrams** изображены иерархически упорядоченные диаграммы модели. Вкладка **Objects** отображает имеющиеся в словаре, но не использованные активности (**Unused Activities**). Активность можно «перетащить» на диаграмму.

2.3. Создание диаграммы дерева узлов

Диаграмма дерева узлов показывает иерархию работ в модели. Для создания диаграммы выбираем в меню пункт **Diagram>Add Node Tree**. В первом окне визарда (мастера) построения дерева (рис. 2.35) необходимо ввести имя диаграммы, узел верхнего уровня (имя активности корня дерева) и число уровней (**Number of Levels**).



Рис. 2.35. Первый диалог построения диаграммы дерева узлов

В одной модели можно построить множество диаграмм дерева узлов, выбирая различные активности в качестве корня и задавая различное число уровней (глубину дерева). Имя и номер диаграммы дерева узлов по умолчанию совпадают с именем и номером корневой активности.

Во втором диалоге (рис. 2.36) задаются свойства диаграммы дерева узлов. Задание свойства Bullet last level (Маркер последнего уровня) означает, что последний уровень декомпозиции будет показан в виде списка (рис. 2.37). Группа свойств Connection Style (Стиль соединения) позволяет выбрать стиль соединительных линий — диагональные (по умолчанию) или ортогональные.



Рис. 2.36. Диалог настройки диаграммы дерева узлов

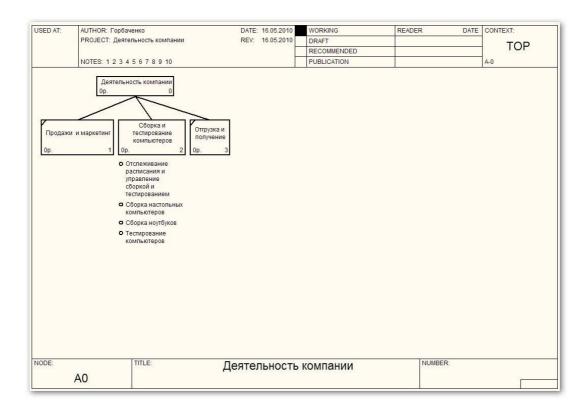


Рис. 2.37. Диаграмма дерева узлов

Для отображения нижнего уровня дерева узлов в виде прямоугольников необходимо отключить свойство **Bullet last level**. Полученное дерево показано на рис. 2.38.

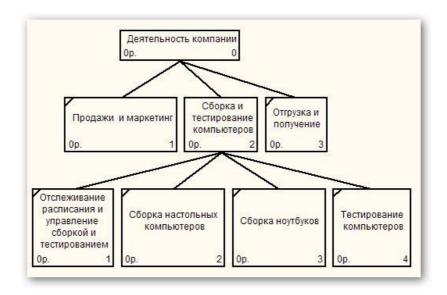


Рис. 2.38. Модифицированная диаграмма дерева узлов

2.4. Создание FEO-диаграммы

Диаграммы "только для экспозиции" (FEO) представляют собой просто картинки, отображающие альтернативные точки зрения, отдельные детали диаграммы и т. п. Эти диаграммы автоматически не поддерживают синтаксис стандарта IDEF0.

Для создания FEO-диаграммы выбираем пункт меню **Diagram>Add FEO diagram**. В первом диалоге (рис. 2.39) задаем имя диаграммы и выбираем, для какой диаграммы модели создается FEO-диаграмма.

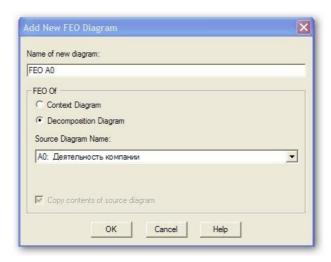


Рис. 2.39. Первый диалог создания FEO-диаграммы

По умолчанию на FEO-диаграмму копируется выбранная диаграмма модели. Удалим в скопированной диаграмме некоторые стрелки (рис. 2.40). Обратите внимание, что в диаграмме рис. 2.40 явно нарушены требования стандарта IDEF0.

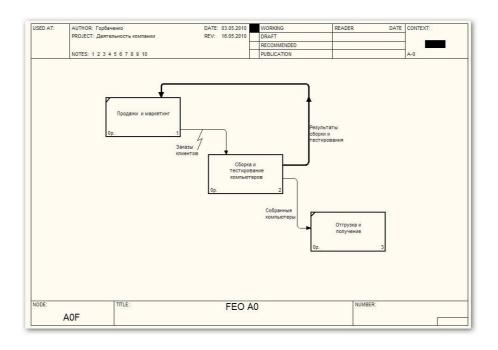


Рис. 2.40. FEO-диаграмма

Для перехода между стандартной диаграммой и FEO-диаграммой можно использовать навигатор и кнопку

. По нажатию на эту кнопку происходит переход к FEO-диаграмме и диаграмме дерева узлов на выбранном уровне модели.

2.5. Расщепление и слияние моделей

Слияние и расщепление моделей необходимо для коллективной работы над моделью. Руководитель проекта может создать декомпозицию верхнего уровня и провести расщепление модели на отдельные модели. Аналитики работают над отдельными моделями, а затем сливают отдельные модели в единую модель. Отдельная ветвь модели может быть отщеплена для использования в качестве независимой модели.

Проведем расщепление активности "Сборка и тестирование компьютеров". На диаграмме A0 правой кнопкой щелкаем на активности "Сборка и тестирование компьютеров" и выбираем из контекстного меню **Split** model. Возникает диалог **Split Options** (рис. 2.41).

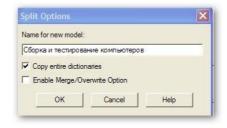


Рис. 2.41. Диалог Split Options

Задаем имя модели (лучше назвать по имени расщепляемой активности) и свойства. Зададим свойство **Copy entire dictionaries**, позволяющее копировать словари в отщепляемую модель.

После подтверждения в навигаторе появилась новая модель, на диаграмме A0 модели "Деятельность компании" появилась стрелка вызова "Сборка и тестирование компьютеров".

Контекстная диаграмма модели "Сборка и тестирование компьютеров" имеет вид рис. 2.42. В старой модели активность "Сборка и тестирование компьютеров" стала недекомпозированной.

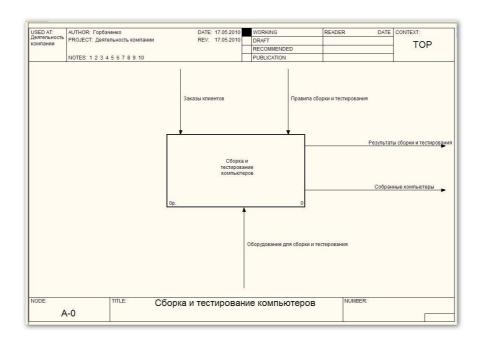


Рис. 2.42. Контекстная диаграмма модели "Сборка и тестирование компьютеров"

На контекстной диаграмме модели "Сборка и тестирование компьютеров" создадим новую стрелку "Неисправные компоненты". На диаграмме декомпозиции направим эту стрелку, как показано на рис. 2.43.

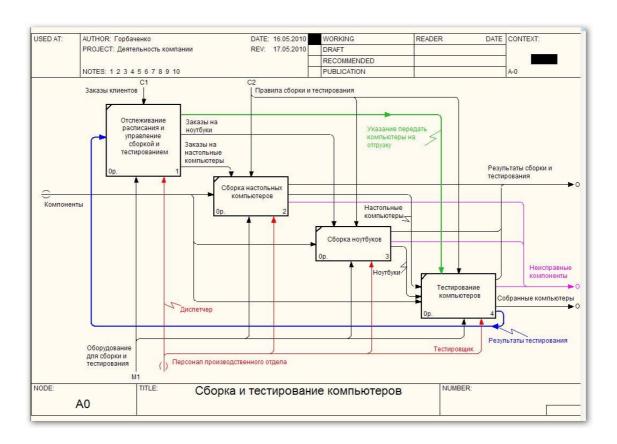


Рис. 2.43. Диаграмма декомпозиции модели "Сборка и тестирование компьютеров"

Теперь произведем слияние моделей. На диаграмме A0 модели деятельность компании щелкнем правой кнопкой мыши по активности "Сборка и тестирование компьютеров" и из контекстного меню выберем Merge model. В диалоге слияния моделей (рис. 2.44) включаем опцию Cut/Paste entire dictionaries и щелкаем по кнопке OK.

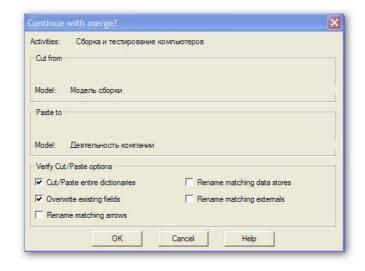


Рис. 2.44. Диалог слияния моделей

Две модели слились. Модель "Сборка и тестирование компьютеров" осталась и может быть сохранена в отдельном файле. В модели "Деятельность компании" исчезла стрелка вызова. Появилась неразрешенная граничная стрелка

"Неисправные компоненты". Эту стрелку туннелируем вручную. Полученная диаграмма декомпозиции модели "Деятельность компании" показана на рис. 2.45.

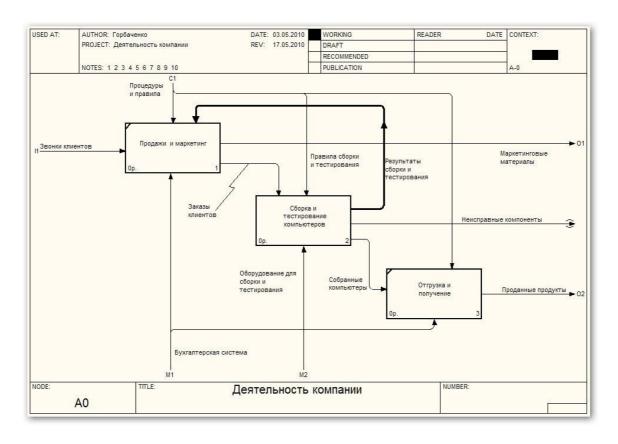


Рис. 2.45. Диаграмма декомпозиции модели "Деятельность компании"

2.6. Задание для самостоятельной работы

На основе методологии IDEF0 разработать модель работы системы «Поликлиника», обладающей следующими функциональными возможностями:

- хранение данных о пациентах, врачах, приемах, диагнозах, лечениях, лекарствах;
 - корректировка данных о пациентах и врачах;
 - поиск данных о пациентах и врачах по фамилии или адресу;
 - добавление нового лекарства с описанием его свойств;
 - поиск справочных данных по лекарствам;
 - регистрация приемов на дому или в поликлинике.
 - формирование статистической информации за отчетный период.

Входные данные системы:

- личные данные пациента;
- личные данные врача;
- сведения по приемам пациентов;
- описание лекарств;
- характеристика заболеваний.

Выходные данные системы:

- статистическая информация за отчетный период:
 - статистика осмотра пациентов на дому;
 - статистика заболеваний;
 - статистика приемов в поликлинике по специалистам;
- справочная информация:
 - сводные данные по врачу;
 - сводные данные по пациенту;
 - сведения о медицинских препаратах.

Модель должна включать в себя контекстную диаграмму, диаграммы декомпозиции, диаграмму дерева узлов, FEO-диаграмму.

3. СОЗДАНИЕ МОДЕЛИ В СТАНДАРТЕ DFD

3.1. Создание контекстной диаграммы

Методология DFD может быть использована *для создания новой модели и для декомпозиции работы*. Создадим новую модель работы "Оформление заказов". Для этого в диалоге создания модели (рис. 3.1) выберем тип модели DFD.

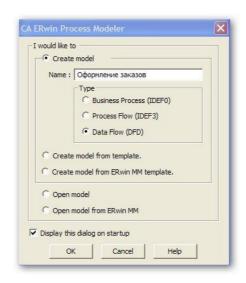


Рис. 3.1. Диалог создания модели

В открывшемся окне появляется единственная контекстная активность. Обратите внимание, что изображение активности немного отличается от ее изображения в методологии IDEF0: у активности закруглены углы. Построим контекстную диаграмму, как показано на рис. 3.2. Зададим имя и свойства активности.

Внесем две внешние сущности: источник и приемник. В нашем случае источником и приемником будет одна внешняя сущность "Клиенты" (рис. 3.3). С целью повышения наглядности покажем на диаграмме две внешние сущности с одинаковыми именами. Обратите внимание, что внешние сущности не участвуют в рассматриваемой работе и не подвергаются декомпозиции. Создается внешняя сущность с помощью кнопки.

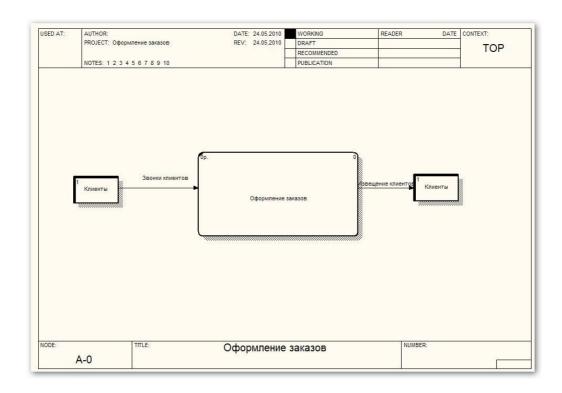


Рис. 3.2. Контекстная диаграмма в методологии DFD



Рис. 3.3. Изображение внешней сущности

3.2. Создание диаграммы декомпозиции

Произведем *декомпозицию* контекстной диаграммы. Оформление заказа начинается с телефонного звонка клиента. При оформлении заказа необходимо проверить, существует ли клиент в базе данных. Если клиента нет в базе, то необходимо занести данные о клиенте в базу клиентов. Далее производится оформление и внесение заказа в список заказов. При оформлении заказа используются как база клиентов, так и список продуктов. Заканчивается оформление заказа извещением по телефону клиента о результатах оформления заказа (можно было бы включить и уточнение заказа). Таким образом, в простейшем случае декомпозиция будет включать две активности: "Проверка и внесение клиента" и "Внесение заказа".

При создании диаграммы декомпозиции в диалоге **Activ- ity Box Count** (рис. 3.4) следует выбрать тип диаграммы декомпозиции (IDEF0 выбрать нельзя).



Рис. 3.4. Диалог Activity Box Count для методологии DFD

Свойство Include Externals & Data Stores означает, что на дочернюю диаграмму будут мигрировать внешние сущности и хранилища данных с родительской диаграммы. При этом родительская диаграмма копируется на дочернюю. На дочерней диаграмме надо удалить родительскую активность и создать необходимое число активностей. При удалении активности на дочерней диаграмме необходимо разрешить туннелированные стрелки на родительской диаграмме. Если это свойство не включено, то внешние сущности и хранилища не мигрируют на дочернюю диаграмму; можно также задать число сущностей на дочерней диаграмме. В нашем примере выберем миграцию внешних сущностей и хранилищ (на нашей родительской диаграмме нет хранилищ, но в принципе они возможны).

Построим диаграмму декомпозиции, как показано на рис. 3.5.

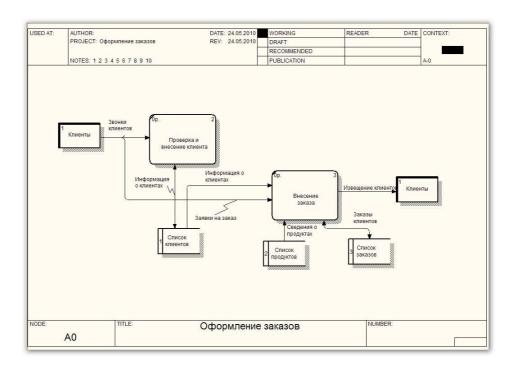


Рис. 3.5. Диаграмма декомпозиции в методологии DFD

На диаграмме декомпозиции присутствуют *хранилища* (рис. 3.6), которые создаются с помощью кнопки .



Рис. 3.6. Изображение хранилища

Некоторые стрелки на диаграмме декомпозиции являются *двунаправлен- ными*. Сначала рисуется однонаправленная стрелка. Чтобы сделать стрелку двунаправленной, щелкните правой кнопкой мыши по стрелке, из контекстного меню выберите пункт **Style** и на вкладке **Style** меню свойств стрелки выберите вариант двунаправленной стрелки (**Bidirectional**).

3.3. Задание для самостоятельной работы

В модели, построенной по методологии IDEF0, декомпозируйте одну из активностей по методологии DFD. Учтите, что в методологии DFD нет стрелок управления и механизмов. При декомпозиции родительской активности по методологии DFD на диаграмму декомпозиции будут мигрировать стреки с родительской диаграммы. Удалите их на диаграмме DFD, а на родительской диаграмме спрячьте в туннель (Change to Tunnel). На диаграмме декомпозиции нарисуйте внешние сущности и хранилища.

4. COЗДАНИЕ МОДЕЛИ В CTAHДAPTE IDEF3

4.1. Создание диаграммы декомпозиции

В стандарте IDEF3 может быть создана контекстная диаграмма. Но обычно этот стандарт используют для декомпозиции активностей (по крайней мере, в BPwin). Проведем декомпозицию активности "Сборка настольных компьютеров". В диалоге **Activity Box Count** (рис. 4.1) выберем нотацию IDEF3 и число работ, равное 4.



Рис. 4.1. Выбор нотации в диалоге Activity Box Count

Возникает диаграмма, содержащая 4 работы (UOW) – рис 4.2.

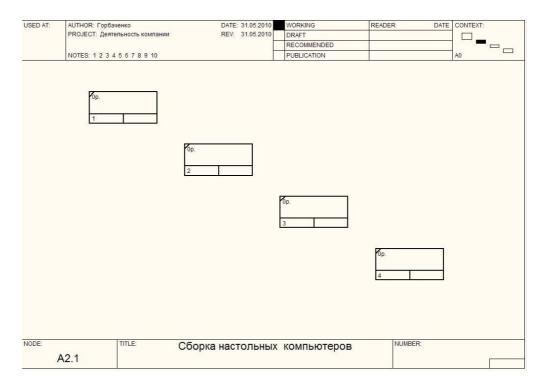


Рис. 4.2. UOW на диаграмме декомпозиции

Примем, что диаграмма декомпозиции должна содержать 7 работ:

- Подготовка компонентов.
- Установка материнской платы и винчестера.
- Установка карт-ридера.
- Установка DVD.

- Установка сетевой карты.
- Инсталляция операционной системы (ОС).
- Инсталляция дополнительного программного обеспечения (ПО).

С помощью кнопки добавим к диаграмме еще три работы. В диалоге **Activity Properties** зададим имена и свойства работ. На вкладке **UOW** (рис. 4.3) зададим объекты (**Objects**), с которыми работает UOW. Например, для работы "Подготовка компонентов" можно указать подготавливаемые компоненты компьютера.

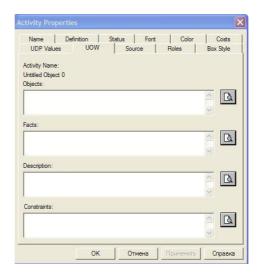


Рис. 4.3. Вкладка UOW в диалоге Activity Properties

Задаются факты (**Facts**) — данные, которые используются работой, например, какие операционные системы доступны. В ограничениях (**Constrains**) указываются ограничения на работу, например, порядок выполнения работ, необходимость в дополнительных работах.

С помощью кнопки создадим объект "Ссылка". Зададим имя ссылки "Компоненты" и свяжем его стрелкой с работой "Подготовка компонентов". В контекстном меню свойств стрелки необходимо с помощью радиокнопки задать тип стрелки Referent (рис. 4.4).

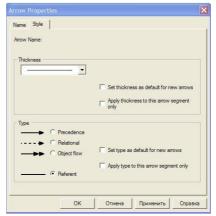


Рис. 4.4. Задание типа стрелки

Свяжем стрелкой работы "Подготовка компонентов" и "Установка материнской платы". Очевидно, что после подготовки компоненты передаются на сборку. Поэтому стиль стрелки изменим на **Object Flow** (Поток объектов). В IDEF3 имена стрелок могут отсутствовать, но BPwin требует задавать имена.

Далее необходимо внести в диаграмму перекрестки. Окончательный вид диаграммы показан на рис. 4.5.

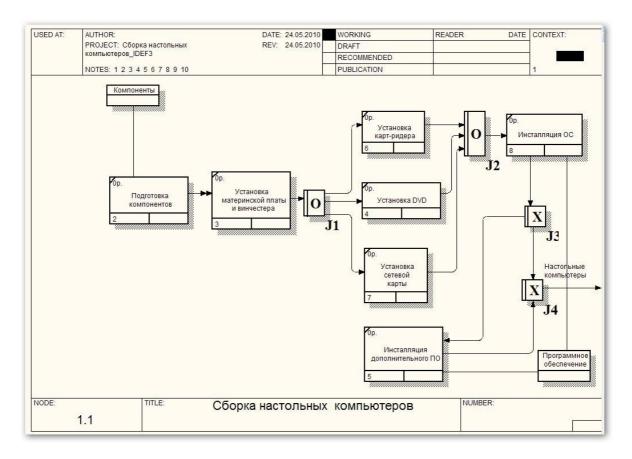


Рис. 4.5. Результат построения диаграммы декомпозиции

При выборе типа перекрестка необходимо руководствоваться свойствами перекрестков (табл. 1.3).

Для создания перекрестка используется кнопка . Левой клавишей мыши укажем на свободной области диаграммы место, где будет добавлен этот перекресток, откроется окно для выбора типа перекрестка (рис. 4.6).



Рис. 4.6. Окно выбора типа перекрестка

4.2. Задание для самостоятельной работы

В модели, построенной по методологии IDEF0, декомпозируйте одну из активностей по методологии IDEF3.

ЛИТЕРАТУРА

- 1. *Грекул, В. И.* Проектирование информационных систем / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2008. 300 с.
- 2. *Маклаков*, *C. В.* Создание информационных систем с AllFusion Modeling Suite. М.: ДИАЛОГ-МИФИ, 2007. 432 с.
- 3. *Маклаков*, *C. В.* Моделирование бизнес-процессов с AllFusion Process Modeler. М.: Диалог-МИФИ, 2008. 224 с.
- 4. Дубейковский, В. И. Практика функционального моделирования с All-Fusion Process Modeler 4.1. Где? Зачем? Как?. М.: Диалог-МИФИ, 2004. 464 стр.
- 5. *Марка*, Д. А. Методология структурного анализа и проектирования / Д. А. *Марка*, К. *Мак Гоуэн*. М.: МетаТехнология, 1993. 240 с.
- 6. Р 50.1.028-2001. Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования. М.: Госстандарт России, 2001. 53 с.
 - 7. IDEF3 // http://en.wikipedia.org/wiki/IDEF3
- 8. *Mayer*, *R. J.* Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report/*R. J. Mayer*, *C. P. Menzel*, *M. K. Painter*, *P. S. deWitte*, *T. Blinn*, *B. Perakath*. Knowledge Based Systems, Inc., 1995. 236 p.
 - 9. IDEF. Process Description Capture Method //http://www.idef.com/IDEF3.htm
 - 10. Верников, Γ . Основы IDEF3// http://www.cfin.ru/vernikov/idef/idef3.shtml
- 11. *Беспалов, Р. С.* Инструментарий разработчика бизнес-процессов // http://www.rb-erp.ru/Texts/Modeling_Bespalov.pdf
- 12. CA ERwin Process Modeler. Process Flow Modeling. Overview Guide r. 7.3. 81 P.
- 13. *Кельтон В.* Имитационное моделирование. Классика CS. 3-е изд. / В. Кельтон, А. Лоу. СПб.: Питер; Киев: Издательская группа BHV, 2004. 847 с.
- 14. Kelton, W. D. Simulation with Arena / W. D. Kelton, P. P. Sadowwski, D. T. Sturrock. McGraw-Hill Science, 2009. 636 p.
- 15. Леоненков, А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose. М.: Интернет-университет информационных технологий-ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2006. 320 с.
- 16. Коберн, A. Современные методы описания функциональных требований к системам. M.: ЛОРИ, 2002. 226 с.
- 17. *Орлов, С. А.* Технологии разработки программного обеспечения. СПб.: Питер, 2002. 464 с.
- 18. *Маклаков*, *С. В.* Создание информационных систем с AllFusion Modeling Suite. М.: ДИАЛОГ-МИФИ, 2007. 432 с.

19. *Ампилогов*, *A*. Первые шаги с CA ERwin Process Modeler // http://www.interface.ru/home.asp?artId=22274