

**Автономная некоммерческая организация высшего образования  
«СЕВЕРО-ЗАПАДНЫЙ ОТКРЫТЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»**

**Направление 09.03.02 «Информационные системы и технологии»**

**Дисциплина «Архитектура информационных систем»**

**Методические указания к выполнению  
контрольной работы**

**Составитель:**

к.т.н., доцент Рахманова Ирина Олеговна

**Санкт-Петербург  
2015**

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ В СТАНДАРТЕ <i>IDEF0</i> .....	5
1.1 Синтаксис и семантика IDEF0-диаграмм .....	5
1.2 Задание № 1. Создание контекстной диаграммы .....	9
1.3 Задание № 2. Создание диаграммы декомпозиции .....	10
1.4 Задание № 3. Создание диаграммы дерева узлов (Node Tree) .....	14
1.5 Задание № 4. Создание диаграммы для презентации (For Exhibition Only) .....	14
1.6 Задание № 5. Расщепление и слияние моделей.....	15
2 МОДЕЛИРОВАНИЕ ПРОЦЕССОВ В СТАНДАРТЕ <i>IDEF3</i> .....	16
2.1 Синтаксис и семантика IDEF3-диаграмм .....	16
2.2 Задание № 6. Создание IDEF3 - диаграммы .....	23
2.3 Задание № 7. Создание сценария.....	25
2.4 Задание № 8. Создание диаграммы Swim Lane .....	26
3 РЕИНЖИНИРИНГ БИЗНЕС-ПРОЦЕССОВ .....	27
3.1 Общая схема разработки моделей .....	27
3.2 Задание № 9. Функционально-стоимостной анализ (Activity Based Costing) .....	28
3.3 Задание № 10. Использование свойств, определяемых пользователем (User Defined Properties)...	30
3.4 Задание № 11. Создание модели TO-BE .....	32
3.5 Задание № 12. Создание организационной диаграммы (Organization Chart) .....	35
4 ИНТЕГРАЦИЯ ФУНКЦИОНАЛЬНЫХ МОДЕЛЕЙ И МОДЕЛЕЙ ДАННЫХ .....	37
4.1 Задание № 13. Создание диаграммы потоков данных (Data Flow Diagramming).....	37
4.2 Задание № 14. Создание сущностей и их экспорт в AllFusion ERwin Data Modeler.....	39

## ВВЕДЕНИЕ

Функциональное моделирование является важнейшим элементом концептуального анализа, который выполняется всякий раз в целях улучшения деятельности организации (фирмы, предприятия, компании, далее - организации). Реорганизация деятельности, особенно если такая реорганизация связана с внедрением корпоративных информационных систем, связана с серьезным риском. Разработка же и анализ функциональной модели деятельности организации позволяет достаточно глубоко погрузиться в предметную область, выявить и оценить бизнес-процессы, протекающие на предприятии, определить информационные потоки, локализовать узкие места в деятельности организации и т.д., что значительно снижает вероятность проявления возможных негативных последствий. Для полноценного анализа необходимо знать не только как работает организация в целом, как она взаимодействует с внешними предприятиями, фирмами, поставщиками, но и как организована деятельность на каждом рабочем месте.

Для функционального анализа на концептуальном уровне удобно использовать простые, доступные для широкого понимания и использования, хорошо проработанные методики. Поскольку реинжиниринг бизнес-процессов, как правило, связан с внедрением или модернизацией информационной системы организации, в настоящее время эти методики объединены стандартом, охватывающим класс информационных систем организационного типа (*Management Information Systems - MIS*) и поддерживаются соответствующими инструментальными программными средствами.

Взаимосвязанная совокупность методик (стандартов) с общим названием *IDEF (Integrated DEFinition)* включает методики функционального, информационного и поведенческого моделирования и проектирования, в которые входят *IDEF-модели*, показанные в табл. 1.

Таблица 1 Семейство стандартов IDEF

Название	Назначение
<i>IDEF0</i>	Функциональное моделирование ( <i>Function Modeling Method</i> )
<i>IDEF1</i> и <i>IDEF1X</i>	Информационное моделирование ( <i>Information and Data Modeling Method</i> )
<i>IDEF2</i>	Поведенческое моделирование ( <i>Simulation Modeling Method</i> )
<i>IDEF3</i>	Моделирование процессов ( <i>Process Flow and Object State Description Capture Method</i> )
<i>IDEF4</i>	Объективно-ориентированное проектирование ( <i>Object-oriented Design Method</i> )
<i>IDEF5</i>	Систематизация объектов приложения ( <i>Ontology Description Capture Method</i> )
<i>IDEF6</i>	Использование рационального опыта проектирования ( <i>Design Rational Capture Method</i> )
<i>IDEF8</i>	Взаимодействие человека и системы ( <i>Human-System Interaction Design</i> )
<i>IDEF9</i>	Учет условий и ограничений ( <i>Business Constraint Discovery</i> )
<i>JDEF14</i>	Моделирование вычислительных сетей ( <i>Network Design</i> )

*IDEF0* реализует методику функционального моделирования сложных систем. Она основана на известной методологии *SADT (Structured Analysis and Design Technique)*, предложенной еще в 1973 г. Д. Россом и впоследствии ставшей основой стандарта *IDEF0*. Эта методика рекомендуется для начальных стадий проектирования сложных систем управления, производства, бизнеса, включающих людей, оборудование, программное обеспечение и т.д.

*IDEF1X* и *IDEF1* реализуют методики инфологического проектирования баз данных. *IDEF1X* включает графический язык для описания информационных объектов и отношений между ними в нотации семантической модели данных «сущность-связь» (*ERD - Entity-Relationship Diagrams*).

Разработка информационной модели по *IDEF1X* выполняется в несколько этапов:

- выясняются цели проекта, составляется план сбора информации, при этом исходные положения для информационной модели обычно следуют из *IDEF0-модели*;
- выявляются и определяются основные сущности - элементы базы данных, в которых будут храниться данные системы;
- выявляются и определяются основные отношения, результаты чего представляются графически в виде так называемых *ERD-диаграмм*;
- детализируются нестандартные отношения, определяются ключевые атрибуты сущностей;
- формируется логическая структура каждой сущности;
- определяются правила ссылочной целостности.

*IDEF2* и *IDEF3* реализуют поведенческое моделирование. Если методика *IDEF0* связана с функциональными аспектами и позволяет отвечать на вопрос: «Что делает эта система?», то в этих методиках детализируется ответ: «Как система это делает». В основе поведенческого моделирования лежат модели и методы имитационного моделирования систем массового обслуживания, сети Петри, возможно применение модели конечного автомата, описывающей поведение системы как последовательности состояний.

Перечисленные методики относятся к так называемым структурным методам.

*IDEF4* реализует объектно-ориентированный анализ больших систем. Он предоставляет пользователю графический язык для изображения классов, диаграмм наследования, таксономии методов.

*IDEF5* направлена на представление онтологической информации приложения в удобном для пользователя виде. Для этого используются символические обозначения (дескрипторы) объектов, их ассоциаций, ситуаций и схемный язык описания отношений классификации, «часть-целое», перехода и т. п. В методике имеются правила связывания объектов (термов) в предложения и аксиомы интерпретации термов.

*IDEF6* направлена на сохранение рационального опыта проектирования информационных систем, что способствует предотвращению структурных ошибок.

*IDEF8* предназначена для проектирования диалогов человека и технической системы.

*IDEF9* служит для анализа имеющихся условий и ограничений (в том числе физических, юридических, политических) и их влияния на принимаемые решения в процессе реинжиниринга.

*IDEF14* предназначена для представления и анализа данных при проектировании вычислительных сетей на графическом языке с описанием конфигураций, очередей, сетевых компонентов, требований к надежности и т.п.

Таким образом, для адекватного представления сложной системы, характеризуемой структурой, выполняемыми процессами (функциями), поведением системы во времени применяют функциональные, информационные и поведенческие модели, пересекающиеся друг с другом.

Функциональная модель системы описывает совокупность выполняемых системой функций, характеризует морфологию системы (ее построение) - состав подсистем, их взаимосвязи.

Информационная модель отображает отношения между элементами системы в виде структур данных (состав и взаимосвязи).

Поведенческая (событийная) модель описывает информационные процессы (динамику функционирования). В ней существенными являются такие категории, как состояние системы, событие, переход из одного состояния в другое, условия перехода, последовательность событий.

# 1 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ В СТАНДАРТЕ *IDEF0*

## 1.1 Синтаксис и семантика *IDEF0*-диаграмм

Метод *IDEF0* можно считать следующим этапом развития хорошо известного графического языка функционального описания систем *SADT* (*Structured Analysis and Design Technique*). Исторически *IDEF0*, как стандарт, был разработан в 1981 году в рамках обширной программы автоматизации промышленных предприятий США, которая носила обозначение *ICAM* (*Integrated Computer Aided Manufacturing*). Собственно семейство стандартов *IDEF* унаследовало свое обозначение от названия этой программы (*IDEF=ICAM DEFINITION*). В процессе практической реализации, участники программы *ICAM* столкнулись с необходимостью разработки новых методов анализа процессов взаимодействия в промышленных системах. При этом кроме усовершенствованного набора функций для описания бизнес-процессов, одним из требований к новому стандарту было наличие эффективной методологии взаимодействия в рамках аналитик-специалист. Другими словами, новый метод должен был обеспечить групповую работу над созданием модели, с непосредственным участием всех аналитиков и специалистов, занятых в рамках проекта.

В результате поиска соответствующих решений появился метод функционального моделирования *IDEF0*. С 1981 года стандарт *IDEF0* претерпел несколько незначительных изменений, в основном ограничивающего характера, и последняя его редакция была выпущена в декабре 1993 года Национальным институтом по стандартам и технологиям США (*NIST*).

Графический язык *IDEF0* достаточно прост и гармоничен. В его основе лежат четыре основных понятия.

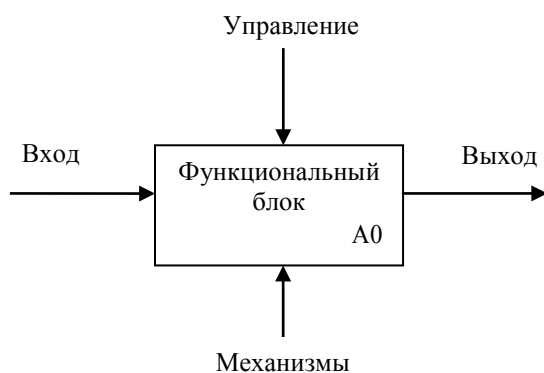
**Первым** из них является понятие *функционального блока* (*Activity Box*). Функциональный блок графически изображается в виде прямоугольника (рис.1) и представляет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано, по возможности, в глагольном наклонении (например, «производить продукцию», а не «производство продукции»).

Каждая из четырех сторон функционального блока имеет свое определенное значение (роль), а именно:

- верхняя сторона имеет значение «Управление» (*Control*);
- левая сторона имеет значение «Вход» (*Input*);
- правая сторона имеет значение «Выход» (*Output*);
- нижняя сторона имеет значение «Механизм» (*Mechanism*).

Каждый функциональный блок в рамках единой рассматриваемой системы должен иметь свой уникальный идентификационный номер.

**Вторым** понятием метода *IDEF0* является понятие *интерфейсной дуги* (*Arrow*). Интерфейсные дуги часто называют потоками или стрелками. Интерфейсная дуга отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, отображенную данным функциональным блоком.



**Рис.1. Функциональный блок**

Графическим отображением интерфейсной дуги является однонаправленная стрелка. Каждая интерфейсная дуга должна иметь свое уникальное наименование (*Arrow Label*). По требованию стандарта, наименование должно быть, по возможности, оборотом существительного.

С помощью интерфейсных дуг отображают различные объекты системы. Такими объектами могут быть элементы реального мира (детали, вагоны, сотрудники и т.д.) или информация (документы, данные, инструкции и т.д.).

В зависимости от того, к какой из сторон подходит данная интерфейсная дуга, она носит название «входящей», «исходящей» или «управляющей». Кроме того, «источником» (началом) и «приемником» (концом) каждой функциональной дуги могут быть только функциональные блоки, при этом «источником» может быть только выходная сторона блока, а «приемником» любая из трех оставшихся.

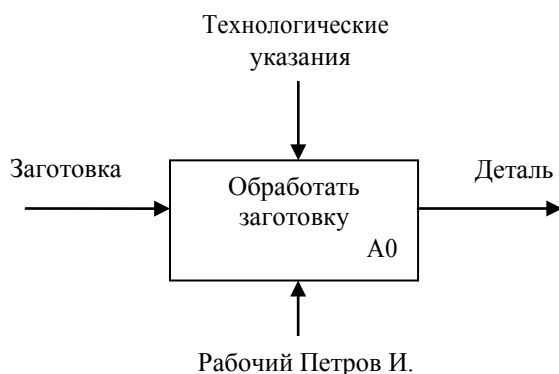
Необходимо отметить, что любой функциональный блок по требованиям стандарта должен иметь по

крайней мере одну управляющую интерфейсную дугу и одну исходящую. Это и понятно - каждый процесс должен протекать по каким-то правилам (отображаемым управляющей дугой), а также выдавать некоторый результат (выходящая дуга), иначе его рассмотрение не имеет смысла.

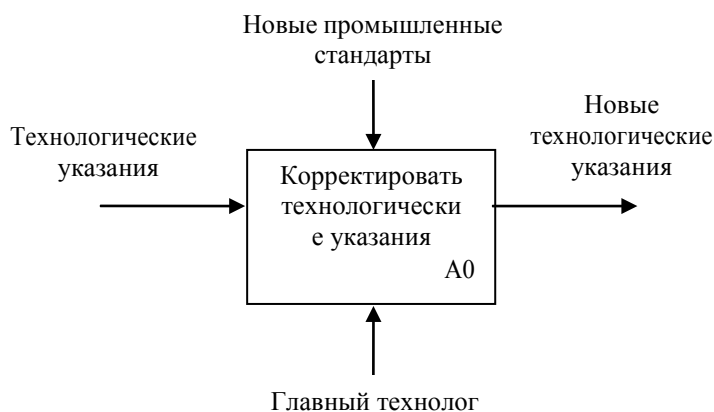
При построении *IDEFO* - диаграмм важно правильно отделять входящие интерфейсные дуги от управляющих, что часто бывает непросто. К примеру, на рис. 2 изображен функциональный блок «Обработать заготовку».

В реальном процессе рабочему, производящему обработку, выдают заготовку и технологические указания по обработке (или правила техники безопасности при работе на станке). Ошибочно может показаться, что и заготовка и документ с технологическими указаниями являются входящими объектами. На самом деле, в этом процессе заготовка обрабатывается по правилам, отраженным в технологических указаниях, которые должны, соответственно, изображаться управляющей интерфейсной дугой.

В другом случае (рис.3) технологические указания обрабатываются главным технологом в целях внесения изменений. При этом старые указания отображаются уже входящей интерфейсной дугой, а управляющим объектом являются, например, новые промышленные стандарты, исходя из которых проводятся данные изменения.



**Рис.2.** Представление функции «Обработать заготовку»



**Рис.3.** Представление функции технолога

Приведенные выше примеры подчеркивают внешне схожую природу входящих и управляющих интерфейсных дуг. Однако для систем одного класса всегда есть определенные разграничения. Например, в случае рассмотрения предприятий и организаций существуют пять основных видов объектов: материальные потоки (детали, товары, сырье и т.д.), финансовые потоки (наличные и безналичные средства, инвестиции и т.д.), потоки документов (коммерческие, финансовые и организационные документы), потоки информации (данные о намерениях, устные распоряжения и т.д.) и ресурсы (сотрудники, станки, машины и т.д.). При этом в различных случаях входящими и исходящими интерфейсными дугами могут отображаться все виды объектов, управляющими - только относящиеся к потокам документов и информации, а дугами-механизмами - только ресурсы.

Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта *IDEFO* от других методов (например, *DFD (Data Flow Diagram)* или *WFD (Work Flow Diagram)*).

**Третьим** основным понятием стандарта *IDEFO* является *декомпозиция (Decomposition)*. Операция декомпозиции применяется при разбиении сложного процесса на составляющие его функции. При этом уровень детализации процесса определяется непосредственно разработчиком модели.

Декомпозиция позволяет постепенно и структурировано представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко читаемой.

Модель *IDEFO* всегда начинается с представления системы как единого целого - одного функционального блока с интерфейсными дугами, выходящими за пределы рассматриваемой области. Такая диаграмма с одним функциональным блоком называется *контекстной диаграммой*, и обозначается идентификатором А-0.

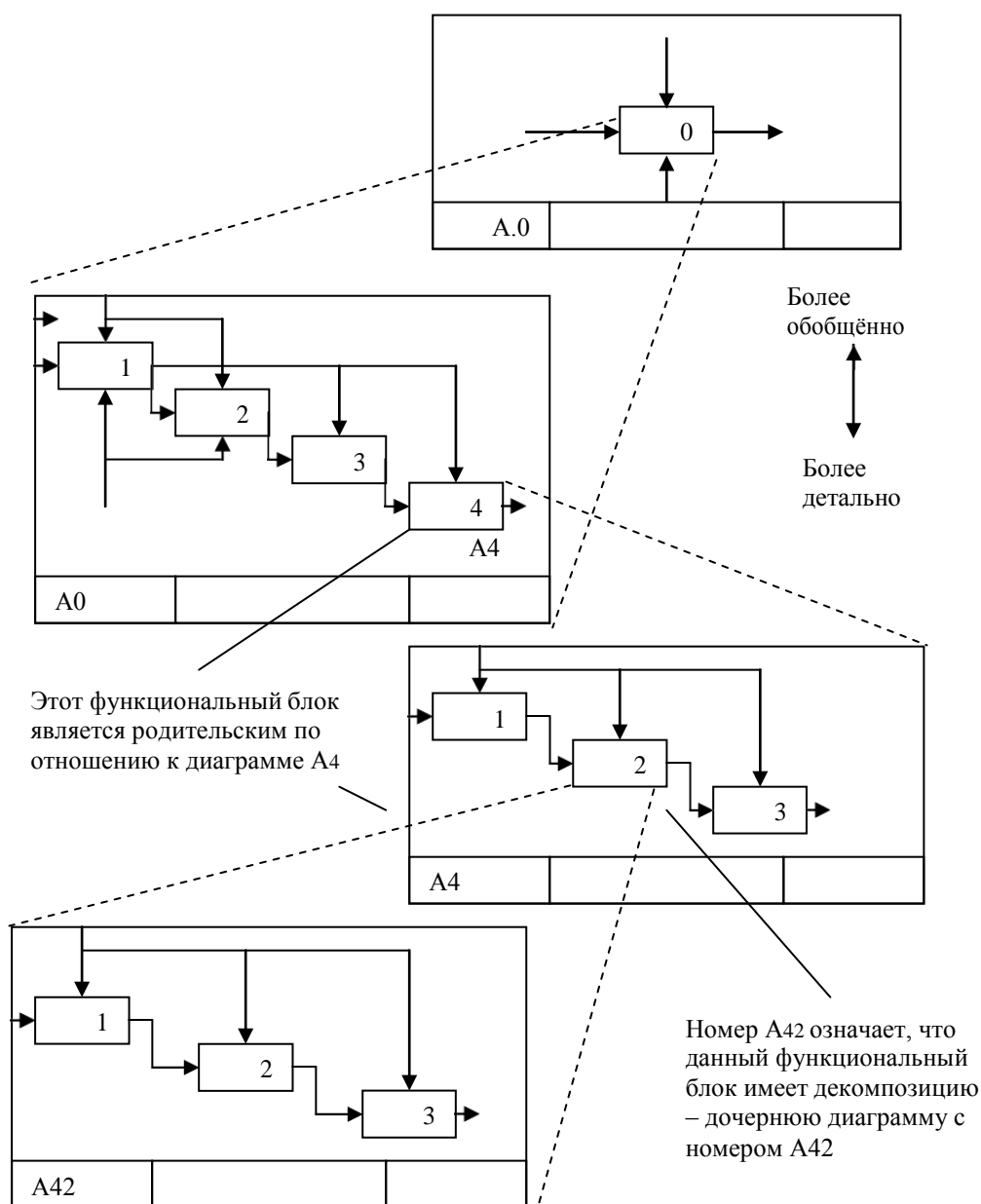
В пояснении к контекстной диаграмме должна быть указана *цель (Purpose)* построения диаграммы в виде краткого описания и зафиксирована *точка зрения (Viewpoint)*.

Определение и формализация цели разработки *IDEFO* - модели является крайне важным моментом. Фактически, цель определяет соответствующие области в исследуемой системе, на которых необходимо сосредоточиться в первую очередь. Например, если моделируется деятельность предприятия с целью построения его информационной системы, то построенная модель будет существенно отличаться от той, которая была разработана для того же самого предприятия, но уже с целью оптимизации технологических процессов.

При построении модели точка зрения определяет основное направление развития модели и уровень необходимой детализации. Четкое фиксирование точки зрения позволяет разгрузить модель, отказавшись от детализации и исследования отдельных элементов, не являющихся необходимыми, исходя из выбранной цели. Например, функциональные модели одного и того же предприятия с точек зрения главного технолога и финансового директора будут существенно различаться по направленности их детализации. Это связано с тем, что в конечном итоге, финансового директора мало интересуют производственные аспекты обработки сырья, а главному технологу ни к чему прорисованные схемы финансовых потоков. Правильный выбор точки зрения существенно сокращает временные затраты на построение конечной модели.

В процессе декомпозиции функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы и называется *дочерней (Child diagram)* по отношению к нему (каждый из функциональных блоков, принадлежащих дочерней диаграмме соответственно называется дочерним блоком - *Child Box*). В свою очередь, функциональный блок - предок называется *родительским блоком* по отношению к дочерней диаграмме (*Parent Box*), а диаграмма, к которой он принадлежит - *родительской диаграммой (Parent Diagram)*. Каждая из подфункций дочерней диаграммы может быть далее детализирована путем аналогичной декомпозиции соответствующего ей функционального блока. Важно отметить, что в каждом случае декомпозиции функционального блока все интерфейсные дуги, входящие в данный блок, или исходящие из него фиксируются на дочерней диаграмме. Этим достигается структурная целостность *IDEFO* - модели. Графически механизм декомпозиции показан на рис.4.

Следует обратить внимание на взаимосвязь нумерации функциональных блоков и диаграмм - каждый блок имеет свой уникальный порядковый номер на диаграмме (цифра в правом нижнем углу прямоугольника), а обозначение под правым углом указывает на номер дочерней для этого блока диаграммы. Отсутствие этого обозначения говорит о том, что декомпозиции для данного блока не существует.



**Рис.4. Декомпозиция функциональных блоков**

Часто бывают случаи, когда отдельные интерфейсные дуги не имеет смысла продолжать рассматривать в дочерних диаграммах ниже какого-то определенного уровня в иерархии, или наоборот - отдельные дуги не имеют практического смысла выше какого-то уровня. Например, интерфейсную дугу, изображающую «деталь» на входе в функциональный блок «Обработать на токарном станке» не имеет смысла отражать на диаграммах более высоких уровней - это будет только перегружать диаграммы и делать их сложными для восприятия. С другой стороны, иногда необходимо избавиться от отдельных «концептуальных» интерфейсных дуг и не детализировать их глубже некоторого уровня. Для решения подобных задач в стандарте *IDEF0* предусмотрено понятие *туннелирования*. Обозначение *туннеля* {Arrow Tunnel} в виде двух круглых скобок вокруг начала интерфейсной дуги обозначает, что эта дуга не была унаследована от функционального родительского блока и появилась (из туннеля) только на этой диаграмме. В свою очередь, такое же обозначение вокруг конца интерфейсной дуги (стрелки) в непосредственной близости от блока-приемника означает тот факт, что в дочерней, по отношению к этому блоку, диаграмме эта дуга отображаться и рассматриваться не будет. Чаще всего отдельные объекты и соответствующие им интерфейсные дуги не рассматриваются на некоторых промежуточных уровнях иерархии. В таком случае они сначала «погружаются в туннель», а затем, при необходимости, «возвращаются из туннеля».

**Последним** из понятий *IDEF0* является *глоссарий* (*Glossary*). Для каждого из элементов *IDEF0* (диаграмм, функциональных блоков, интерфейсных дуг) существующий стандарт подразумевает создание и поддержание набора соответствующих определений, ключевых слов, повествовательных изложений и т.д., которые



характеризуют объект, отображенный данным элементом. Этот набор называется *гlossарием* и является описанием сущности данного элемента. Например, для управляющей интерфейсной дуги «распоряжение об оплате» глоссарий может содержать перечень полей соответствующего дуге документа, необходимый набор виз и т.д. Глоссарий гармонично дополняет наглядный графический язык, снабжая диаграммы необходимой дополнительной информацией.

Обычно *IDEFO-модели* несут в себе сложную и концентрированную информацию. Для ограничения их перегруженности и удобочитаемости в стандарте приняты соответствующие ограничения сложности:

- ограничение количества функциональных блоков на диаграмме тремя-шестью. Верхний предел (шесть) заставляет разработчика использовать иерархии при описании сложных объектов, а нижний предел (три) гарантирует, что на соответствующей диаграмме достаточно деталей, чтобы оправдать ее создание;
- ограничение количества подходящих к одному функциональному блоку (выходящих из одного функционального блока) интерфейсных дуг четырьмя.

Разумеется, строго следовать этим ограничениям вовсе необязательно, однако они являются весьма действенными в реальной работе.

Стандарт *IDEFO* 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 *IDEFO* называется *авторами (Authors)*. Построение первоначальной модели является динамическим процессом, в течение которого авторы опрашивают компетентных лиц о структуре различных процессов. На основе имеющихся положений, документов и результатов опросов создается *черновик (Model Draft)* модели.
2. Распространение черновика для рассмотрения, согласований и комментариев. На этой стадии происходит обсуждение черновика модели с широким спектром компетентных лиц (в терминах *IDEFO-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.
3. 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 *IDEFO* 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 на предприятии (в системе).

Обобщая сказанное, моделирование с использованием метода *IDEFO* включает:

1. Концептуальный этап (определение объекта и цели моделирования, обоснование точки зрения, подбор субъектов и установление сроков моделирования);
2. Этап разработки модели (построение *IDEFO*-диаграмм, проведение циклов «автор-читатель»);
3. Проведение экспериментов с моделью (изучение модели «*AS IS*», получение ответов на вопросы типа «что будет если...»);
4. Интерпретация результатов (построение модели «*TO BE*», документирование результатов).

## 1.2 Задание № 1. Создание контекстной диаграммы

1. Запустите AllFusion Process Modeler.
2. В появившемся диалоге ModelMart Connection Manager (только для AllFusion Process Modeler версии 4.1 и выше) нажмите на кнопку Cancel.
3. Автоматически либо в момент создания новой модели появляется диалог «I would like to». Внесите имя модели «Деятельность компании Quill» и выберите Type - IDEF0. Нажмите OK.
4. Появляется диалог «Properties for New Models», в котором можно выставить значения основных свойств новой модели.
5. Автоматически создается контекстная диаграмма.
6. Закладки в Model Explorer (окно слева) позволяют переходить от одного режима просмотра модели к другому и предоставляют возможность быстро переходить от одной диаграммы к другой.
7. Если Вам непонятно как выполнить то или иное действие, Вы можете вызвать помощь - клавиша F1 или меню Help.
8. Перейдите в меню *Model/Model Properties*. В закладке General диалога Model Properties следует внести имя модели «Деятельность компании Quill», имя проекта «Модель деятельности Quill», имя автора и тип модели -

- Time Frame {AS-IS}.
9. В полях закладки «Purpose» внесите Цель - «Цель: Моделировать текущие (AS-IS) бизнес-процессы компании Quill» и Точку зрения - «Точка зрения: Директор».
  10. В поля закладки «Definition» внесите определение «Эта учебная модель, описывающая деятельность компании Quill» и «Score» - «Общее управление бизнесом компании: исследование рынка, закупка компонент, сборка, тестирование и продажа продуктов».
  11. В закладке Source - «Материалы курса по работе с AllFusion Process Modeler».
  12. Перейдите в меню *Diagram/Diagram Properties* и установите свойства диаграммы.
  13. Перейдите на контекстную диаграмму и правой кнопкой мыши щелкните по работе. В контекстном меню выберите *Name Editor*. В закладке Name внесите имя «Деятельность компании Quill».
  14. В закладке Definition внесите определение «Текущие бизнес-процессы компании Quill».
  15. В закладке Status установите WORKING.
  16. Создайте стрелки на контекстной диаграмме, как показано в табл.2.

**Таблица 2** Стрелки контекстной диаграммы

Наименование (Arrow Name)	Назначение (Arrow Definition)	Тип (Arrow Type)
Бухгалтерская система	Оформление счетов, оплата счетов, работа с заказами	Mechanism
Звонки клиентов	Запросы информации, заказы, тех. поддержка и тд.	Input
Правила и процедуры	Правила продаж, инструкции по сборке, процедуры тестирования, критерии производительности	Control
Проданные продукты	Настольные и портативные компьютеры	Output

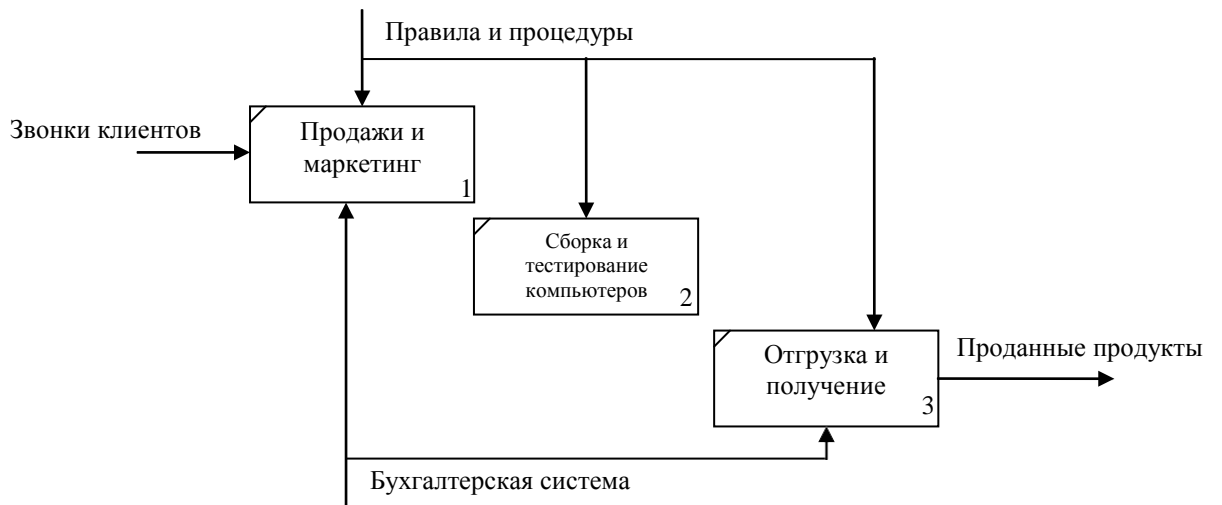
### 1.3 Задание № 2. Создание диаграммы декомпозиции

1. Выберите кнопку перехода на нижний уровень в палитре инструментов и в диалоге *Activity Box Count* установите число работ на диаграмме нижнего уровня - 3 и нажмите ОК. Автоматически будет создана диаграмма декомпозиции. Правой кнопкой мыши щелкните по работе, выберите Name Editor и внесите имя работы. Повторите операцию для всех трех работ. Затем внесите определение, статус и источник для каждой работы согласно табл. 3.

**Таблица 3** Описание работ

Наименование (Activity Name)	Описание (Definition)	Состояние (Status)	Источник (Source)
Продажи и маркетинг	Телемаркетинг и презентации, выставки	WORKING	Материалы курса по работе с AHFusion Process Modeler
Сборка и тестирование компьютеров	Сборка и тестирование настольных и портативных компьютеров	WORKING	Материалы курса по работе с AHFusion Process Modeler
Отгрузка и получение	Отгрузка заказов клиентам и получение компонент от поставщиков	WORKING	Материалы курса по работе с AHFusion Process Modeler

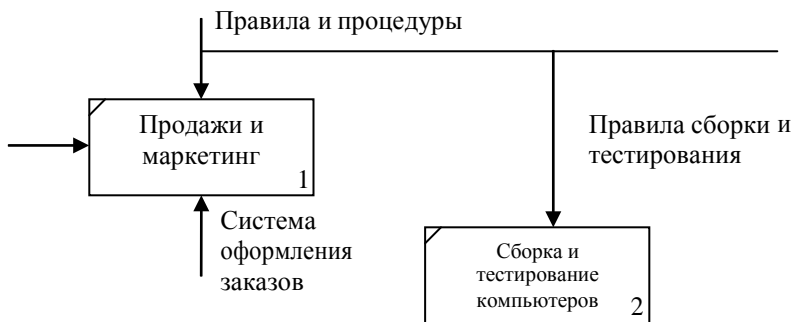
2. Для изменения свойств работ после их внесения в диаграмму можно воспользоваться словарем объектов модели. Вызов словаря - *Model/Diagram Object Editor...* или *Dictionary/Activity...* Результат построения показан на рис.5. Описав имя и свойства работы в словаре, ее можно будет внести в диаграмму позже с помощью кнопки в палитре инструментов. При этом нельзя удалить работу из словаря, если она используется на какой-либо диаграмме. Если Вы удалите работу из диаграммы, из словаря она не удаляется. Имя и описание такой работы может быть использовано в дальнейшем. Для добавления работы в словарь щелкните по кнопке Clear, внесите имя и свойства работы, затем щелкните по Add. Для удаления всех имен работ, не используемых в модели, щелкните по Purge.



**Рис.5. Диаграмма декомпозиции**

3. Перейдите в режим рисования стрелок.

Правой кнопкой мыши щелкните по ветви стрелки управления работы «Сборка и тестирование компьютеров» и переименуйте ее в «Правила сборки и тестирования». Внесите определение для новой ветви: «Инструкции по сборке, процедуры тестирования, критерии производительности и т.д.». Правой кнопкой мыши щелкните по ветви стрелки механизма работы «Продажи и маркетинг» и переименуйте ее в «Система оформления заказов» (рис.6).



**Рис.6. Изменение свойств стрелок**

5. Альтернативный метод внесения имен и свойств стрелок - использование словаря стрелок (вызов словаря - меню *Model/Arrow Editor...* или *Dictionary/Arrow...*) Если Вы опишите имя и свойства стрелки в словаре, ее можно будет внести в диаграмму позже.

6. Создайте новые внутренние стрелки, как показано на рис. 7.

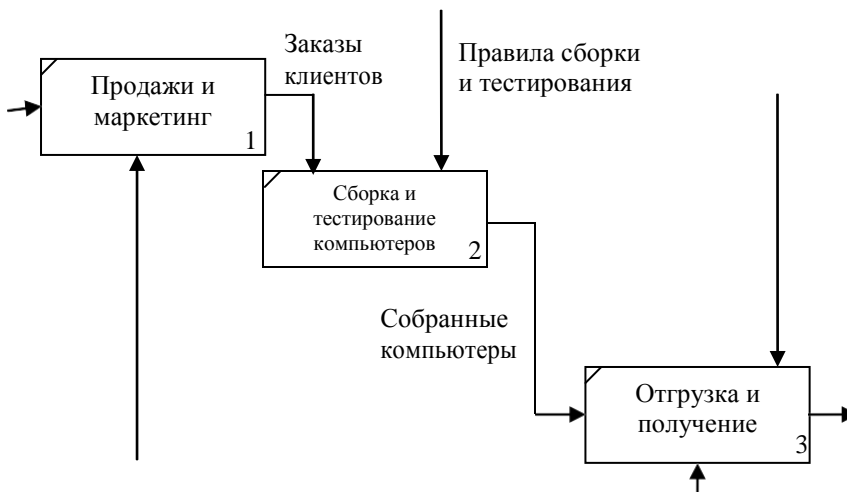


Рис. 7. Создание внутренних стрелок

- Создайте стрелку обратной связи (по управлению) «Результаты сборки и тестирования», идущую от работы «Сборка и тестирование компьютеров» к работе «Продажи и маркетинг» (рис.8). Для наглядности измените стиль стрелки (толщина линий) и установите опцию **Extra Arrowhead** (из контекстного меню). Методом drag&drop перенесите имена стрелок так, чтобы их было удобнее читать. Если необходимо, установите **Squiggle** (из контекстного меню).

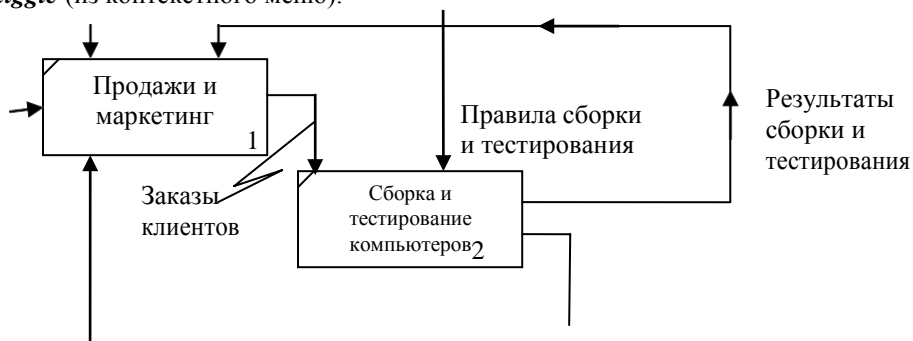


Рис. 8 Создание обратной связи по умолчанию

- Создайте новую граничную стрелку выхода «Маркетинговые материалы», выходящую из работы «Продажи и маркетинг». Эта стрелка автоматически не попадает на диаграмму верхнего уровня и имеет квадратные скобки на наконечнике. Правой клавишей мыши щелкните по квадратным скобкам и выберите меню **Arrow Tunnel** и в диалоге Border Arrow Editor выберите **Resolve Border Arrow**. Для стрелки «Маркетинговые материалы» выберите опцию **Trim** из контекстного меню.
- Декомпозируйте работу «Сборка и тестирование компьютеров» на основе следующей информации. 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 инструкций по сборке. Когда группа компьютеров, соответствующая группе заказов, собрана, она направляется на тестирование. Тестировщики тестируют каждый компьютер и в случае необходимости могут заменить неисправные компоненты.

Тестировщики направляют результаты тестирования Диспетчеру, который на основании этой информации принимает решение о передаче Компьютеров, соответствующих группе заказов на отгрузку (табл. 4,5).

Таблица 4 Работы

Наименование (Activity Name)	Описание (Activity Definition)	Состояние (Activity Status)
Отслеживание расписания и управление сборкой и тестированием	Просмотр заказов, установка расписания выполнения заказов, просмотр результатов тестирования, формирование групп заказов на сборку и отгрузку.	WORKING
Сборка настольных компьютеров	Сборка настольных компьютеров в соответствии с инструкциями и указаниями диспетчера	WORKING
Сборка ноутбуков	Сборка ноутбуков в соответствии с инструкциями и указаниями диспетчера	WORKING
Тестирование компьютеров	Тестирование компьютеров и компонент. Замена неработающих компонент.	WORKING

Таблица 5 Стрелки

Наименование (Arrow Name)	Тип источника (Arrow Source Type)	Источник (Arrow Source)	(Arrow Dest.)	(Arrow Dest. Type)
Диспетчер	Персонал производственного отдела		Отслеживание расписания и управление сборкой и тестированием	Mechanism

Заказы клиентов	{Border}	Control	Отслеживание расписания и управление сборкой и тестированием	Control
Заказы на настольные компьютеры	Отслеживание расписания и управление сборкой и тестированием	Output	Сборка настольных компьютеров	Control
Заказы на ноутбуки	Отслеживание расписания и управление сборкой и тестированием	Output	Сборка ноутбуков	Control
Компоненты	{Tunnel}	Input	Сборка настольных компьютеров	Input
			Сборка ноутбуков	Input
			Тестирование компьютеров	Input
Настольные компьютеры	Сборка настольных компьютеров	Output	Тестирование компьютеров	Input
Ноутбуки	Сборка ноутбуков	Output	Тестирование компьютеров	Input
Персонал производственного отдела	{Tunnel}	Mechanism	Сборка настольных компьютеров	Mechanism
			Сборка ноутбуков	Mechanism
Правила сборки и тестирования	Правила и процедуры		Сборка настольных компьютеров	Control
			Сборка ноутбуков	Control
			Тестирование компьютеров	Control
Результаты сборки и тестирования	Сборка настольных компьютеров	Output	{Border}	Output
	Сборка ноутбуков	Output		
	Тестирование компьютеров	Output		
Результаты тестирования	Тестирование компьютеров	Output	Отслеживание расписания и управление сборкой и тестированием	Input
Собранные компьютеры	Тестирование компьютеров	Output	{Border}	Output
Тестирующий	Персонал производственного отдела		Тестирование компьютеров	Mechanism
Указание передать компьютеры на отгрузку	Отслеживание расписания и управление сборкой и тестированием	Output	Тестирование компьютеров	Control

## 1.4 Задание № 3. Создание диаграммы дерева узлов (Node Tree)

Диаграмма *Node Tree* предназначена для представления иерархии функций (работ) в виде отдельной схемы. При этом начальный уровень и глубину представления декомпозиции можно выбрать произвольно в соответствующем диалоге.

1. Выберите меню **Diagram/Add Node Tree...** Установите опции диалога Node Tree Wizard.
2. Щелкните по «Finish». Создается диаграмма дерева узлов.
3. Для проведения деловой встречи директору необходимо иметь общую картину происходящих в компании бизнес-процессов. Создайте диаграмму дерева узлов. Установите фокус на диаграмме дерева узлов, перейдите в меню **Node tree Diagram Properties** (правая кнопка мыши) к закладке Style диалога **Node Tree Properties** отключите опцию **Bullet Last Level**. Щелкните по ОК. Посмотрите результат.

## 1.5 Задание № 4. Создание диаграммы для презентации (For Exhibition Only)

При обсуждении бизнес-процессов возникла необходимость детально рассмотреть взаимодействие работы «Сборка и тестирование компьютеров» с другими работами. Чтобы не портить диаграмму декомпозиции, создайте FEO диаграмму, на которой будут только стрелки работы «Сборка и тестирование компьютеров».

1. Выберите пункт меню **Diagram/Add FEO Diagram...**
2. В диалоге «Add New FEO Diagram» выберите тип и внесите имя диаграммы. Щелкните по кнопке ОК.
3. Для определения диаграммы перейдите в **Diagram/Diagram Properties** и в закладке Diagram Text внесите определение.
4. Удалите лишние стрелки на диаграмме FEO (рис. 9).



Рис. 9. Диаграмма FEO

Для перехода между стандартной диаграммой, деревом узлов и FEO используйте возможности Model Explorer.

FEO-диаграмма всегда создается на основе существующей IDEF0-диаграммы. Однако в отличие от последней FEO-диаграмма не требует четкого соблюдения требований стандарта. На ней допускаются двунаправленные стрелки, нестандартные подходы стрелок к блокам и др. Основное назначение этой диаграммы - наглядно представить информацию для обсуждения и редактирования.

Допустим, что в результате проведения экспертизы у персонала производственного отдела получена дополнительная информация: оказалось, что неисправные компоненты направляются на отгрузку. Для уточнения информации необходима дополнительная экспертиза в отделе отгрузки. Создайте FEO-диаграмму для проведения такой экспертизы.

Создайте FEO-диаграмму на основе диаграммы A0 и добавьте стрелку «Неисправные компоненты». Стрелка должна идти с выхода «Сборка и тестирование компьютеров» на вход «Отгрузка и получение» (рис. 10).



Рис. 10. Диаграмма для экспертизы

## 1.6 Задание № 5. Расщепление и слияние моделей

В случае большой сложности и (или) неопределенности функциональной области к моделированию могут быть привлечены специалисты различных областей. При этом естественным является желание разбить сложную модель на отдельные независимые подмодели с целью их дальнейшего анализа.

Основным условием расщепления модели является наличие хотя бы одного уровня декомпозиции у отделяемого функционального блока. После отщепления блока он помечается как недекомпозируемый (черточка в левом верхнем углу), а внизу блока появляется направленная вниз стрелка вызова.

Условия слияния моделей:

- обе модели должны быть открыты в AllFusion Process Modeler;
- имя модели-источника должно совпадать с именем стрелки вызова в модели-цели;
- стрелка вызова должна исходить из недекомпозируемого блока;
- модель-источник должна иметь хотя бы одну диаграмму декомпозиции.

### Расщепление модели

1. Перейдите на диаграмму A0. Правой кнопкой мыши щелкните по работе «Сборка и тестирование компьютеров» и выберите Split model.
2. В диалоге Split Option внесите имя новой модели «Сборка и тестирование компьютеров» и щелкните по ОК.
3. Посмотрите на результат: в Model Explorer появилась новая модель, на диаграмме A0 модели «Деятельность компании Quill» появилась стрелка вызова «Сборка и тестирование компьютеров».
4. В модели «Деятельность компании Quill» внесите цель и точку зрения. Цель – «Документировать работу по сборке и тестированию компьютеров», Точка зрения – «Директор».
5. Создайте в модели «Сборка и тестирование компьютеров» новую стрелку «Неисправные компоненты». На диаграмме A0 это будет граничная стрелка выхода, на диаграмме A0 - граничная стрелка выхода от работ «Сборка настольных компьютеров», «Тестирование компьютеров» и «Сборка ноутбуков».

### Слияние моделей

1. Перейдите на диаграмму A0 модели «Деятельность компании Quill».
2. Правой кнопкой мыши щелкните по работе «Сборка и тестирование компьютеров» и выберите Merge model.
3. В диалоге Merge Model включите опцию Paste/Merge Dictionaries и щелкните по ОК.
4. Посмотрите на результат. В Model Explorer видно, что две модели слились. Модель «Сборка и тестирование компьютеров» осталась и может быть сохранена в отдельном файле. На диаграмме A0 модели «Деятельность компании Quill» исчезла стрелка вызова «Сборка и тестирование компьютеров». Появилась неразрешенная граничная стрелка «Неисправные компоненты». Направьте эту стрелку к входу работы «Отгрузка и получение».

## 2 МОДЕЛИРОВАНИЕ ПРОЦЕССОВ В СТАНДАРТЕ *IDEF3*

### 2.1 Синтаксис и семантика *IDEF3*-диаграмм

Стандарт *IDEF3* базируются на концепции моделирования бизнес-процессов организации.

*Бизнес-процесс* - это совокупность технологических и организационно-деловых процессов, выполняемая целенаправленно в рамках заранее заданной организационной структуры. При этом важнейшим атрибутом процесса является время. Иными словами, в отличие от *IDEF0*, *IDEF3* предназначен для моделирования деятельности, развернутой во времени.

Бизнес-процессы могут быть разного масштаба: масштаба предприятия - в него вовлечены работники нескольких подразделений; бизнес-процесс может не выходить за рамки отдела или производственного цеха и т.п. Внутри одного бизнес-процесса, часть составляющих его технологических и организационно-деловых процессов может быть организована в отдельный вложенный бизнес-процесс меньшего масштаба. Отдельные технологические и организационно-деловые процессы могут раскладываться на операции (законченные части процесса, выполняемые на одном рабочем месте, такие как «выписать накладную», «составить договор»), которые в свою очередь делятся на функции (законченные части операции, выполняемые одними и теми же средствами - позвонить, записать, фрезеровать и т.п.).

Концепция бизнес-процессов позволяет по-новому взглянуть на работу предприятия, уточнить и возможно перераспределить обязанности (роли) работников, оценить эффективность использования ресурсов, увидеть недостатки, искусно скрытые в организационной структуре. Таким образом, выявление и идентификация, анализ и внесение изменений в бизнес-процессы могут быть использованы как средства для достижения эффективной работы предприятия. Основную роль в этом играет моделирование бизнес-процессов. Основная цель моделирования при этом состоит в том, чтобы идентифицировать и документировать требуемые аспекты работы системы. Мощностью модели в этом случае заключается в ее способности упростить реальную систему, что дает возможность предсказывать факты о системе на основании соответствующих фактов, полученных из модели.

Метод моделирования процессов *IDEF3* объединяет положительные черты описания (как наиболее точного представления, но в тоже время слишком субъективного) и моделирования. Метод *IDEF3* устраняет конфликт противоречивости описаний различных специалистов.

Моделирование в стандарте *IDEF3* производится с использованием графического представления процесса, материальных и информационных потоков в этом процессе, взаимоотношений между операциями и объектами в процессе. При помощи *IDEF3* описывается логика выполнения работ, очередность их запуска и завершения, т.е. *IDEF3* предоставляет инструмент моделирования сценариев действий сотрудников организации, отделов, цехов и т.п. Например, порядок обработки заказа или события, на которые необходимо реагировать за определенное время, выполнение действий по производству товара и т.д.

Метод *IDEF3* использует категорию *сценариев* для упрощения структуры описаний сложного многоэтапного процесса. Сценарии определяют граничные условия описания. При этом под *сценарием* (*Scenario*) понимается повторяющаяся последовательность ситуаций или действий, которые описывают типичный процесс, присутствующий в системе или описание последовательности изменений свойств объекта в рамках рассматриваемого процесса (например, описание последовательности обработки менеджером заявки).

*IDEF3* предоставляет инструмент для наглядного исследования и моделирования сценариев выполнения процессов. Метод позволяет проводить описание с необходимой степенью подробности посредством декомпозиции. *IDEF3*, как инструмент моделирования, фиксирует следующую информацию о процессе:

- объекты, которые участвуют при выполнении сценария;
- роли, которые выполняют эти объекты (например, менеджер, агент, транспорт и т.д.);
- отношения между работами в ходе выполнения сценария процесса;
- состояния и изменения, которым подвергаются объекты;
- время выполнения и контрольные точки синхронизации работ;
- ресурсы, которые необходимы для выполнения работ.

Основное назначение метода *IDEF3* состоит в том, чтобы предоставить специалисту (эксперту) предметной области инструмент, при помощи которого он сможет представлять знания о выполнении операций в системе или организации в целом. Цель описания может состоять как в документальном оформлении и распространении знаний о процессе, так и в идентификации противоречивости или несовместимости выполнения отдельных операций. Приобретение знаний допускается прямым сбором утверждений о практике выполнения процессов и возникновении различных ситуаций в процессе в форме, которая является наиболее естественной. Описание процесса в *IDEF3* представляет собой структурированную базу знаний, которая состоит из набора диаграмм описания процесса, объектных диаграмм изменения состояний объектов и уточняющих форм.

В стандарте *IDEF3* существуют два типа диаграмм, представляющие описание одного и того же сценария процесса в разных аспектах. Диаграммы, относящиеся к первому типу называются *диаграммами*



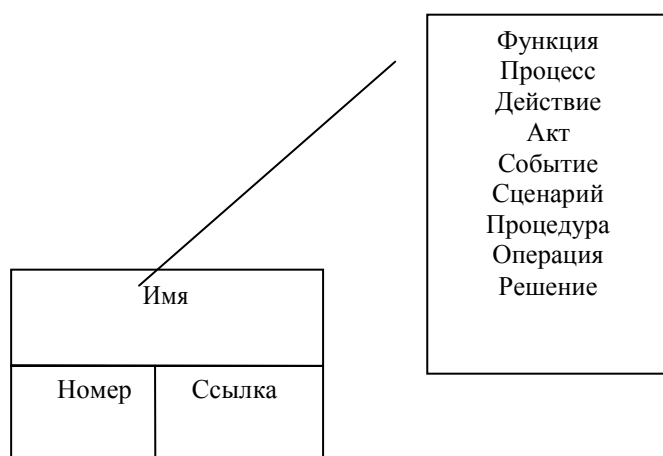
описания последовательности выполнения процесса (*Process Flow Description Diagrams - PFDD*). Второй тип диаграмм описывает состояния объекта и трансформаций в процессе и называется *сеть изменений состояний объекта {Object State Transition Network - OSTN}*.

Если диаграммы *PFDD* описывают процесс, то диаграммы *OSTN* используются для иллюстрации трансформаций объекта, которые происходят на каждой стадии выполнения соответствующих работ (с точки зрения объекта). *Состояния объекта и изменение состояния* являются ключевыми понятиями *OSTN* диаграммы.

Ключевыми понятиями диаграммы *PFDD* являются понятия *процесс* и *логика процесса*. Эти диаграммы обеспечивают механизм визуализации для централизованных процессом описаний сценария. Графическими элементами диаграммы процесса, являются *единицы работ (Unit Of Work - UOW-элементы)*, *связи, перекрестки, ссылки и примечания*. Объекты ссылки и примечания - конструкции, которые являются общими для диаграмм процесса и объектов.

Описание процесса представляет собой всевозможные ситуации, которые могут иметь место в моделируемой системе в логических и временных отношениях. Каждый (*UOW-элемент* представлен полем, отображающим его название (рис.15.). Номер идентификатора назначается последовательно.

В правом нижнем углу *UOW-элемента* располагается ссылка для указания либо на элементы из функциональной модели *IDEF0*, либо для указания на конкретных исполнителей, которые будут выполнять указанную работу.



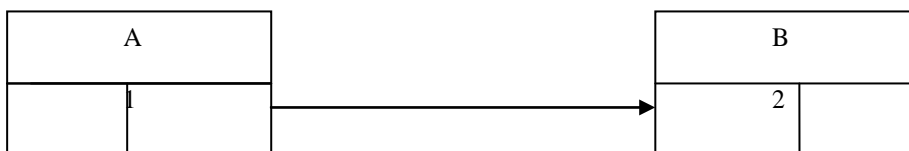
**Рис.15. Синтаксис UOW-элемента**

*IDEF3-элемент* диаграммы описания процесса типа *связь* необходим для связи элементов диаграммы и описания динамики происходящих процессов. Связи используются, прежде всего, для обозначения отношений между функциональными *UOW-элементами*. Для отображения временной последовательности выполнения сценариев в диаграммах описания процесса используются два основных типа связей: *связи старшинства (precedence)* и *относительные связи (relation)*. Для описания специфических отношений между элементами предназначены четыре дополнительных типа связей, как показано на рис.16. Использование в *IDEF3* - диаграммах описания процесса различных типов связей дает возможность пользователям метода фиксировать дополнительную информацию о специфике отношений между элементами диаграммы.



**Рис.16. Типы связей в диаграммах описания процесса**

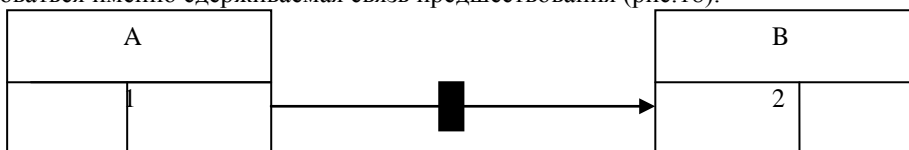
*Связи старшинства* выражают временные отношения старшинства между элементами диаграммы. При этом первый элемент должен завершиться прежде, чем начнет выполняться следующий. Графически, стрелка предшествования (старшинства) отображается сплошной линией с одиночной стрелкой, как показано на рис. 17.



**Рис.17. Простая связь старшинства**

*Сдерживаемые связи старшинства* указывают (в дополнение к семантике запуска связей простого старшинства) некоторые дополнительные условия функционирования системы. Эти дополнительные условия не только выражают то, как система работает, но и устанавливают требования к тому, как система должна себя вести.

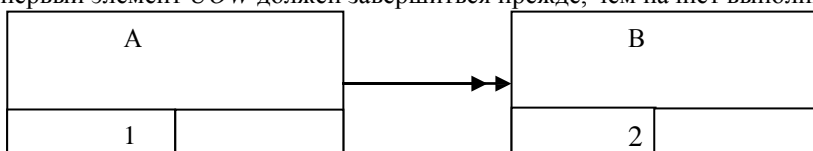
Существует также обобщенное представление сдерживаемых связей предшествования, когда в процессе разработки модели неясно, какая именно связь предшествования должна использоваться, но ясно, что должна использоваться именно сдерживаемая связь предшествования (рис.18).



**Рис.18. Обобщенное представление сдерживаемых связей предшествования**

Использование *относительной связи* указывает на тот факт, что между взаимодействующими элементами диаграммы описания процесса существует отношение неопределенного типа. Относительные связи графически показываются как пунктирные линии.

Для отображения движения объектов используется тип связи *поток объектов*, который показывается как сплошная линия с двойной стрелкой (рис. 19). Этот тип связи выражает перенос одного или нескольких объектов от одного функционального элемента к другому. Этот вид связи элементов *IDEF3* наследует все свойства простой связи старшинства. Таким образом, между UOW-элементами происходит передача объекта(ов), причем первый элемент UOW должен завершиться прежде, чем начнет выполняться следующий.



**Рис.19. Представление связи поток объектов**

*Перекрестки* используются для отображения логики отношений между множеством событий и временной синхронизации активизации элементов диаграмм *IDEF3*. Различают *перекрестки слияния (Fan-in Junction)* и *разветвления (Fan-out Junction)* стрелок (рис.20).

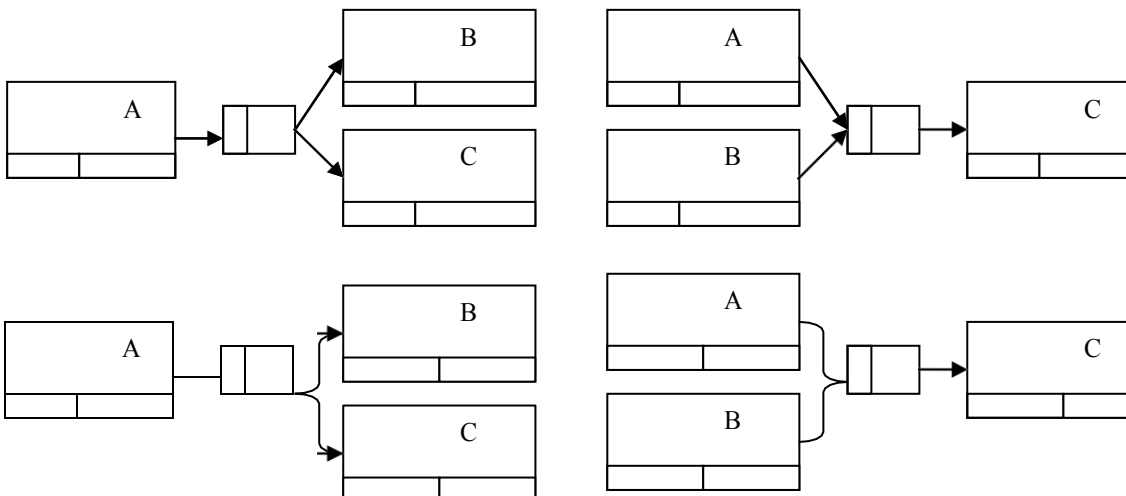
Перекресток не может использоваться одновременно для слияния и для разветвления. При внесении перекрестка в диаграмму необходимо указать тип перекрестка. Тип перекрестка определяет логику и временные параметры отношений между элементами диаграммы. Все перекрестки в PFDDO-диаграмме нумеруются, а каждый номер имеет префикс «J».

Тип перекрестка обозначается как:

& - логический И,

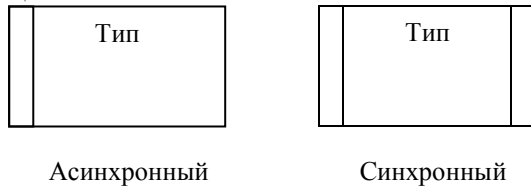
О - логический ИЛИ,

X - логический перекресток НЕЭКВИВАЛЕНТНОСТИ (исключающее ИЛИ).



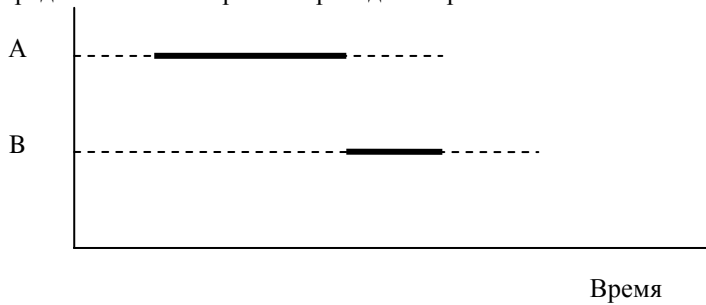
**Рис.20. Перекрестки разветвления и слияния**

Стандарт *IDEF3* предусматривает разделение перекрестков типа & и O на *синхронные* и *асинхронные* (рис.21). Это разделение позволяет учитывать в диаграммах описания процессов синхронизацию времени активизации.



**Рис.21. Обозначение синхронных и асинхронных перекрестков**

Существенным в методе *IDEF3* является понятие *график запуска*. *График запуска* - это визуальное отображение временной последовательности выполнения *UOW*-элементов. Возможный график запуска для ситуации, представленной на рис.17 приведен на рис.22.

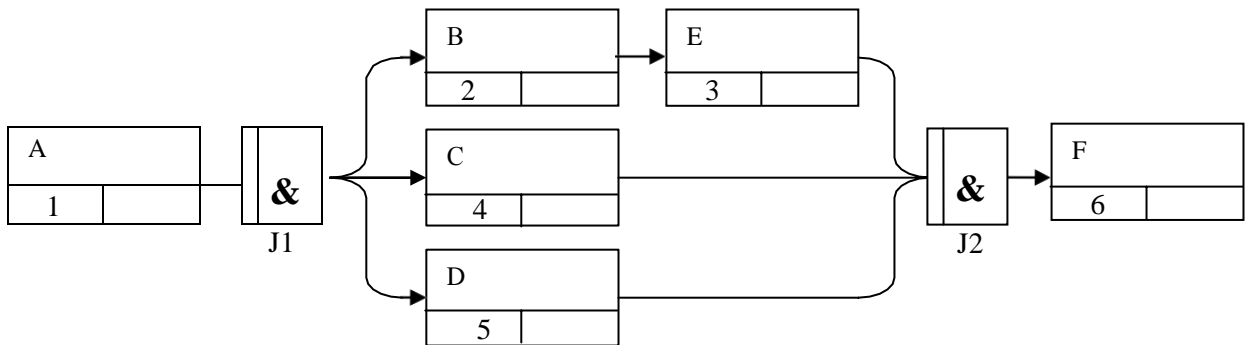


**Рис.22. Пример графика запуска**

Визуальное отображение на графике запуска временной последовательности выполнения *UOW*-элементов помогает правильно понять, как перекрестки описывают логику отношений между элементами диаграммы описания процессов и каким образом перекрестки позволяют синхронизировать по времени выполнение (*UOW*-элементов).

Метод *IDEF3* использует пять логических типов для моделирования возможных последовательностей действий процесса в сценарии, которые приведены в табл.6.

Примеры использования перекрестков, а также графики запуска для них показаны на рис.23...32.



**Рис.23. Использование перекрестков асинхронный AND**

**Таблица 2** Типы перекрестков

Наименование и обозначение	Смысл в случае слияния стрелок ( <i>Fan-in Junction</i> )	Смысл в случае разветвления стрелок ( <i>Fan-out Junction</i> )
<i>Asynchronous</i> AND 	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
<i>Synchronous</i> AND 	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно

<i>Asynchronous OR</i> 	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
<i>Synchronous OR</i> 	Один или несколько предшествующих процессов завершаются одновременно	Несколько следующих процессов запускаются одновременно
<i>XOR</i> 	Только один предшествующий процесс завершен	Запускается только один следующий процесс

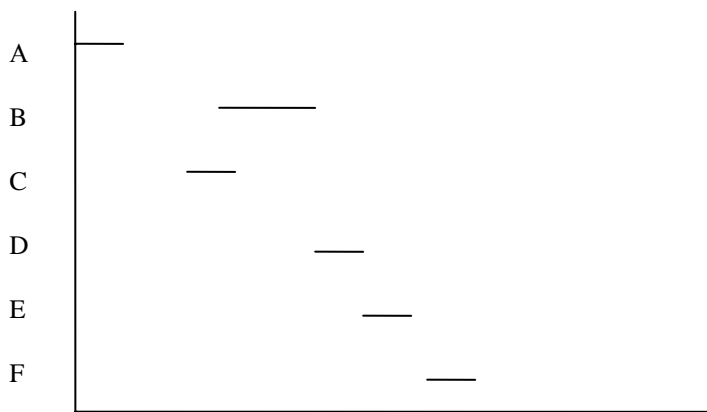


Рис.24. Возможный график запуска для рис.23

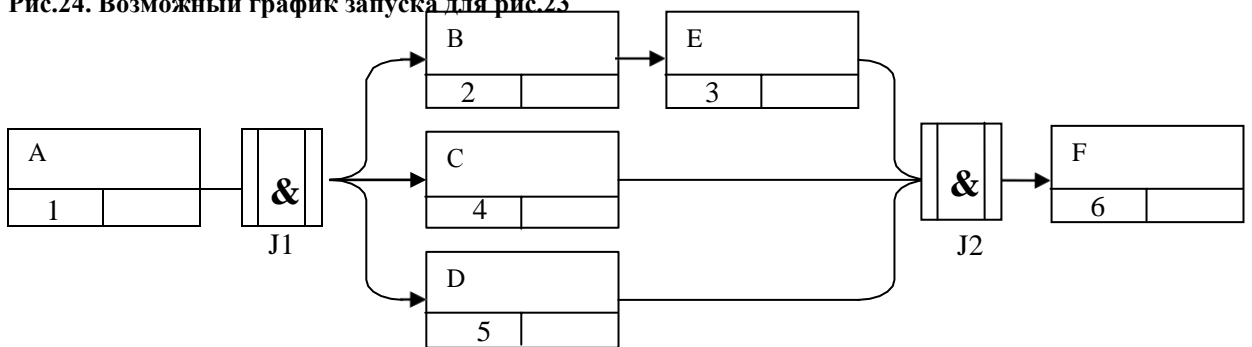


Рис.25. Использование перекрестков синхронный AND

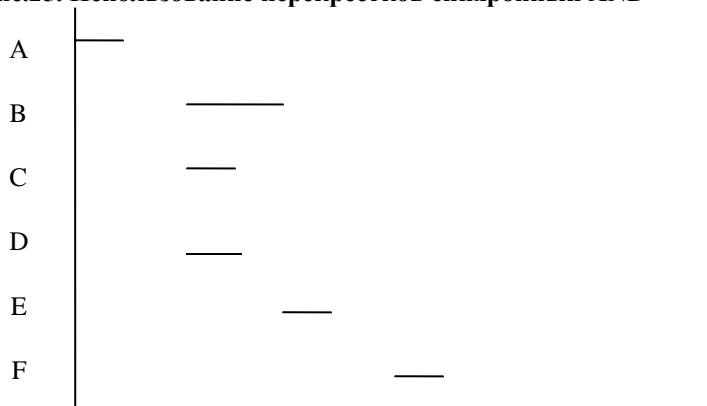


Рис.26. Возможный график запуска для рис.25

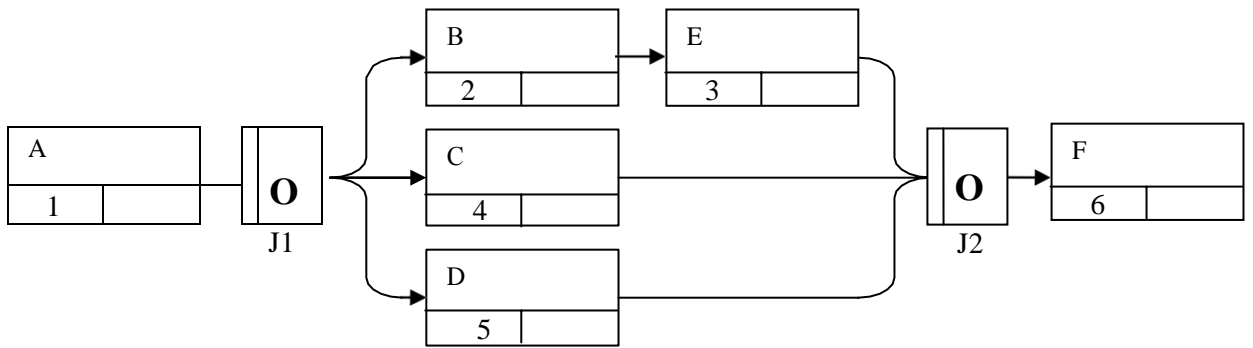


Рис.27. Использование перекрестков асинхронный OR

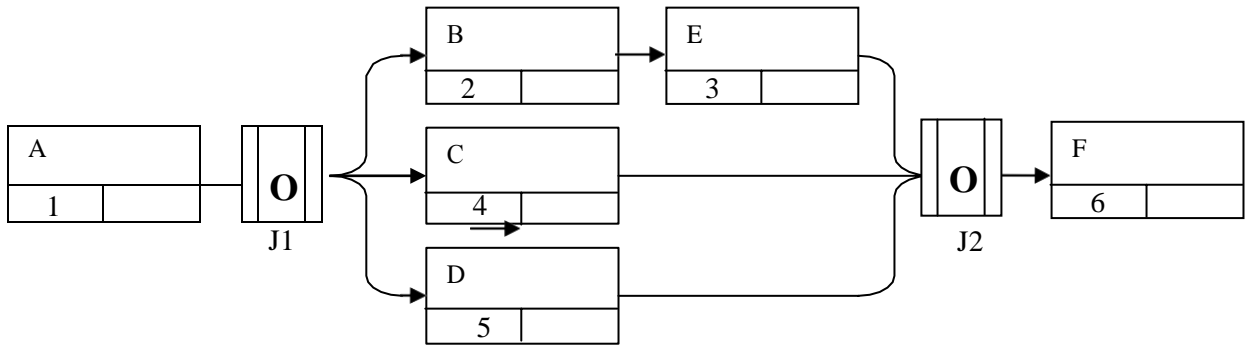


Рис.28. Использование перекрестков синхронный OR

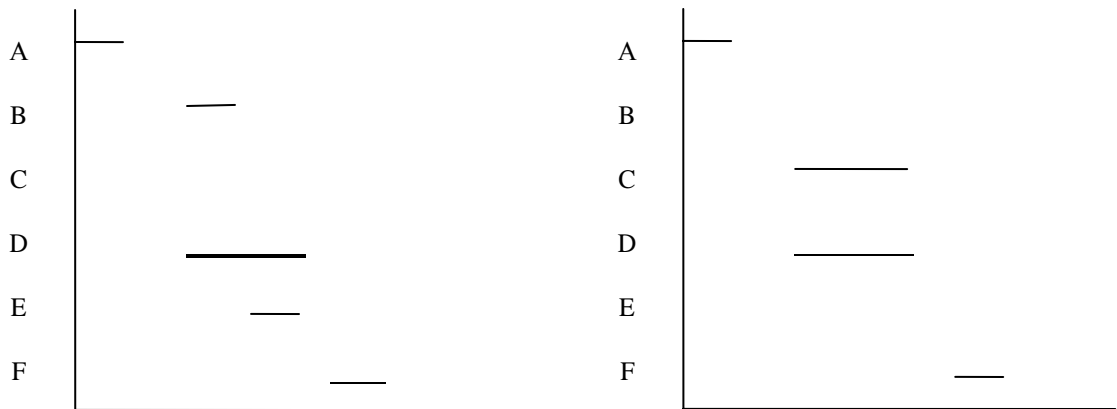


Рис.29. Возможный график запуска для рис.27 и 28

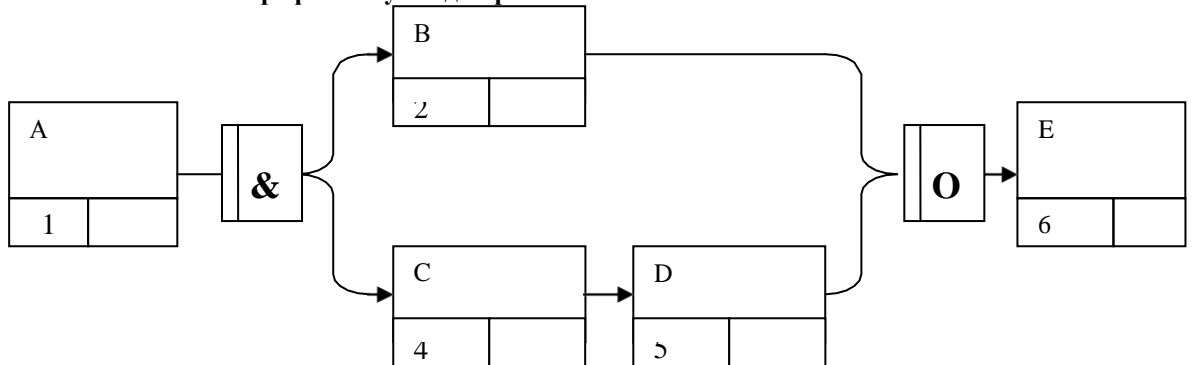


Рис.30. Использование асинхронного перекрестка AND и асинхронного перекрестка слияния OR

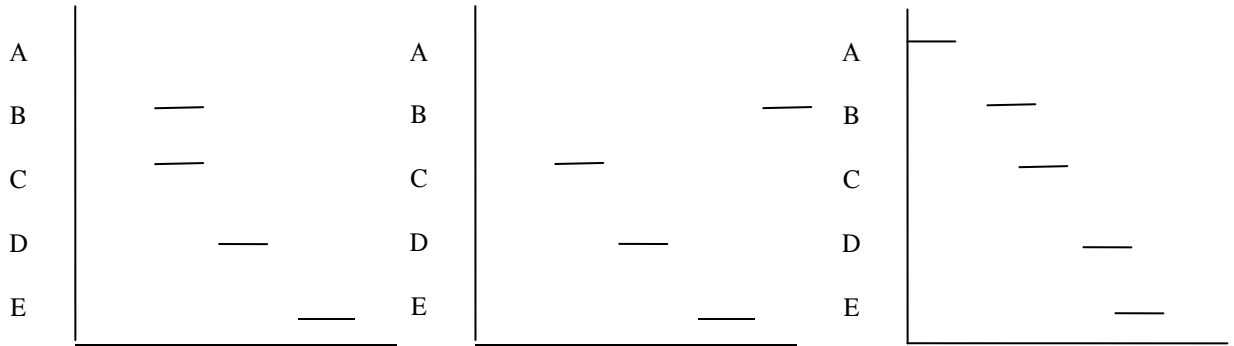


Рис.31. Возможные графики запуска для рис.30

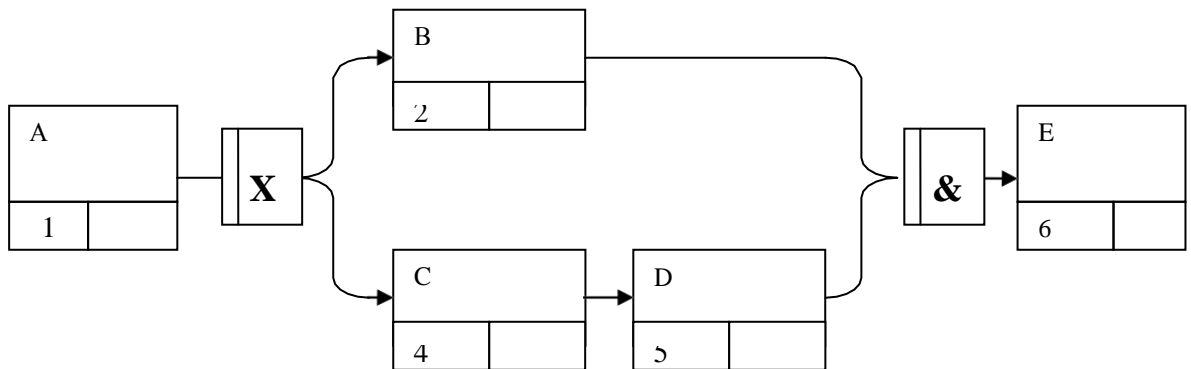


Рис.32. Недопустимое совместное использование перекрестков

Объект *ссылка* (*referent*) применяется, когда некоторый объект (идею, данные) нельзя связать со стрелками (связями) *UOW*-элементами или перекрестками. Графическое обозначение ссылки показано на рис.33.

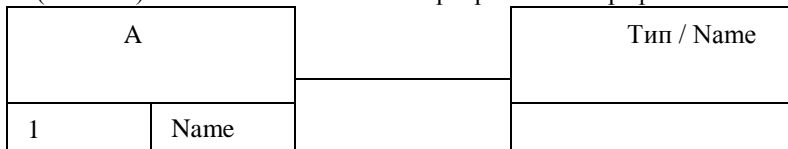
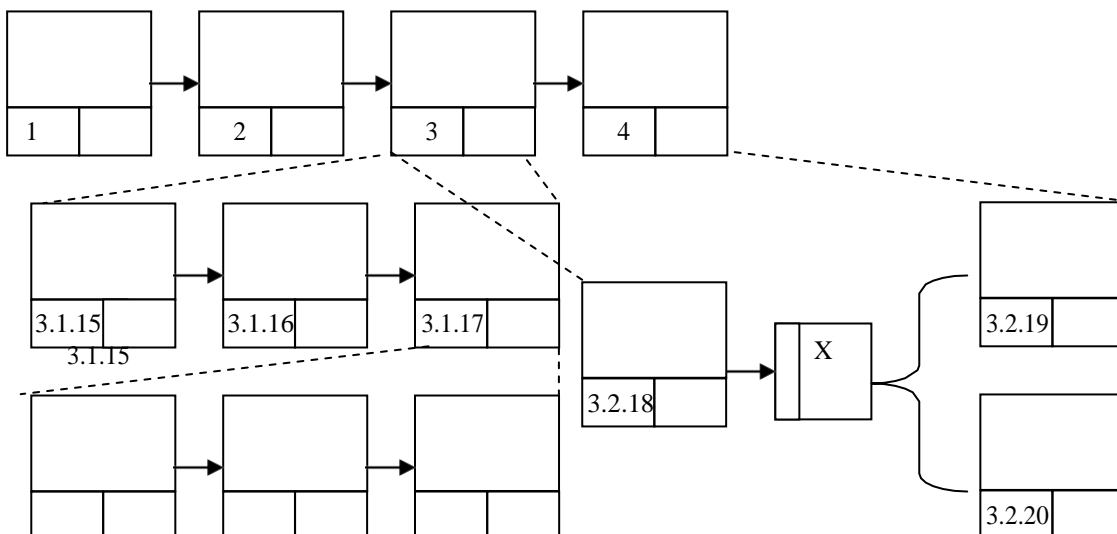


Рис.33. Объект *ссылка*

Метод *IDEF3* дает возможность представлять процесс в виде иерархически организованной совокупности диаграмм. Диаграммы состоят из нескольких элементов описания процесса, причем каждый функциональный элемент потенциально может быть детализирован на другой диаграмме. Такая декомпозиция формирует границы описания процесса, а каждый *[/CW*-элемент рассматривается, как формальная граница некоторой части целой системы, которая описывает весь процесс. Декомпозированная диаграмма, называемая *диаграммой-потомком*, более детально описывает процесс. Декомпозируемый *UOW*-элемент называется *родительским*, а содержащая его диаграмма - *родительской диаграммой*.

Пример декомпозиции и нумерации *UOW*-элементов при этом показан на рис.34.



**Рис.34. Пример декомпозиции и нумерации UOW-элементов**

Применяя декомпозицию неоднократно, можно структурировать описание процесса до любого уровня детализации. Декомпозиция обеспечивает более детальное описание UOW-элементов. Каждый UO ^-элемент может иметь любое число различных декомпозиции на том же самом уровне детализации, с целью представления различных точек зрения или обеспечения большей подробности при описании исходного процесса.

## 2.2 Задание № 6. Создание IDEF3 - диаграммы

1. Перейдите на диаграмму A2 и декомпозируйте работу «Сборка настольных компьютеров». В диалоге Activity Box Count установите число работ 4 и нотацию IDEF3. Возникает диаграмма IDEF3, содержащая работы (UOW).
2. Правой кнопкой мыши щелкните по работе, выберите в контекстном меню Name Editor и внесите имя работы - «Подготовка компонент». Затем в закладке Definition внесите определение «Подготавливаются все компоненты компьютера согласно спецификации заказа». В закладке UOW внесите следующую информацию:

Objects	Компоненты: винчестеры, корпуса, материнские звуковые карты, дисководы CD-ROM и флоппи, материнские платы, видеокарты, модемы, программное обеспечение
Facts	Доступные операционные системы: Windows 2000, Windows NT
Constrains	Установка модема требует установки дополнительного программного обеспечения

1. Внесите в диаграмму еще 3 работы. Имена работ:
  - Установка материнской платы и винчестера;
  - Установка модема;
  - Установка дисковода CD-ROM;
  - Установка флоппи- дисковода;
  - Инсталляция операционной системы;
  - Инсталляция дополнительного программного обеспечения.
2. С помощью кнопки палитры инструментов создайте объект ссылки. Внесите имя объекта внешней ссылки - «Компоненты».
3. Свяжите объект ссылки и работу «Подготовка компонент».
4. Свяжите стрелкой работы «Подготовка компонент» (выход) и «Установка материнской платы и винчестера». Измените стиль стрелки на Object Flow (рис.35).

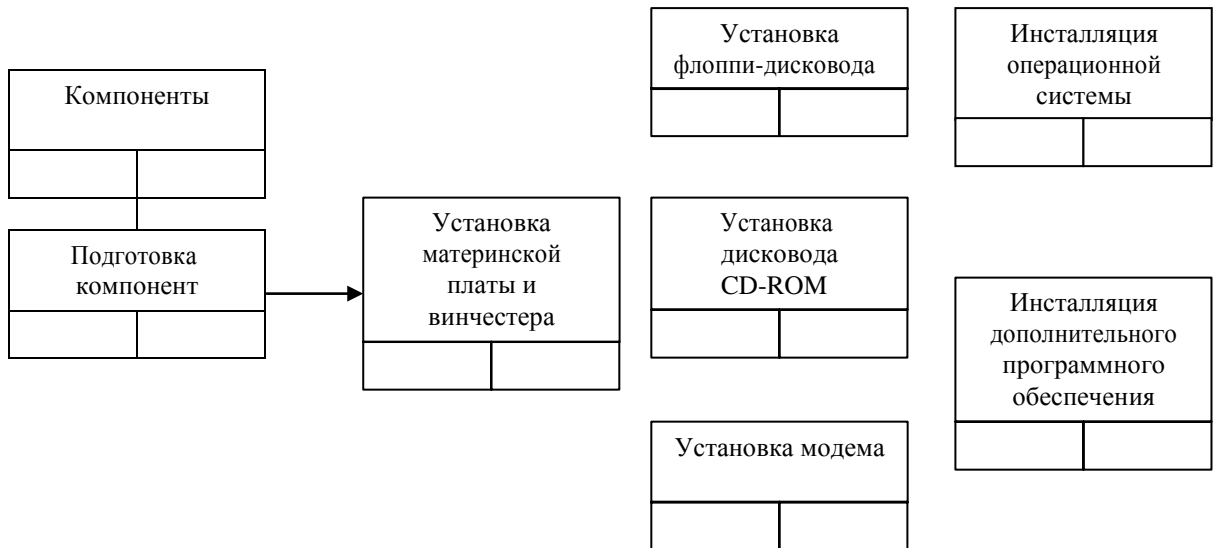


Рис. 35. Подготовка IDEF3-диаграммы

В IDEF3 имя стрелки может отсутствовать, хотя AllFusion Process Modeler воспринимает отсутствие имени как ошибку.

- Внесите два перекрестка типа «асинхронное ИЛИ» и свяжите работы с перекрестками, как показано на рис. 36.

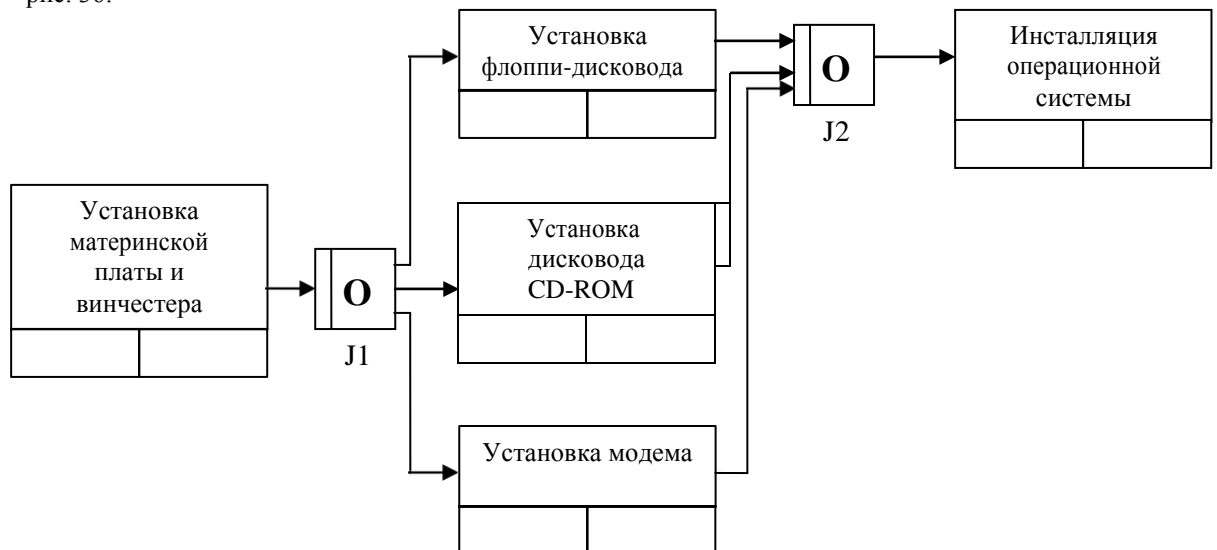


Рис.36. Внесение перекрестков «асинхронный ИЛИ»

Правой кнопкой щелкните по перекрестку для разветвления (fan-out), выберите Name Editor и внесите имя «Компоненты, требуемые в спецификации Заказа».

- Создайте два перекрестка типа «исключающее ИЛИ» и свяжите работы, как показано на рис. 37.



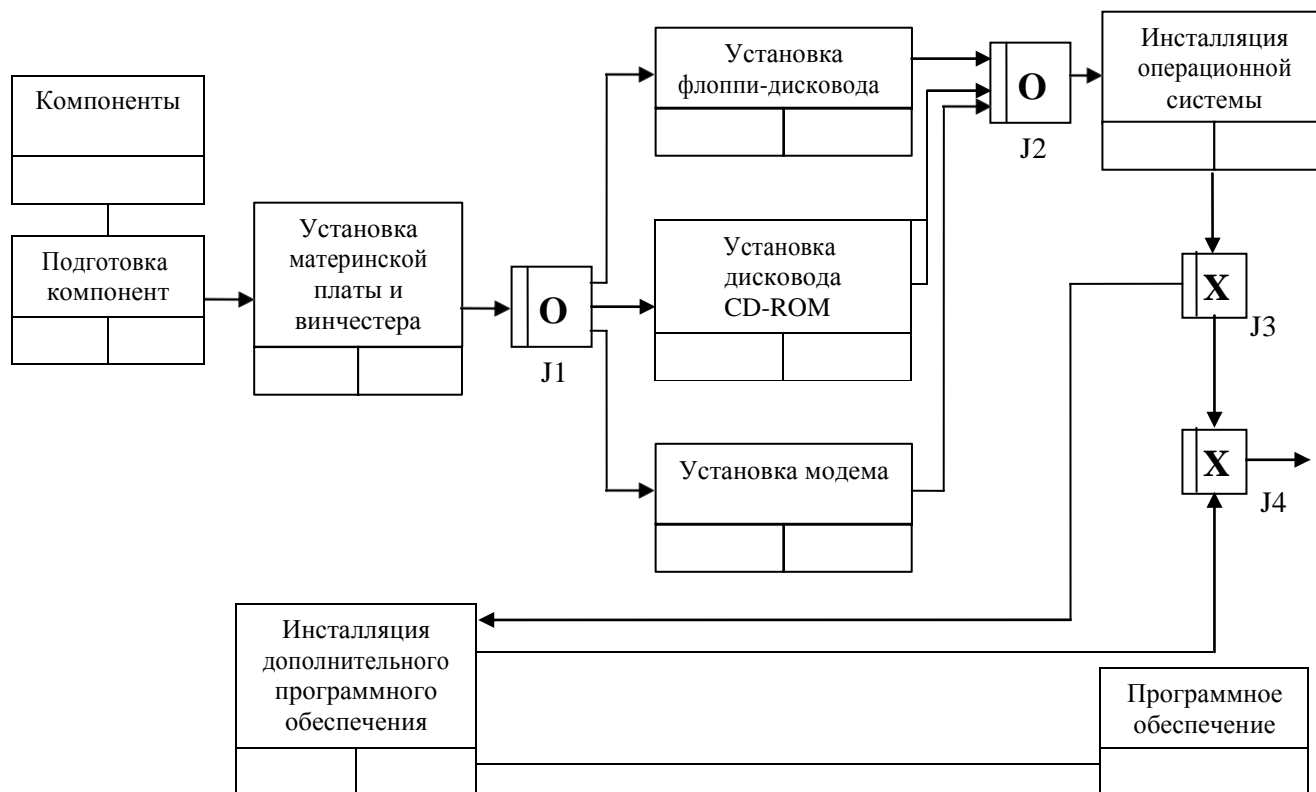


Рис. 37. Диаграмма IDEF3

В результате проведения сеансов экспертизы с тестировщиками выявлена следующая информация:

- Каждый тестировщик имеет собственную периферию (монитор, клавиатура, мышь) для проверки компьютера.
- Каждый тестировщик подсоединяет кабель питания и периферию для настольного компьютера и кабель питания для ноутбука.
- Каждый тестировщик запускает с дискеты программу диагностики, которая тестирует компоненты компьютера.
- Если программа диагностики определяет неработающий компонент, тестировщик заменяет его исправным. Тестирование и замена компонент проводится до тех пор, пока все компоненты компьютера не будут исправлены.
- Каждый проверенный компьютер хранится до тех пор, пока диспетчер не даст распоряжение об отгрузке партии.
- Неисправные компоненты направляются на отгрузку для возврата поставщикам.

На основании этой информации самостоятельно декомпозируйте (в нотации IDEF3) работу «Тестирование компьютеров» диаграммы A2.

Создайте UOW:

- Подключение периферии;
- Запуск программы диагностики;
- Формирование партии;
- Замена неисправных компонент ;

Создайте 4 объекта ссылок:

- Периферия;
- Компьютер;
- Заказы;
- Компоненты;

### 2.3 Задание № 7. Создание сценария

1. Выберите пункт меню Diagram/Add IDEF3 Scenario...
2. Создайте диаграмму FEO на основе диаграммы IDEF3 «Сборка настольных компьютеров» (A22.1).
3. Удалите все перекрестки с диаграммы, показанной на рис.38
4. Свяжите все UOW-элементы по ходу течения времени (слева - направо).

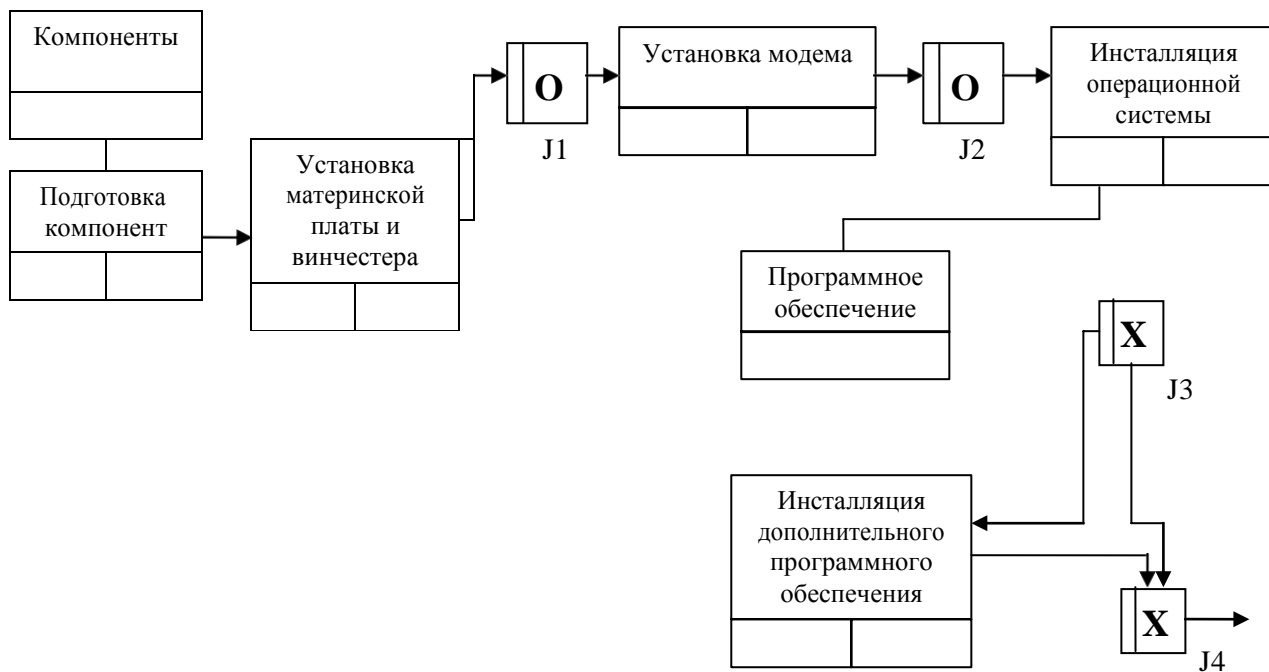


Рис. 38. Создание сценария

## 2.4 Задание № 8. Создание диаграммы Swim Lane

Диаграмма Swim Lane («дорожки») представляет собой разновидность диаграммы IDEF3. Swim Lane позволяет явно описать роли в конкретном процессе. Эта диаграмма разделена на полосы («дорожки»), каждой из которых может быть поставлена в соответствие роль из словаря Role. После преобразования IDEF3-диаграммы в Swim Lane следует просто распределить работы (UOW-элементы) между ролями (по дорожкам).

- 1 Вызовите словарь групп ролей Role Group с помощью меню **Dictionary/Role Group...** и создайте группу с названием «Персонал сборочного цеха».
- 2 Вызовите словарь ролей Role с помощью меню **Dictionary/Role...** и опишите роли, связанные с группой «Персонал сборочного цеха»:
  1. Контролер;
  2. Сборщик;
  3. Системный администратор;
  4. Тестирующий.
- 3 Для каждого объекта IDEF3-диаграммы «Сборка настольных компьютеров», вызвав контекстное меню **Roles**, укажите исполнителей из списка ролей.
- 4 На основе IDEF3-диаграммы создайте диаграмму Swim Lane.
- 5 Распределите UOW-элементы по ролям.

## 3 РЕИНЖИНИРИНГ БИЗНЕС-ПРОЦЕССОВ

### 3.1 Общая схема разработки моделей

Основная цель реинжиниринга бизнес-процессов в терминах *IDEFO* заключается в построении модели *TO-BE* («как будет»). Естественно предположить, что данная модель базируется на реальном положении дел (*AS-IS*). Однако в отличие от модели *AS-IS*, модель *TO-BE* носит предписывающий характер. В ней должны быть учтены и исправлены все недостатки, выявленные в процессе анализа модели *AS-IS*. Очевидно, что процесс построения обеих моделей носит итерационный характер и подчиняется одинаковым правилам моделирования.

В общем случае процесс моделирования включает сбор информации об исследуемой области, документирование полученной информации и представление ее в виде модели, а также уточнение модели посредством итеративного рецензирования. Кроме того, этот процесс подсказывает вполне определенный путь выполнения согласованной и достоверной структурной декомпозиции, что является ключевым моментом в квалифицированном анализе деятельности организации.

В определенном смысле моделирование объединяет итеративный процесс создания модели, нотации, управляющие конфигурацией модели, язык ссылок для диаграмм, язык функций моделей с графическим языком описания системы, а также рекомендации по реализации аналитических проектов. Нотации, управляющие конфигурацией, гарантируют, что новые диаграммы будут корректно встроены в иерархическую структуру модели. Язык ссылок, правила сокращений для ссылок, адресованных к отдельным частям диаграммы, облегчают оформление замечаний при рецензировании модели. Язык функций позволяет декларативно определять правила работы системы, что часто является особенно важным завершающим шагом в описании системы.

Высокая эффективность процесса моделирования обусловлена его организацией, в основе которой лежит разделение функций, выполняемых участниками создания проектов: эксперты являются источниками информации, авторы создают диаграммы и модели, библиотекарь координирует обмен письменной информацией, читатели рецензируют и утверждают модели, а комиссия технического контроля принимает и утверждает модель.

В процессе моделирования сведения об изучаемой системе получают с помощью методов экспертного опроса.

Для получения наиболее полной информации используются различные ее источники (например, чтение документов, опрос экспертов, наблюдение за работой организации). Независимо от конкретного источника информации необходимо определить свои потребности в информации прежде, чем выбрать этот источник. Во время опроса графический язык используется как средство для заметок, которые служат основой для построения диаграмм.

На рис. 39 изображен процесс моделирования, описанный с помощью *IDEFO*-диаграммы. Диаграмма отражает тот факт, что процесс моделирования является итеративной последовательностью шагов, приводящих к точному описанию системы.

Создание модели (блок 2 на рис. 39) - это второй важный этап в процессе моделирования, на котором аналитик документирует полученные им знания о данной предметной области, представляя их в виде одной или нескольких диаграмм. Процесс создания модели осуществляется с помощью специального метода детализации ограниченного субъекта. Автор вначале анализирует объекты, входящие в систему, а затем использует полученные знания для анализа функций системы. На основе этого анализа создается диаграмма, в которой объединяются сходные объекты и функции. Этот конкретный путь проведения анализа и документирования его результатов является особенностью всего процесса моделирования.

Моделирование - инженерная дисциплина. Это означает, что модели создаются, исходя из действительной ситуации, и что эти модели проходят через серию последовательных улучшений до тех пор, пока они в точности не будут представлять реальный мир. Одним из основных компонент при этом является итеративное рецензирование, в процессе которого автор и эксперт многократно совещаются (устно и письменно) относительно достоверности создаваемой модели. Итеративное рецензирование называется циклом «автор-читатель».

Цикл «автор-читатель» начинается в тот момент, когда автор принимает решение распространить информацию о какой-либо части своей работы с целью получения отзыва о ней. Материал для распространения оформляется в виде «папок» - небольших пакетов с результатами работы, которые критически обсуждаются другими специалистами в течение определенного времени. Сделанные письменные замечания также помещаются в папку в виде нумерованных комментариев. Папки с замечаниями являются, таким образом, обратной связью, которую авторы получают на свою работу. Читатели - это те, кто читает и критикует создаваемую модель (блок 4 на рис. 39), а затем помещает замечания в папки. Их работа возможна благодаря тому, что графический язык диаграмм позволяет создавать диаграммы и модели, которые можно легко и быстро читать. Простота графического языка потому не случайна. Она позволяет получить представление о системе, на основе которого можно дать обоснованное заключение о достоверности модели.

Обычно отдельная папка рецензируется одновременно несколькими читателями, и все их замечания

поступают к определенному сроку к автору. Затем автор отвечает на каждое замечание и обобщает критику, содержащуюся в замечаниях. С помощью таких обсуждений можно достаточно быстро обмениваться идеями. Таким образом, поддерживается как параллельный, так и асинхронный просмотр модели, что является наиболее эффективным способом распределения работы в коллективе.

Как правило, модель редко создается одним автором. На практике над различными частями модели могут совместно работать множество авторов, потому что каждый функциональный блок модели представляет отдельный субъект, который может быть независимо проанализирован и декомпозирован. Таким образом, модель сама координирует работу коллектива авторов, в то время как процесс моделирования координирует совместное рецензирование возникающих идей.

Организация своевременной обратной связи имеет важнейшее значение для эффективного моделирования, потому что устаревшая информация потенциально способна свести на нет все усилия по разработке модели. Поэтому выделяется специальная роль наблюдателя за процессом рецензирования. На рис.25 показано, что эту роль выполняет библиотекарь, который является главным координатором процесса моделирования, обеспечивая своевременное и согласованное распространение рабочих материалов. Библиотекарь распространяет полученные от авторов папки, контролирует их движение, рассылает напоминания о своевременном возвращении авторам папок с замечаниями и о сроках ответов авторов на предложения читателей. Кроме того, библиотекарь печатает законченные модели после того, как они одобрены и приняты к использованию.

Модели создаются с конкретной целью, и эта цель записана на диаграмме А-0 модели. В каком-то смысле эта цель определяет, как будет использоваться модель. Следовательно, как только завершено создание модели с требуемым уровнем детализации и модель проверена, она может применяться для достижения поставленной цели. Например, для обеспечения и поддержания ре-инжиниринга бизнес-процессов организации.

В процессе моделирования рекомендуется выделить специальную группу людей, ответственных за то, что создаваемая в процессе анализа модель будет точна и используется в дальнейшем. Эта группа, называемая условно комиссией технического контроля (блок 5 на рис. 39), отвечает за контроль качества моделей, создаваемых авторами проекта. Комиссия следит за выполняемой работой и ее соответствием конечным целям всего проекта. Специалисты комиссии обсуждают модель и оценивают, насколько она может быть использована и будет использована соответствующим образом в ходе выполнения проекта для достижения его глобальных целей.

Таким образом, комиссия технического контроля находится в наиболее выгодном положении при определении текущего направления развития проекта и выработке предложений по его корректировке. Модели, которые достигли желаемого уровня детализации и точности с точки зрения технических требований, направляются в комиссию технического контроля для обсуждения и утверждения. Комиссия оценивает, насколько применима данная модель. Если модель признана комиссией применимой, она публикуется. В противном случае авторам направляются замечания для необходимой доработки.

## **3.2 Задание № 9. Функционально-стоимостной анализ (Activity Based Costing)**

*Функционально-стоимостной анализ (ФСА)* — метод определения стоимости и других характеристик изделий, услуг и потребителей, в основе которого лежит использование функций и ресурсов, задействованных в производстве, маркетинге, продаже, доставке, технической поддержке, оказании услуг, обслуживании клиентов, а также в обеспечении качества.

Метод ФСА разработан как «операционно-ориентированная» альтернатива традиционным финансовым подходам. В частности, в отличие от традиционных финансовых подходов метод ФСА:

- предоставляет информацию в форме, понятной для персонала предприятия, непосредственно участвующего в бизнес-процессе;
- распределяет накладные расходы в соответствии с детальным просчетом использования ресурсов, подробным представлением о процессах и функциях их составляющих, а также их влиянием на себестоимость.

В основе применения метода ФСА лежит разработка и применение на практике ФСА-моделей. Цель создания ФСА-модели для совершенствования деятельности предприятий — достичь улучшений в работе предприятий по показателям стоимости, трудоемкости и производительности. Проведение расчетов по ФСА-модели позволяет получить большой объем информации для принятия решения. При этом данная информация, особенно взаимосвязи отдельных ее элементов, для лиц, принимающих решения, является, как правило, неожиданной. Полученная информация позволяет обосновывать и принимать решения в процессе применения других методов совершенствования финансово-хозяйственной деятельности организации. Основные направления использования ФСА-модели для реорганизации бизнес-процессов — это повышение производительности, снижение стоимости, трудоемкости, времени и повышение качества.

Функционально-стоимостной анализ позволяет выполнить следующие виды работ:

- формирование релевантной информации об эффективности деятельности центров ответственности на предприятии;
- определение и проведение общего анализа себестоимости бизнес-процессов на предприятии (маркетинг, производство продукции и оказание услуг, сбыт, менеджмент качества, техническое и гарантийное обслуживание и др.);
- проведение сравнительного анализа и обоснование выбора рационального варианта технологии реализации бизнес-процессов;
- проведение функционального анализа, связанного с установлением и обоснованием выполняемых структурными подразделениями предприятий функций с целью обеспечения выпуска высокого качества продукции и оказания услуг;
- определение и анализ основных, дополнительных и ненужных функциональных затрат;
- сравнительный анализ альтернативных вариантов снижения затрат в производстве, сбыте и управлении за счет упорядочения функций структурных подразделений предприятия;
- анализ интегрированного улучшения результатов деятельности предприятия и др.

Построение функционально-стоимостных моделей осуществляется на основе применения методологической и технологической взаимосвязи между *IDEFO*- и ФСА-моделями.

Связанность методов *IDEFO* и ФСА заключается в том, что оба метода рассматривают финансово-хозяйственную деятельность предприятия как множество последовательно выполняемых функций, а дуги входов, выходов, управления и механизмов функций *IDEFO*-модели соответствуют стоимостным объектам и ресурсам ФСА-модели.

На уровне функционального блока связь *IDEFO* и ФСА-моделей базируется на трех принципах:

1. Функция характеризуется числом, которое представляет собой стоимость или время выполнения этой функции.
2. Стоимость или время функции, которая не имеет декомпозиции, определяется разработчиком модели.
3. Стоимость или время функции, которая имеет декомпозицию, определяется как сумма стоимостей (времен) всех подфункций на данном уровне декомпозиции.

Следует отметить, что в *AllFusion Process Modeler* реализован упрощенный вариант ФСА. В то же время в программном продукте *EasyABC* ФСА реализован полностью, но в явном виде программная поддержка взаимосвязи между *IDEFO*-моделью и *EasyABC* поддерживается только путем экспорта диаграммы *Node Tree*.

#### Исходные данные для ФСА

На производственном участке работают 5 сборщиков и 1 тестировщик.

В среднем в день собирается 12 настольных компьютеров и 20 ноутбуков.

Двое сборщиков - стажеры.

Зарплата диспетчера - 500\$ в месяц, сборщиков и тестировщиков - 10\$ в час, стажеров - 3\$ в час.

Средняя стоимость компонент для настольного компьютера - 800\$, для ноутбука - 1400\$.

1. В диалоге *Model Properties* (вызывается из меню *Model*) в закладке *ABC Units* установите единицы измерения денег и времени.
2. Перейдите в *Model/Cost Center Editor...* и в диалоге *Cost Center Editor* внесите название и определение центров затрат (табл. 7).

Таблица 7 Описание центров затрат

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

3. Для внесения центра затрат наберите наименование, определение и щелкните по кнопке *Add*.

Стоимость каждой работы отображается в нижнем левом углу прямоугольника.

Для отображения частоты или продолжительности работы перейдите в диалог *Model Properties* (закладка *Display*), и переключите радиокнопки в группе *ABC Units*. Можно вообще отключить режим отображения информации об *ABC*, отключив опцию *Activity Cost/Freq/Dur.* в диалоге *Model Properties* или меню *View*.

4. Для назначения стоимости работе следует щелкнуть по ней правой кнопкой мыши и выбрать в контекстном меню *Cost Editor*.
5. По табл. 8 внесите параметры *ABC*.

Таблица 8 Исходные данные для ФСА

Activity Name	Cost Center	Cost (SUS)	Duration (Days)	Frequency
---------------	-------------	------------	-----------------	-----------

Отслеживание расписания и управление сборкой и тестированием	Управление	25,00	1,00	1,00
Сборка настольных компьютеров	Рабочая сила	5,00	1,00	12,00
	Компоненты	800,00		
Сборка ноутбуков	Рабочая сила	7,50	1,0	20,00
	Компоненты	1400,00		
Тестирование компьютеров	Рабочая сила	2,00	1,00	32,00

6. Посмотрите результат - стоимость работы верхнего уровня.
7. Сгенерируйте отчет Activity Cost Report.

### 3.3 Задание № 10. Использование свойств, определяемых пользователем (User Defined Properties)

1. Перейдите в *Model/UDP Definition Editor...* в диалоге User-Defined Property Dictionary Editor внесите название категорий.
2. Для внесения категории необходимо в поле New Keyword внести наименование категории и щелкнуть по кнопке Add Keyword.
3. Внесите следующие категории:
  - Resources Consumption (расход ресурсов);
  - Documentation (документация);
  - Information System (информационная система);
  - Quality Measure (мера измерения качества).
4. Создайте UDP. Для этого в поле "**User-Defined Property (UDP) to be added after selected property**" внесите имя UDP, например, «Quality». Затем выберите тип UDP из выпадающего списка Datatype - «Text List (Single selection)». Затем щелкните по кнопке Add.
5. Для UDP типа List необходимо задать список значений. В поле New Category/Member внесите значение «A-Terrific» и щелкните по кнопке Add Member. Затем внесите другие значения UDP Quality:
  - B-Good
  - C-OK
  - D-Poor
  - E-Awful.
6. Для включения UDP в категорию щелкните по UDP в списке, затем щелкните по категории в нижнем списке, затем щелкните по кнопке Update. По табл. 9 внесите UDP.

Таблица 9 Определение UDP

Наименование UDP	Тип	Члены	Категория
Application (приложения)	Text List (Multiple Selection)	COS- Customer Order System (модуль оформления заказов) ESS- Employee Sheduler System (модуль создания и контроля расписания выполнения работ) PIS- Parts and Inventory System (модуль учета комплектующих и оборудования) PTS- Procedures and Troubleshooting System (модуль процедур сборки и поиска неисправностей)	Information System
Screen	Command		Information System
Additional Documentation (дополнительная документация)	Command List	Winword.EXE sample1.doc Winword.EXE sample2.doc POWERPNT.EXE sample3.ppt	Documentation

Change History (история изменения)	Paragraph Text		Documentation
Electricity Consumption (расход электроэнергии)	Real Number		Resources Consumption

7. По таблице 10 внесите значения UDP для работ.

Таблица 10. Значения UDP для работ

Activity Name	Additional Documentatio n	Application	Change History	Electricity Consumption	Quality
Отслеживание расписания и управление сборкой и тес- тированием	Winword.EXE sample2.doc	COS-Customer Order System ESS- Employee Sheduler System	История изменения спецификаций	10,00	B-Good
Сборка настольных компьютеров		PIS- Parts and Inventory System PTS- Procedures and Troubleshooting		20,00	A-Terrific
Сборка ноутбуков		PIS- Parts and Inventory System PTS- Procedures and Troubleshooting		25,00	C-OK
Тестирование компьютеров		PIS- Parts and Inventory System PTS- Procedures and Troubleshooting System		40,00	B-Good

8. После внесения UDP типа Command или Command List, щелчок по кнопке > приведет к запуску приложения.
9. В диалоге Activity Properties щелкните по кнопке Categories. В появившемся диалоге Activity Categories Editor отключите категорию Information System. Щелкните по ОК. Посмотрите результат.
10. Свойства UDP можно присвоить не только работам, но и стрелкам. Щелкните по стрелке правой кнопкой и выберите в контекстном меню UDP Editor. По табл. 11 задайте значения UDP стрелкам.

Таблица 11 Значения UDP для стрелок

Arrow Name	Quality
Заказы на настольные компьютеры	B-Good
Ноутбуки	B-Good
Собранные компьютеры	A-Terrific

II. Посмотрите отчет по UDP. Меню **Tools/Reports/Diagram Object Report**.

12. Самостоятельно проделайте следующее.

Создайте еще два UDP (табл. 12).

Таблица 12 Новые UDP

Наименование UDP	Тип	Члены	Категория
Responsibility (ответственность)	Text List (Single Selection)	Ivanov Petrov Sidorov	Quality Measure

Customer Satisfaction (оценка клиента)	Integer List (Single Selection)	1 2 3 4	Quality Measure
--	---------------------------------	------------------	-----------------

Задайте свойства работам (табл. 13).

Таблица 13. Использование новых UDP

Activity Name	Responsibility	Customer Satisfaction
Отслеживание расписания и управление сборкой и тестированием	Ivanov	4
Сборка настольных компьютеров	Petrov	4
Сборка ноутбуков	Petrov	5
Тестирование компьютеров	Sidorov	4

### 3.4 Задание № 11. Создание модели TO-BE

Оценка бизнес-процессов модели AS-IS показала недостаточную эффективность деятельности компании Quill. В первую очередь это касается производственного отдела. Собираемые компьютеры не всегда пользуются достаточным спросом. Закупаемые компоненты часто чрезмерно дороги при посредственном качестве. Функциональность компьютеров не соответствует требованиям рынка.

В результате анализа компания принимает решение реорганизовать функции производства и тестирования компьютеров. Кроме того, принимается решение оставить функциональность «Продажи и маркетинг» и «Отгрузка и получение» пока без изменений.

Принято решение сформировать отдел дизайна, который должен формировать конфигурацию компьютеров, разрабатывать корпоративные стандарты, подбирать приемлемых поставщиков, разрабатывать инструкции по сборке, процедуры тестирования и устранения неполадок для всего производственного отдела.

Работа «Сборка и тестирование компьютеров» должна быть реорганизована и названа «Производство продукта».

Рассмотрим новые роли персонала.

Дизайнер должен:

- разрабатывать систему;
- разрабатывать стандарты на продукцию, документировать и передавать спецификации в отдел маркетинга и продаж;
- определять, какие компоненты (software и hardware) должны закупаться для сборки компьютеров;
- обеспечивать документацией и управлять процедурами сборки, тестирования и устранения неполадок.

Функции Диспетчера в работе «Сборка и тестирование компьютеров» должны быть заменены на функции Планировщика.

Планировщик должен:

- обрабатывать заказы клиентов и генерировать заказы на сборку;
- получать коммерческий прогноз из отдела маркетинга и формировать требования на закупку компонент;
- собирать информацию от поставщиков и нести ответственность за оформление заказов на поставку.

Диспетчер должен составлять расписание производства на основании заказов на сборку, полученных в результате работы «Планировать производство». Кроме того, Диспетчер должен получать копии заказов клиентов и отвечать за упаковку и комплектацию заказанных компьютеров, передаваемых в работу «Отгрузка и получение».

Лабораторная работа состоит из пяти частей:

1. Расщепление и модификация модели.
2. Слияние расщепленной модели с исходной моделью.
3. Использование Model Explorer для реорганизации дерева декомпозиции.
4. Модификация диаграммы IDEF3 «Собрать продукт» с целью отображения новой информации.
5. Добавление декомпозиции работы «Продажи и маркетинг».

#### Часть 1. Расщепление и модификация модели



- 1 Сохраните старую модель в новый файл и отредактируйте свойства модели:
  - Model Name: Предлагаемая модель компании Quill.
  - Time Frame: TO-BE.
  - Purpose: документировать предлагаемые изменения бизнес-процессов компании Quill.
- 2 Переименуйте работу «Сборка и тестирование компьютеров» в «Производство продукта». Расщепите эту работу в модель с тем же названием.
- 3 Модифицируйте отщепленную модель. Переместите работу «Тестирование компьютеров» с диаграммы A0 «Производство продукта» на диаграмму A2.1 «Сборка настольных компьютеров».
- 4 Переименуйте работу «Сборка настольных компьютеров» на диаграмме A0 в «Сборка продукта».
- 5 Удалите работу «Сборка ноутбуков».
- 6 Переименуйте стрелку «Заказы на настольные компьютеры» в «Заказы на изготовление».
- 7 Переименуйте «Отслеживание расписания и управление сборкой и тестированием» в «Планирование производства».
- 8 Создайте работу «Разработать конфигурацию».
- 9 Создайте ветвь стрелки «Персонал производственного отдела», назовите ее «Дизайнер», и направьте как механизм к работе «Разработать конфигурацию».
- 10 Создайте стрелку «Стандарты на продукцию» и направьте ее от выхода «Разработать конфигурацию» к границе диаграммы. Туннелируйте эту стрелку (Resolve Border Arrow). Создайте ветвь этой стрелки, идущую к управлению работы «Планирование производства» и назовите ее «Список необходимых компонент».
- 11 Удалите стрелку «Правила сборки и тестирования». Создайте ветвь стрелки «Стандарты на продукцию», идущую к управлению работы «Сборка продукта» и назовите ее «Правила сборки и тестирования».
- 12 Переименуйте стрелку «Диспетчер» в «Планировщик производства».
- 13 Добавьте стрелку «Прогноз продаж» как граничную управляющую к работе «Планирование производства».
- 14 Добавьте стрелку «Информация от поставщика» как граничную управляющую к работе «Планирование производства».
- 15 Добавьте стрелку «Заказ поставщику» как граничную стрелку выхода от работы «Планирование производства».
- 16 Туннелируйте эти стрелки (Resolve Border Arrow).
- 17 На диаграмме A-0 туннелируйте стрелку (Resolve Border Arrow) «Собранные компьютеры» и свяжите ее на диаграмме A0 с выходом работы «Сборка продукта».
- 18 Сохраните модифицированную модель (рис. 40).

Результат приведен на рисунках.

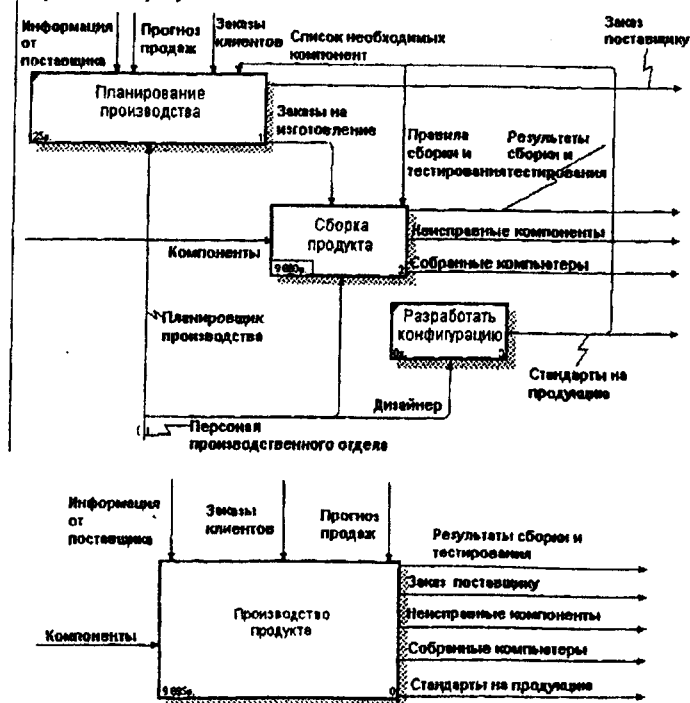


Рис. 40. Оптимизация модели AS-IS

## Часть 2. Слияние модели

1. Перейдите к работе «Производство продукта» в модели «Деятельность компании Quill». Щелкните правой кнопкой мыши по работе. В контекстном меню выберите Merge Model. В появившемся диалоге Continue with Merge? установите опцию Paste/Merge entire dictionaries, опцию Overwrite existing fields и щелкните по ОК. Модели должны слиться.
2. На диаграмме A0 туннелируйте стрелки (Resolve Border Arrow) «Информация от поставщика» и «Заказ поставщику».
3. Направьте стрелку «Прогноз продаж» с выхода «Продажи и маркетинг» на управление «Производство продукта».
4. Направьте стрелку «Стандарты на продукцию» с выхода «Производство продукта» на управление «Продажи и маркетинг».
5. Удалите ветвь стрелки управления «Правила и процедуры» работы «Производство продукта».
6. Закройте модель «Производство продукта».

## Часть 3. Использование Model Explorer для реорганизации дерева декомпозиции

Существуют причины, по которым работа «Разработать конфигурацию» должна быть на верхнем уровне, на диаграмме A0. Действительно, дизайнер разрабатывает стандарты на продукцию, включая правила сборки и тестирования и список необходимых для закупки компонент. Тем самым дизайнер управляет производством продукта в целом, кроме того, управляет работой «Продажи и маркетинг». Было бы логично перенести эту работу на уровень выше.

Используя возможности Model Explorer перенесите работу «Разработать конфигурацию» с диаграммы A2 «Производство продукта» на диаграмму A0.

Разрешите и перенаправьте стрелки, как показано на рис.41.



Рис. 41. Модель TO-BE

Рис. 41. Модель TO-BE

## Часть 4. Модификация диаграммы IDEF3 «Сборка продукта» с целью отображения новой информации

Так же как в модели AS-IS, сборка продукта включает сборку компонент и установку программного обеспечения. Однако теперь в работу «Сборка продукта» включена работа «Тестирование компьютера».

Тестирование начинается после окончания процесса сборки компьютера и окончания процесса установки программного обеспечения. Если компьютер неисправен, в процессе тестирования у него заменяют компоненты, информация о неисправных компонентах может быть направлена на работу «Подготовка компонент». Такая информация может помочь более тщательно подготавливать компоненты к сборке. Результатом процесса тестирования является заказанные компьютеры и неисправные компоненты.

Модифицируйте диаграмму IDEF3 «Сборка продукта» в соответствии с приведенной информацией (рис.

42).

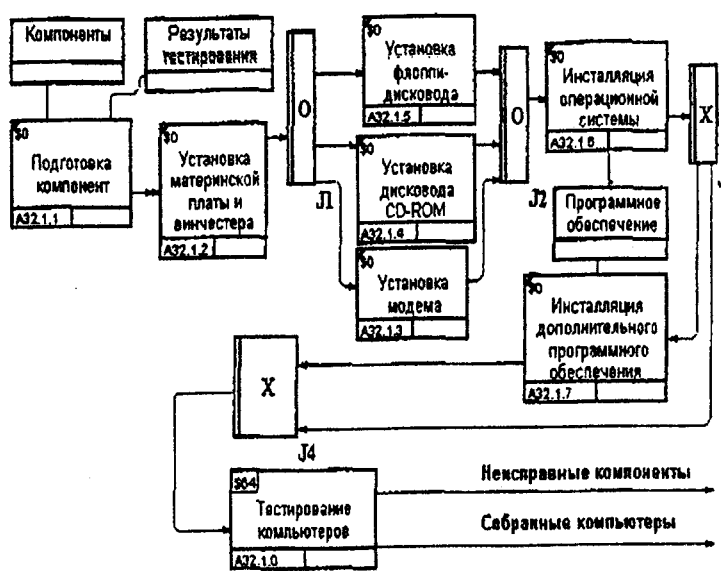


Рис. 42. Доработка диаграммы IDEF3

### Часть 5. Декомпозиция работы «Продажи и маркетинг»

Работа по продажам и маркетингу заключается в ответах на телефонные звонки клиентов, предоставлении клиентам информации о ценах, оформлении заказов, внесении заказов в информационную систему и исследовании рынка.

На основе этой информации декомпозируйте работу «Продажи и маркетинг» (IDEFO). Создайте следующие работы:

- Предоставление информации о ценах
- Оформление заказов
- Исследование рынка.

Результат декомпозиции представлен на рис. 43.

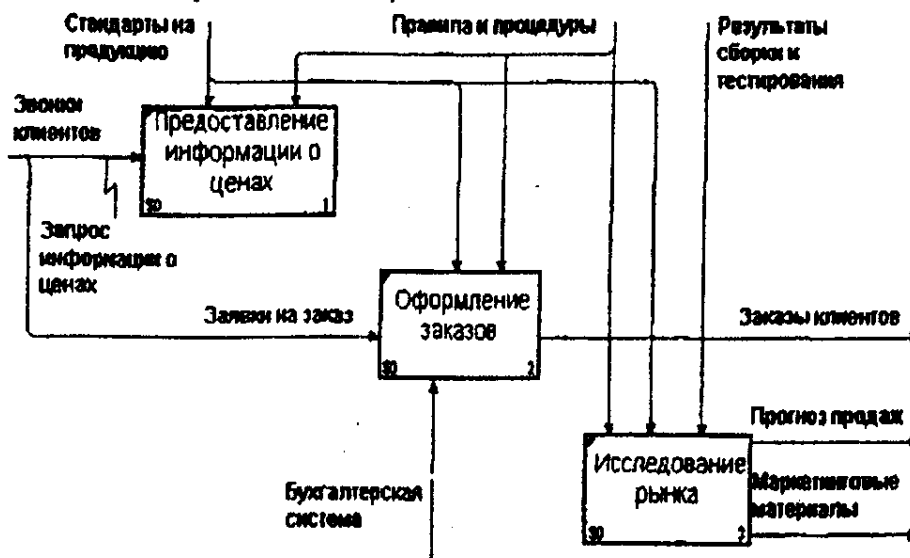


Рис.43. Декомпозиция работы «Продажи и маркетинг»

## 3.5 Задание № 12. Создание организационной диаграммы (Organization Chart)

1. Вызовите словарь групп ролей Role Group с помощью меню *Dictionary/Role Group...* и создайте необходимое

количество групп.

2. Вызовите словарь ролей Role с помощью меню *Dictionary/Role...* и опишите роли, связав их с требуемыми группами. В данном случае **роли** - это аналог **должностей**.
3. Вызовите словарь ресурсов «Resource» с помощью меню *Dictionary/Resource...* и впишите имена людей, связав их с группами ролей и ролями.
4. С помощью меню *Diagram/Add Organization Chart...* вызовите диалог «Organization Chart Wizard» и, следуя его указаниям, создайте первый уровень организационной иерархии на основании словарей групп ролей, ролей и ресурсов.
5. Далее, вызывая контекстное меню *Edit subordinate list...*, создайте следующий уровень иерархии по аналогии с пунктом 4.

## 4 ИНТЕГРАЦИЯ ФУНКЦИОНАЛЬНЫХ МОДЕЛЕЙ И МОДЕЛЕЙ ДАННЫХ

### 4.1 Задание № 13. Создание диаграммы потоков данных (Data Flow Diagramming)

В AllFusion Process Modeler реализована нотация Гейна-Сарсона для построения диаграмм потоков данных (DFD-диаграмм). Диаграмма потоков данных описывает:

- функции обработки информации (activity);
- документы и другие объекты, которые участвуют в обработке информации (arrow);
- внешние ссылки (external reference) на объекты, являющиеся внешними по отношению к модели;
- хранилища данных (data store).

Взаимосвязь указанных компонент имеет вид, показанный на рис.44.

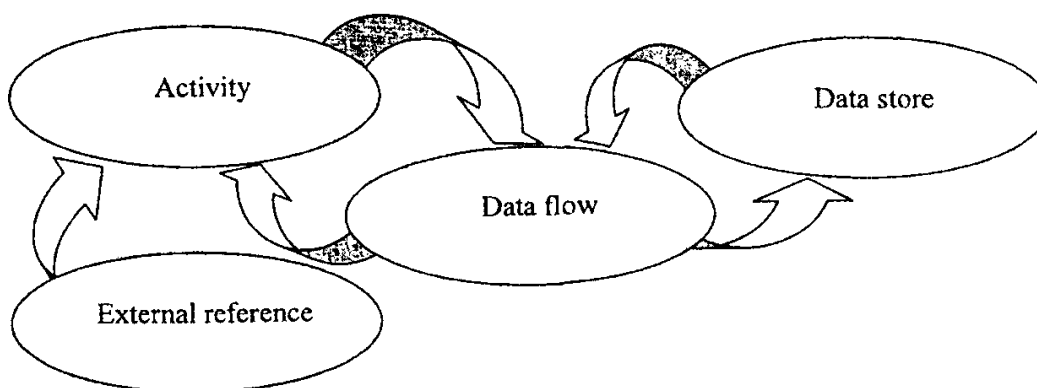


Рис. 44. Семантика DFD-диаграммы

Перед тем как приступить к созданию диаграмм DFD, необходимо выполнить декомпозицию работы «Продажи и маркетинг». Декомпозиция должна содержать как минимум две дочерние работы, связанные друг с другом:

- Оформление заказа
- Исследование рынка

При оформлении заказа важно проверить, существует ли такой клиент в нашей базе данных и если не существует, внести его в БД и затем оформить заказ. Оформление заказа начинается со звонка клиента. В процессе оформления заказа БД клиентов может просматриваться и редактироваться. Заказ должен включать как информацию о клиенте, так и информацию о заказанных продуктах. Оформление заказа подразумевает чтение и запись информации о прочих заказах.

В процессе декомпозиции согласно правилам DFD преобразуем граничные стрелки во внутренние, начинающиеся и заканчивающиеся на внешних ссылках.

1. Декомпозируйте работу «Оформление заказов».
2. В диалоге Activity Box Count выберите количество работ 2 и нотацию DFD.
3. Щелкните по ОК и внесите на новой диаграмме DFD имена работ:
  - Проверка и внесение клиента
  - Внесение заказа.
4. Внесите следующие хранилища данных:
  - Список клиентов
  - Список продуктов
  - Список заказов.
5. Удалите граничные стрелки с диаграммы DFD.
6. Внесите внешнюю ссылку «Звонки клиентов».
7. Создайте внутренние ссылки согласно рис. 45. При именовании стрелок используйте словарь.

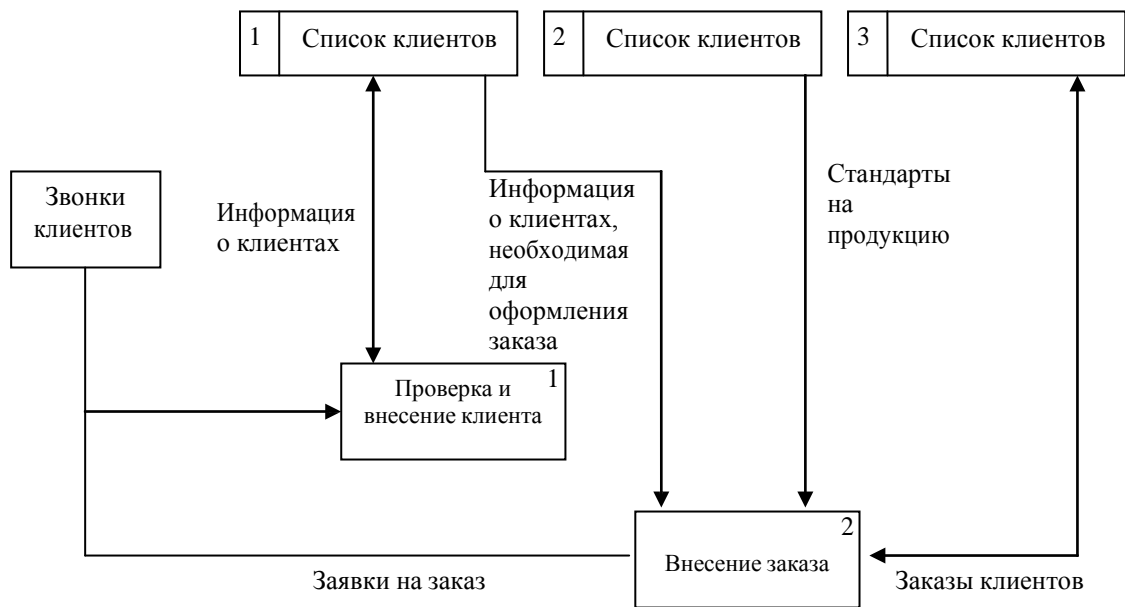


Рис.45. DFD-диаграмма

8. Обратите внимание, что стрелки «Информация о клиентах» и «Заказы клиентов» двунаправленные. Для того чтобы сделать стрелку двунаправленной, щелкните правой кнопкой по стрелке, выберите в контекстном меню пункт Style Editor и в диалоге Style Editor выберите опцию Bidirectional.
9. На родительской диаграмме туннелируйте (Change to Tunnel) стрелки, подходящие и исходящие из работы «Оформление заказов» (рис. 46).

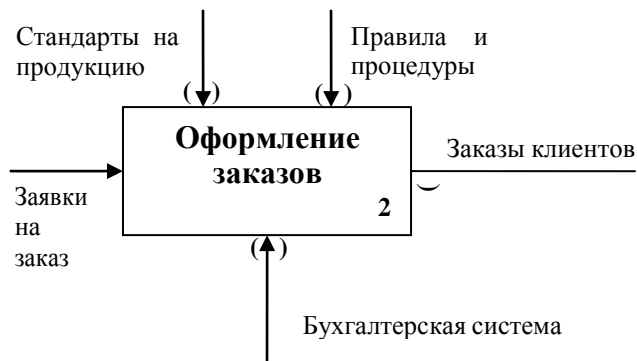


Рис.46. Туннелирование стрелок

Некоторые стрелки с диаграммы IDEF0 (не только с родительской) могут показываться на диаграмме DFD. Для отображения таких стрелок используется инструмент Off-Page Reference.

10. Самостоятельно декомпозируйте работу «Исследование рынка» на диаграмму DFD. Создайте следующие работы: «Разработка прогнозов продаж», «Разработка маркетинговых материалов», «Привлечение новых клиентов».
11. Внесите хранилища данных:
  - Список клиентов
  - Список продуктов
  - Список заказов.
12. Добавьте 2 внешние ссылки:
  - Маркетинговые материалы
  - Прогноз продаж.
13. Свяжите объекты диаграммы DFD стрелками.

## 4.2 Задание № 14. Создание сущностей и их экспорт в AllFusion ERwin Data Modeler

1. Перейдите в меню **Model/Entity/Attribute Editor...**
2. В поле Entity диалога **Entity and Attribute Editor** внесите имя сущности SALES FORECAST, щелкните по Add.
3. В поле Attribute диалога Entity and Attribute Editor внесите имя атрибутов и щелкните по Add:
  - Product Code
  - Sales Forecast End Date
  - Sales Forecast Product Quantity
  - Sales Forecast Start Date
4. Аналогичным образом создайте следующие сущности с атрибутами:
  - CUSTOMER:
    - > Customer Number;
    - > Customer First Name;
    - > Customer Last Name;
    - > Customer Company Name;
    - > Customer Address;
    - > Customer City;
    - > Customer State;
    - > Customer Zip Code;
    - > Customer Phone Number;
    - > Customer Fax Number. SALES ORDER
  - > Sales Order Number;
  - > Sales Order DateTime;
  - > Sales Order Shipment Charge;
  - > Sales Order Total;
  - > Sales Order Shipment DateTime;
  - > Sales Order Status. LINE ITEM
  - > Line Item Sequence Number;
  - > Line Item Quantity;
  - > Line Item Total.
5. Экспортируйте данные в ERwin. Перейдите в меню File/Export ERwin (BPX) и укажите имя файла экспорта.
6. Перейдите в ERwin. Выберите пункт меню File/BPwin Import 1. В диалоге Open File укажите имя файла импорта (\*.bpx).