

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

УТВЕРЖДАЮ

Зав. каф. АОИ, д.т.н., проф.

_____ Ю.П. Ехлаков

" ____ " _____ 2014 г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ
по дисциплине
"МОДЕЛИРОВАНИЕ И АНАЛИЗ БИЗНЕС-ПРОЦЕССОВ"**

для студентов направления подготовки
231000.62 «Программная инженерия» (бакалавриат)

Часть I

Разработчик:
профессор каф. АОИ, д.т.н.

_____ М.П. Силич

СОДЕРЖАНИЕ

Введение	3
Лабораторная работа №1 «Создание IDEF0-модели бизнес-процесса»	4
Лабораторная работа №2 «Создание IDEF3-модели бизнес-процесса»	20
Лабораторная работа №3 «Создание DFD-модели бизнес-процесса»	30
Лабораторная работа №4 «Создание прецедентной UML-модели бизнес-процесса»	35
Лабораторная работа №5 «Создание объектной UML-модели бизнес-процесса»	48
Лабораторная работа №6 «Создание UML- модели информационной системы».....	58
Литература	64
Приложение 1. Варианты индивидуального задания	65

ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для подготовки и выполнения лабораторных работ, включенных в первую часть двухсеместрового курса по дисциплине «Моделирование и анализ бизнес-процессов».

Лабораторные работы по данной части дисциплины имеют **целью**: закрепление теоретического материала, получение навыков самостоятельного моделирования бизнес-процессов по стандартам DFD, IDEFO, IDEF3, UML с помощью CASE-средств структурного и объектно-ориентированного анализа BPWin и Rational Rose.

Выполнение лабораторных работ направлено на формирование следующих **компетенций**:

- способность к формализации в своей предметной области с учетом ограничений используемых методов исследования (ПК-2);
- способность формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта (ПК-6).

Лабораторные работы выполняются индивидуально. Лабораторная работа выполняется в соответствии с порядком, описанном в методических указаниях.

Форма контроля выполнения лабораторной работы: демонстрация преподавателю построенных моделей, собеседование, ответы на вопросы, выполнение дополнительных заданий.

Лабораторная работа №1 «Создание IDEF0-модели бизнес-процесса»

Цель работы: Ознакомиться с основами методологии IDEF0 и основами работы с пакетом BPWin. Получить практические навыки в построении IDEF0-модели бизнес-процесса средствами пакета BPWin.

Порядок выполнения работы.

1. Выбор задания.

Выберите бизнес-процесс, для которого будете формировать модель. Вы можете выбрать один из вариантов процессов, описанных в приложении, или предложить свой вариант.

2. Знакомство с основами методологии IDEF0.

Методология IDEF0 (Integrated DEFinition) представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели предметной области. Функциональная модель IDEF0 отображает функциональную структуру системы, т.е. производимые ею действия и связи между этими действиями. Данная методология применяется при создании новых систем для определения требований и функций и затем для разработки системы, удовлетворяющей требованиям и реализующей функции. Для действующих систем эта методология может использоваться для анализа функций, выполняемых системой, а также для наглядного представления «механизмов», посредством которых эти функции осуществляются. Основной сферой применения методологии IDEF0 является предпроектное обследование и анализ системы.

Методология IDEF0 основана на методе SADT Росса. В рамках проекта ICAM, организованного военными ведомствами США с целью разработки подходов, обеспечивающих повышение эффективности проектирования благодаря систематическому внедрению компьютерных технологий, метод SADT и некоторые аспекты его применения были стандартизированы, после чего получили название методологии IDEF0.

Модель IDEF0 представляет собой набор диаграмм с поддерживающей их документацией, включающей сопровождающие тексты и словарь. Диаграммы модели декомпозируют сложную систему на составные части. Первоначальная (исходная, корневая) диаграмма является наиболее общим и наиболее абстрактным описанием всей системы в целом. Она показывает основную функциональную составляющую системы в виде одного блока. Взаимодействие с окружающим миром описывается в терминах входа (данные или объекты, потребляемые или изменяемые функцией), выхода (основной результат деятельности функции, конечный продукт), управления (стратегии и процедуры, которыми руководствуется функция) и механизмов (необходимые ресурсы). Кроме того, при создании контекстной диаграммы формулируются цель моделирования, область и точка зрения, с которой будет строиться модель.

Затем общая функция разбивается на крупные подфункции. Связи подфункций отображаются на отдельной диаграмме – диаграмме декомпозиции первого уровня. В свою очередь, каждая подфункция может быть декомпозирована на более мелкие подфункции, связи между которыми также отображаются на отдельных диаграммах декомпозиции второго уровня. И так далее до достижения необ-

ходимой детализации описания. Таким образом, модель представляет собой совокупность иерархически выстроенных диаграмм, каждая из которых является детальным описанием какой-либо вышестоящей (родительской) функции или работы (activity).

Каждая диаграмма включает в себя один или несколько функциональных блоков, представляющих какую-либо функцию или работу и изображаемых в виде прямоугольников. Стрелки обозначают объекты или информацию, связывающую работы между собой и с внешним миром. Каждая из стрелок имеет метку, характеризующую ее. Назначение стрелок зависит от стороны блока, в которую стрелка входит или выходит: стрелки, входящие с левой стороны блока, представляют собой предметы или информацию, необходимые для выполнения функции; выходящие из правой стороны блока показывают предметы или данные, полученные в результате выполнения функции; входящие сверху – условия или данные, которые управляют выполнением функции; входящие снизу – механизм, выполняющий функцию (например, инструмент, оборудование или человек).

Стрелки на диаграмме IDEF0 означают ограничения, задаваемые связанными с ними объектами (предметами или данными). Они не представляют собой поток или последовательность. Соединяя выход одного блока с входом другого, они показывают ограничения. Блок, получающий объекты, «ограничен» в том смысле, что функция не может быть выполнена, пока не будут получены объекты, производимые другими блоками. Стрелки, входящие в блок, показывают все объекты, которые необходимы для выполнения функции.

Несколько функций на диаграмме могут выполняться одновременно, если удовлетворены все ограничивающие условия. Ни последовательность, ни время не являются точно определенными в IDEF0. Отсюда следует, что IDEF0-модели – это ни блок-схемы, ни просто диаграммы потоков данных, а предписывающие диаграммы, которые представляют входные/выходные преобразования, а также указывают правила этих преобразований.

3. Знакомство с инструментальным средством BPwin

BPwin является мощным средством моделирования и документирования бизнес-процессов. Этот продукт использует технологию моделирования IDEF0 — наиболее распространенный стандарт, принятый для моделирования бизнес-процессов. Кроме того, BPwin поддерживает методологии моделирования DFD и IDEF3. Методология DFD используется для описания потоков данных, которые возникают в результате деятельности компании. Методология IDEF3 служит для графического описания потока процессов (работ), взаимодействия процессов и объектов.

Функциональность BPwin заключается не только в рисовании диаграмм, но и в проверке целостности и согласованности модели. BPwin обеспечивает логическую четкость в определении и описании элементов диаграмм, а также проверку целостности связей между диаграммами. Инструмент обеспечивает коррекцию наиболее часто встречающихся ошибок при моделировании, таких как «зависание» связей при переходе от диаграммы к диаграмме, нарушение ассоциации связей в различных диаграммах модели и т. п.

Для оценки моделируемых бизнес-процессов BPwin предоставляет разработчику два инструмента — функционально-стоимостной анализ (ABC) и оценку свойств, определяемых пользователем (User Defined Properties, UDP).

BPwin имеет достаточно простой и интуитивно понятный интерфейс пользователя. Общий вид рабочего интерфейса представлен на рис. 1.1. Его основными элементами являются: главное меню, основная панель инструментов (ниже главного меню), навигатор модели (в левой части рабочего интерфейса), окно модели (в правой части рабочего интерфейса) и специальная панель инструментов, вид которой зависит от выбранного типа модели (располагается между навигатором и окном модели). Данное расположение элементов интерфейса принято по умолчанию, однако оно может быть изменено пользователем.

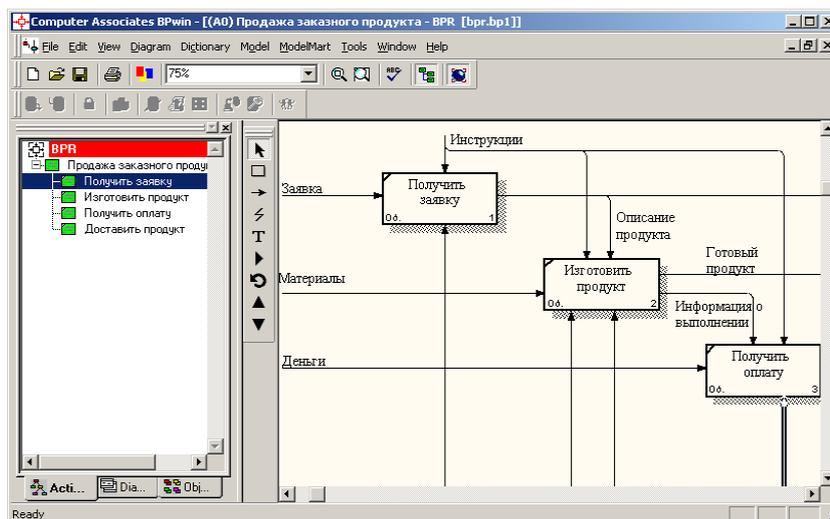


Рис. 1.1. Рабочий интерфейс среды моделирования BPwin

Инструмент навигации Model Explorer имеет три вкладки - Activities, Diagrams и Objects. Вкладка Activities (рис. 1.1) показывает в виде раскрывающегося иерархического списка все блоки (работы, функции), включенные в диаграммы модели. Одновременно могут быть показаны все модели, открытые в BPwin. Блоки с диаграмм IDEF0 показываются зеленым цветом, IDEF3 – желтым и DFD – голубым.

Щелчок по наименованию блока во вкладке Activity переключает окно модели на диаграмму, на которой этот блок размещен. Если с помощью вкладки Activities можно перейти на стандартные диаграммы (контекстную и декомпозиции), то вкладка Diagrams служит для перехода на любую диаграмму модели. После перехода на вкладку Objects на ней показываются все объекты, соответствующие выбранной на вкладке Diagrams диаграмме, в том числе работы, хранилища данных, внешние ссылки, объекты ссылок и перекрестки.

4. Создание контекстной диаграммы

Построение модели в нотации IDEF0 всегда начинается с построения контекстной диаграммы. При создании модели бизнес-процессов контекстная диаграмма содержит единственную функцию, которая определяет модель в целом. Далее можно проводить декомпозицию диаграммы, где будут содержаться функции, связи между ними и свойства. Контекстная диаграмма показывает высокоуровневую функцию модели для дальнейшего исследования в соответствии с целью, границей и точкой зрения.

Для создания модели IDEF0 необходимо выбрать в системном меню пункт File/New. После этого появится диалоговое окно, изображенное на рис. 1.2.

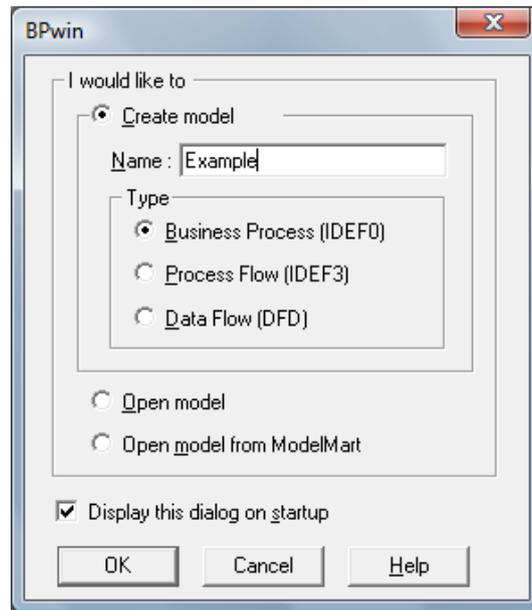


Рис. 1.2. Диалог создания контекстной диаграммы

В данном диалоговом окне необходимо дать имя модели. В нашем случае это Example. После нажатия на кнопку Ok появляется диалоговое окно Properties, изображенное на рисунке 1.3.

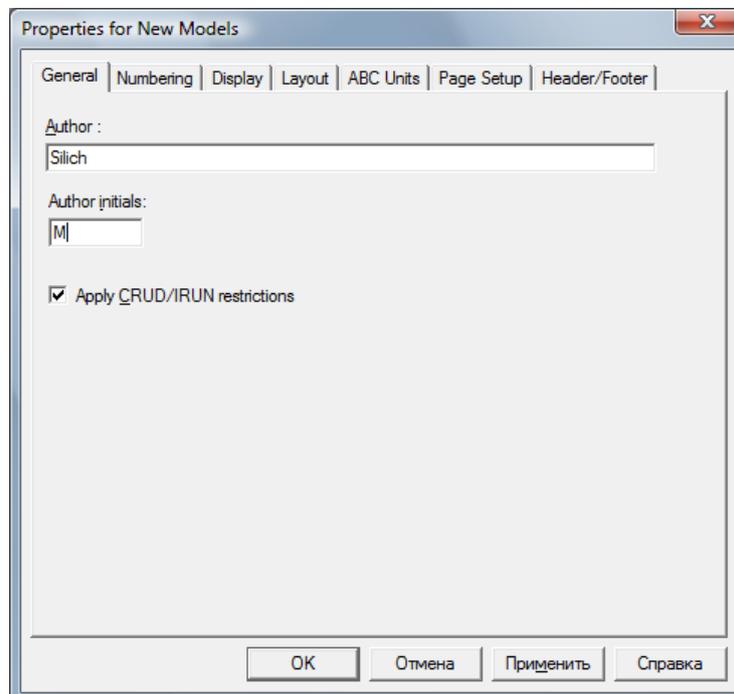


Рис. 1.3. Диалог свойств новой модели

По умолчанию в данном диалоге отображается вкладка General. Здесь в поле Author необходимо набрать имя автора и нажать Ok.

Откроется окно, отображающее контекстную диаграмму A-0 с одним функциональным блоком A0 (см. рис. 1.4).

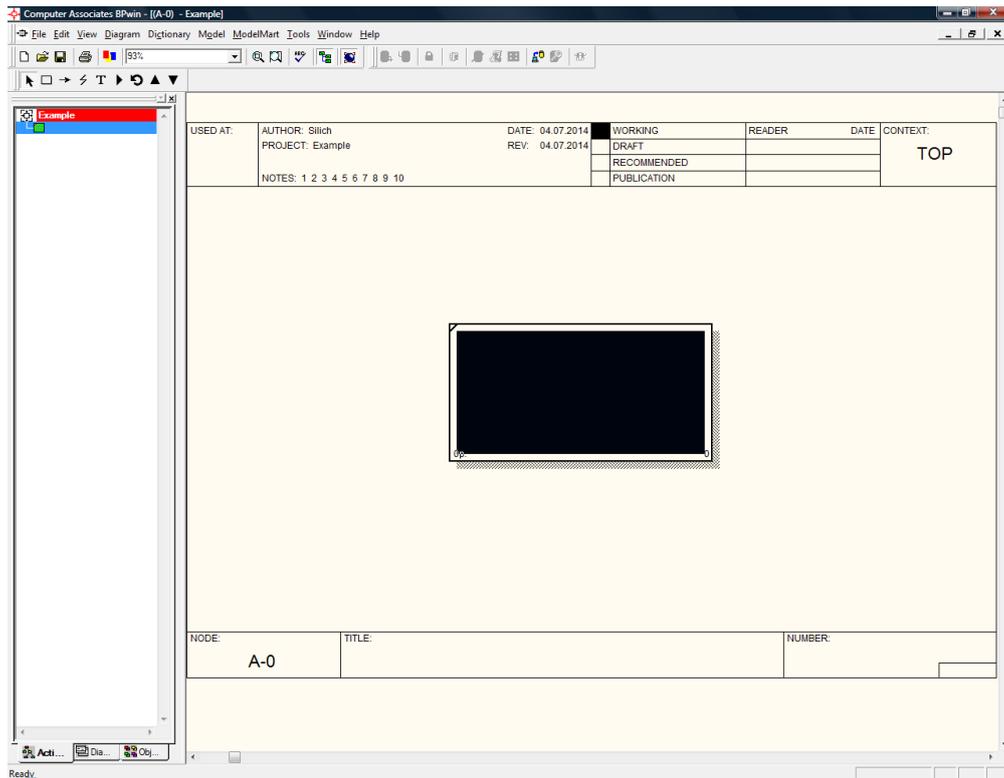


Рис. 1.4. Окно, отображающее контекстную диаграмму

Появившийся функциональный блок представляет собой весь моделируемый процесс в целом. Его необходимо именовать: нажмите правую кнопку мыши, выберите в контекстном меню пункт Name и в появившемся диалоге введите имя (см. рис. 1.5). Имя блока должно быть выражено отглагольным существительным, обозначающим действие (например, "Изготовление детали", "Прием заказа" и т. д.).

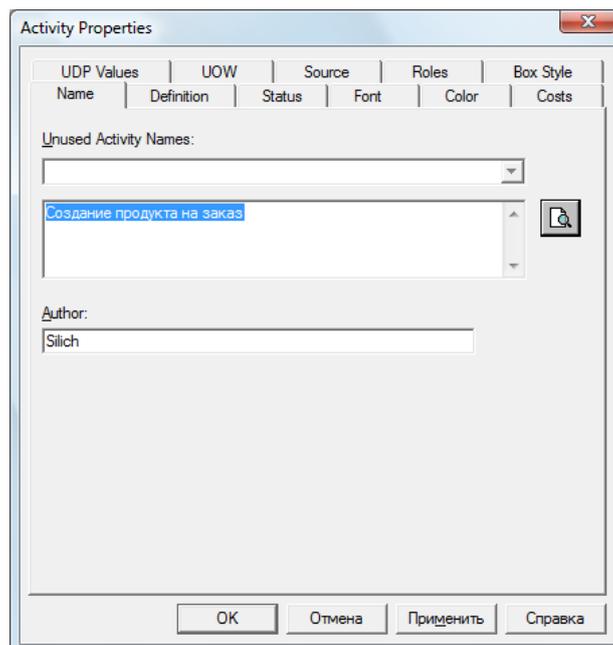


Рис. 1.5. Ввод наименования контекстного блока.

Для описания цели и точки зрения можно использовать два подхода – путем вставки текстового блока на диаграмму или путем определения свойств модели. Для вставки текстового блока необходимо использовать инструмент **T** который производит вставку текстового блока на диаграмму. Пример диаграммы, содержащей текстовый блок, изображен на рисунке 1.6.

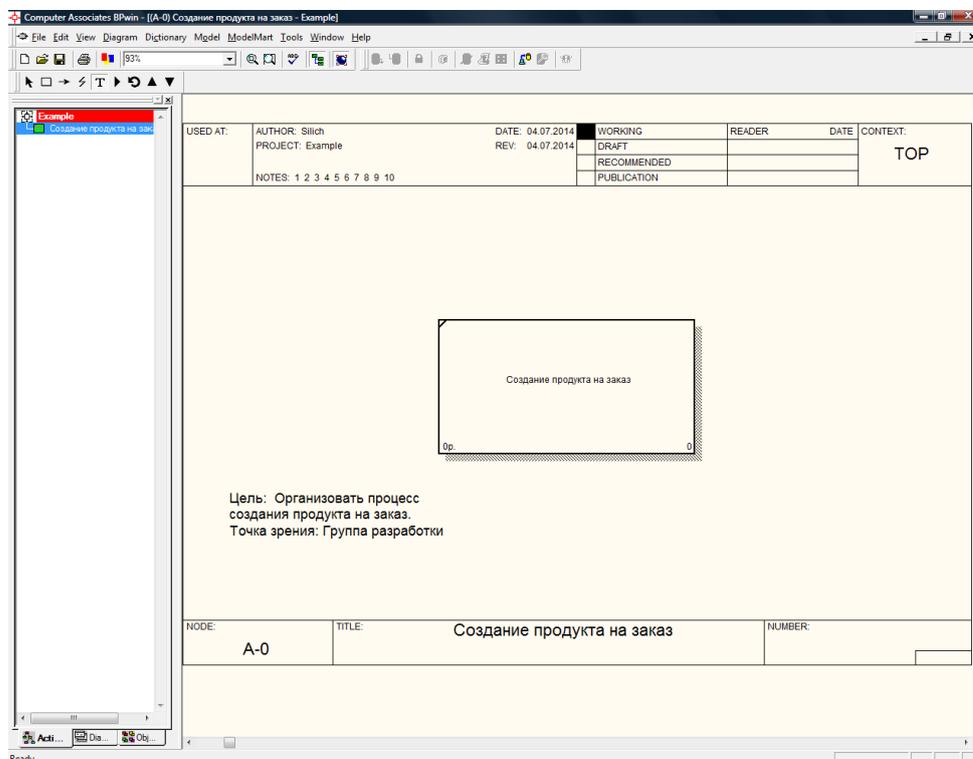


Рис 1.6. Контекстная диаграмма с текстовым блоком

Для определения свойств модели необходимо посредством меню Model / Model properties вызвать диалог Model properties и выбрать вкладку Purpose (см. рис. 1.7).

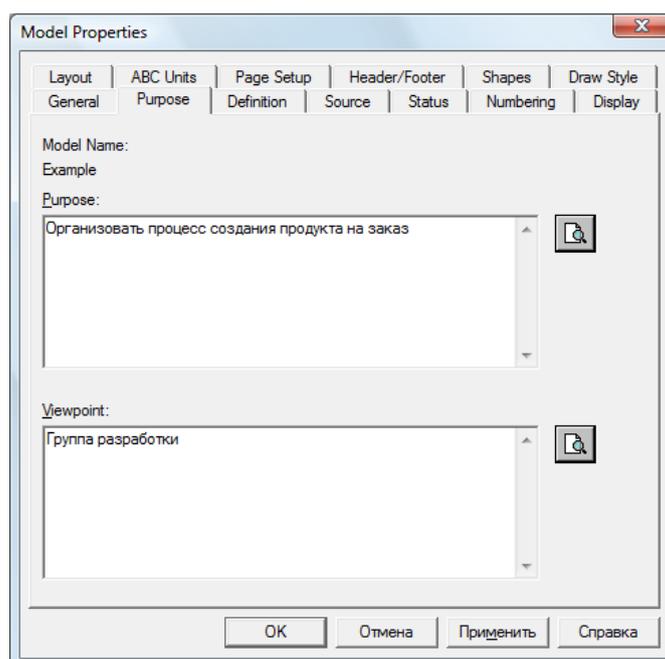


Рис 1.7. Диалог определения цели модели

5. Создание граничных стрелок

Взаимодействие блока с внешним миром описывается в виде стрелок. Стрелки представляют собой некие предметы или данные и именуются существительными (например, "Деталь", "Изделие", "Заказ").

В IDEF0 различают четыре типа стрелок – вход, выход, управление и механизм. Выход и вход показывают, что и из чего делается функцией, управление показывает, как и почему это делается, а механизм показывает, кем и с помощью чего это делается. Каждый тип стрелок подходит к определенной стороне прямоугольника, изображающего функцию, или выходит из нее (рис. 1.8).

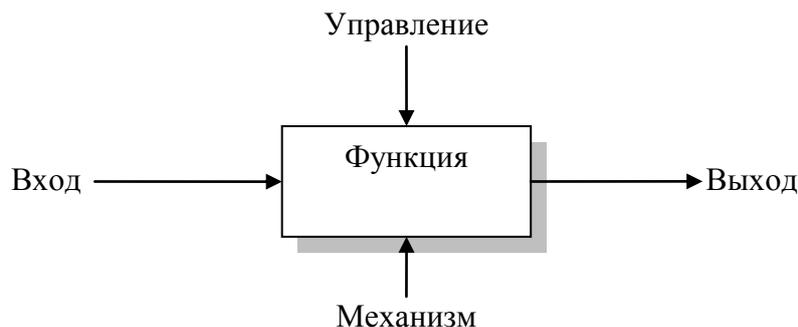


Рис. 1.8. Функциональный блок и входящие/выходящие стрелки

Вход (Input) – материал или информация, которые используются или преобразуются функциональным блоком для получения результата (выхода). Стрелка входа рисуется как входящая в **левую** грань блока. Примеры входов: *сырье, материалы, комплектующие, исходные данные, заявка клиента*.

Выход (Output) – предметы или информация, которые производятся блоком. Стрелка выхода рисуется как исходящая из **правой** грани блока. Каждый блок должен иметь хотя бы одну стрелку выхода. Функция без результата не имеет смысла и не должна моделироваться. Функции преобразуют объекты слева направо (от входа к выходу). Таким образом, блок представляет собой переход от состояния "до" к состоянию "после". Примеры выходов: *продукция, изготовленное изделие, выполненная услуга, подготовленный документ*.

Управление (Control) – условия или данные, которые управляют выполнением функции (правила или стандарты, которыми руководствуются при ее выполнении). Стрелка управления рисуется как входящая в **верхнюю** грань блока. Очень часто сложно определить, являются ли данные входом или управлением. В этом случае подсказкой может служить то, перерабатываются/изменяются ли данные в блоке или нет. Если изменяются, то скорее всего это вход, если нет – управление. Управление влияет на функцию, но не преобразуется ею. Примеры управления: *инструкции, требования, стандарты, правила, указания, задания, план, проект, чертеж*.

Механизм (Mechanism) – ресурсы, которые выполняют работу, например персонал предприятия, станки, устройства и т. д. Стрелка механизма рисуется как входящая в **нижнюю** грань блока. Примеры механизма: *персонал, фирма, отдел приема заявок, отдел разработки, цех, оператор, станок, аппаратура, инструмент, компьютер, информационная система, программное обеспечение*.

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

Для создания стрелок, связанных с функциональным блоком, необходимо использовать инструмент  на панели инструментов. Щелкните левой кнопкой мыши на левой границе диаграммы, отпустите кнопку мыши и переместите появившийся курсор  на левую сторону функционального блока. Большой подсвеченный треугольник показывает ту сторону функционального блока, которая будет к нему привязана (см. рис. 1.9).

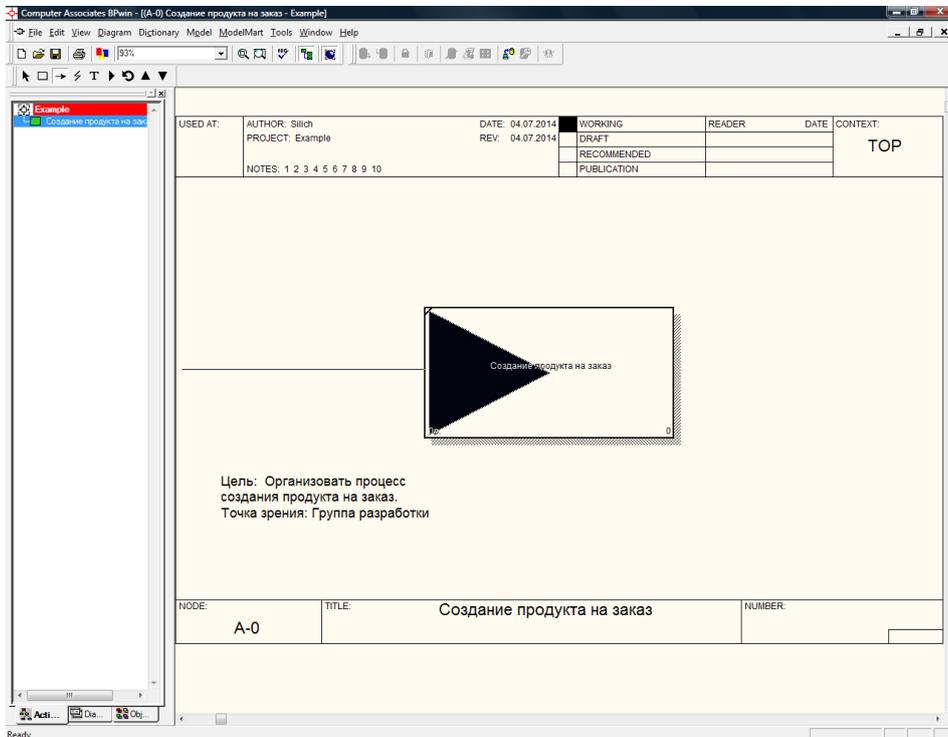


Рис. 1.9. Привязка стрелок к функциональному блоку

Чтобы именовать стрелку необходимо навести указатель мыши на нее и нажать правую кнопку. Появится контекстное меню, в котором необходимо выбрать пункт Name. Далее в диалоге Arrow Properties введите имя стрелки (см. рис. 1.10).

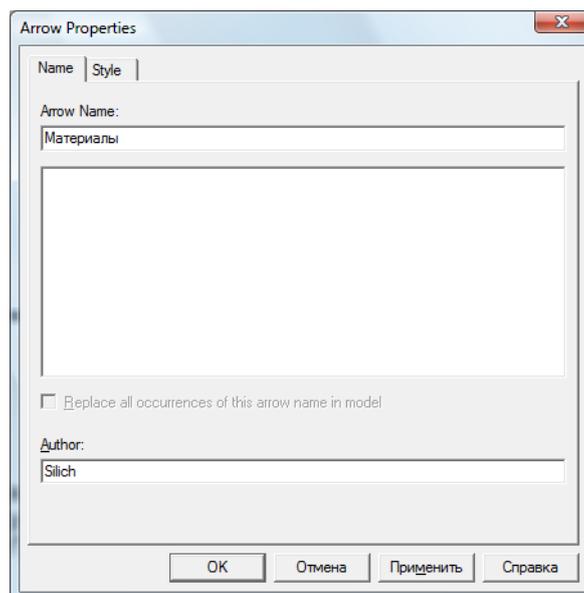


Рис. 1.10. Диалог определения свойств стрелки

Таким же образом можно определять стрелки всех типов, которые соответствуют входу, выходу, механизму и управлению. На рисунке 1.11 приведен пример законченной контекстной диаграммы.

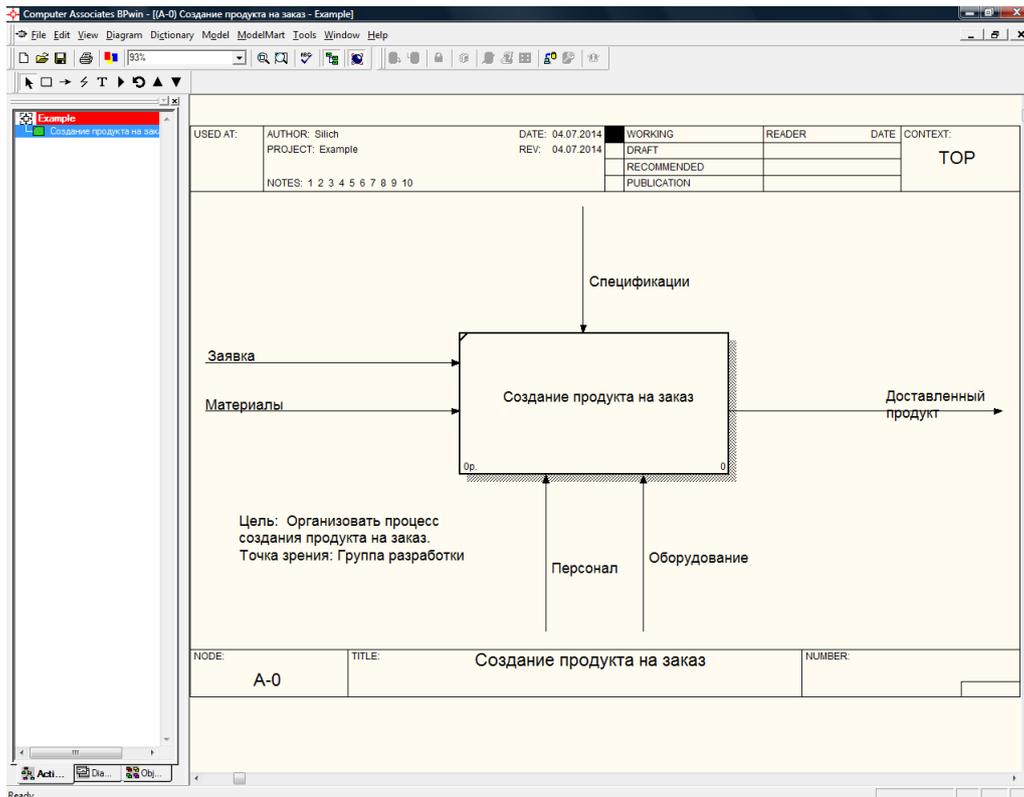


Рис. 1.11. Пример контекстной диаграммы

6. Создание диаграммы декомпозиции

Декомпозиционная диаграмма используется для разбиения функционального блока на образующие его составные части. Например, блок «Изготовление заказного продукта» может быть декомпозирован на такие части как: «Получить заказ клиента», «Выполнить заказ», «Получить оплату заказа», «Отправить заказ клиенту». Каждый из этих блоков в зависимости от требуемой глубины детализации также может быть представлен в виде декомпозиционной диаграммы. Причем любой функциональный блок можно представить в виде IDEF0-, IDEF3- или DFD- диаграммы.

Для декомпозиции необходимо сделать следующее. Выбрать тот блок, который необходимо декомпозировать. В нашем случае это единственный блок на контекстной диаграмме, затем выбрать инструмент  на панели инструментов. После чего появится диалог, изображенный на рисунке 1.12.

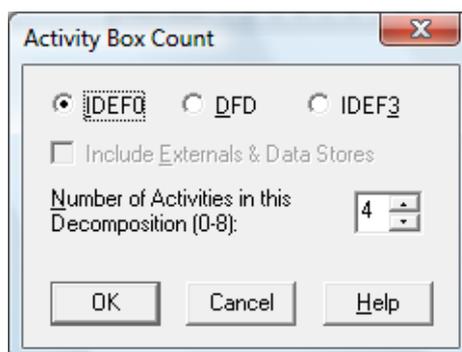


Рис 1.12. Диалог декомпозиции функционального блока

По умолчанию декомпозиционная диаграмма составляется в нотации IDEF0, причем количество функциональных блоков по умолчанию равно 4. Для обеспечения наглядности и лучшего понимания моделируемых процессов рекомендуется использовать от 3 до 6 блоков на одной диаграмме. При нажатии Ok в этом диалоге создается диаграмма A0 с указанным количеством функциональных блоков (см. рис. 1.13).

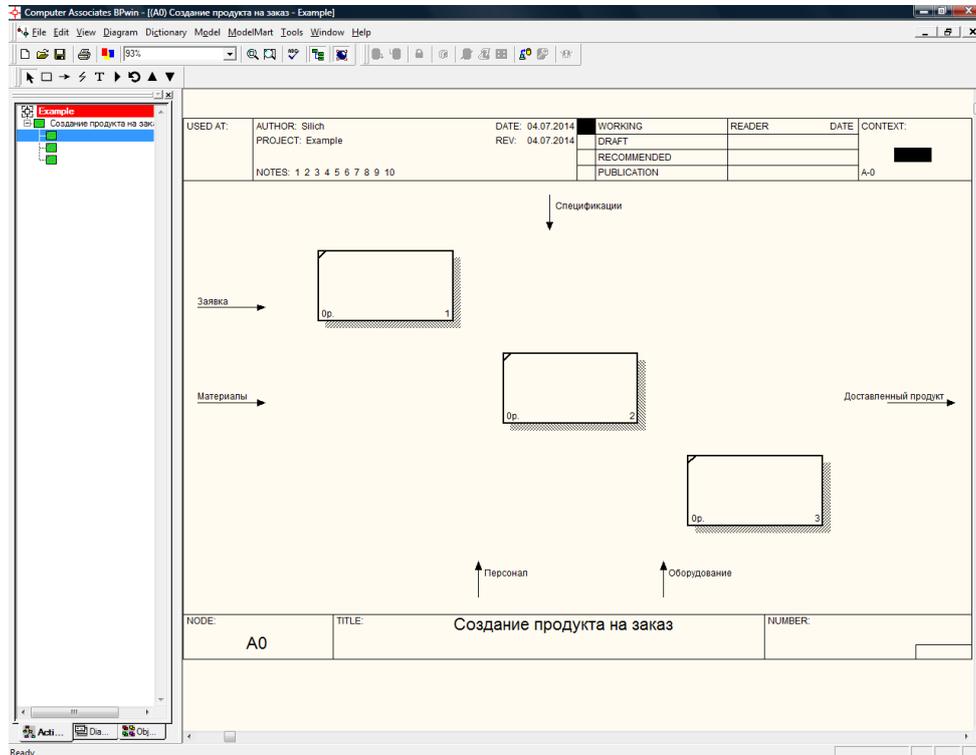


Рис 1.13. Шаблон декомпозиционной диаграммы

Если оказывается, что количество блоков недостаточно, то можно добавить блок, щелкнув сначала по кнопке с символом блока на палитре инструментов, а затем по свободному месту на диаграмме.

Блоки на диаграммах декомпозиции обычно располагаются по диагонали от левого верхнего угла к правому нижнему. Такой порядок называется порядком доминирования. Согласно этому принципу в левом верхнем углу располагается самая важная функция или функция, выполняемая по времени первой. Далее вправо вниз располагаются менее важные или выполняемые позже функции. Такое расположение облегчает чтение диаграмм. На диаграмме декомпозиции функции нумеруются автоматически слева направо. Номер блока показывается в правом нижнем углу. В левом верхнем углу изображается небольшая диагональная черта, которая показывает, что данная работа не была декомпозирована.

Контекстная (корневая) функция имеет номер 0. Блоки декомпозиции имеют номера 1, 2, 3 и т.д. Блоки декомпозиции нижнего уровня имеют номер родительского блока и очередной порядковый номер, например блоки декомпозиции функции 3 будут иметь номера 31, 32, 33, 34 и т. д. Диаграммы имеют номера по родительскому блоку. Контекстная диаграмма всегда имеет номер A-0, декомпозиция контекстной диаграммы – номер A0, остальные диаграммы декомпозиции – номера по соответствующему блоку (например, A1, A2, A21, A213 и т.д.). Имеются незначительные варианты нумерации, которые можно настроить во вкладке Numbering диалога Model Properties (меню Model /Model Properties).

Поименуйте каждый функциональный блок на диаграмме декомпозиции нажатием правой кнопки мыши на этом блоке и выбором пункта Name в появившемся контекстном меню. При этом диаграмма будет иметь вид, изображенный на рисунке 1.14.

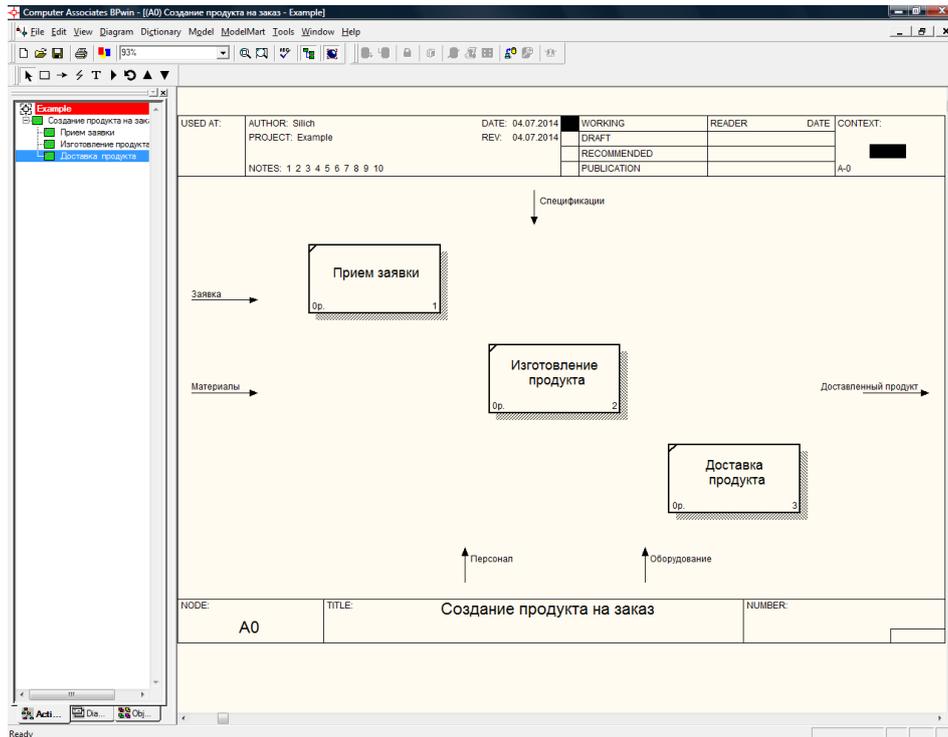


Рис 1.14. Именованые блоки декомпозиционной диаграммы

7. Рисование граничных стрелок

Помимо функциональных блоков на декомпозиционную диаграмму помещаются граничные стрелки, связанные с контекстной диаграммой. Это делается автоматически. Ведь функции нижнего уровня – это то же самое, что и функция верхнего уровня, но в более детальном изложении. Как следствие этого границы функционального блока верхнего уровня – это то же самое, что и границы диаграммы декомпозиции. Стрелки могут иметь ICOM-коды (аббревиатура от Input, Control, Output и Mechanism). Такой код содержит префикс, соответствующий типу стрелки (I, C, O или M) и порядковый номер. Для отображения ICOM-кодов следует включить опцию ICOM codes на вкладке Display диалога Model Properties (меню Model/Model Properties).

Граничные стрелки, автоматически перенесенные с родительской диаграммы, нужно связать с функциональными блоками. Для связывания стрелок входа, управления или механизма необходимо перейти в режим редактирования стрелок (выбрать инструмент ) , щелкнуть по стрелке и щелкнуть по соответствующей стороне блока (левой, если это вход, верхней – если управление, нижней – если это механизм). Для связывания стрелки выхода необходимо щелкнуть по правой стороне блока и затем по стрелке. Пример связывания граничной стрелки с блоком приведен на рисунке 1.15. При необходимости стрелку можно «перенаправить» на другой функциональный блок в пределах диаграммы. Для этого необходимо навести указатель мыши на соответствующую стрелку, и, удерживая кнопку мыши, перетащить стрелку.

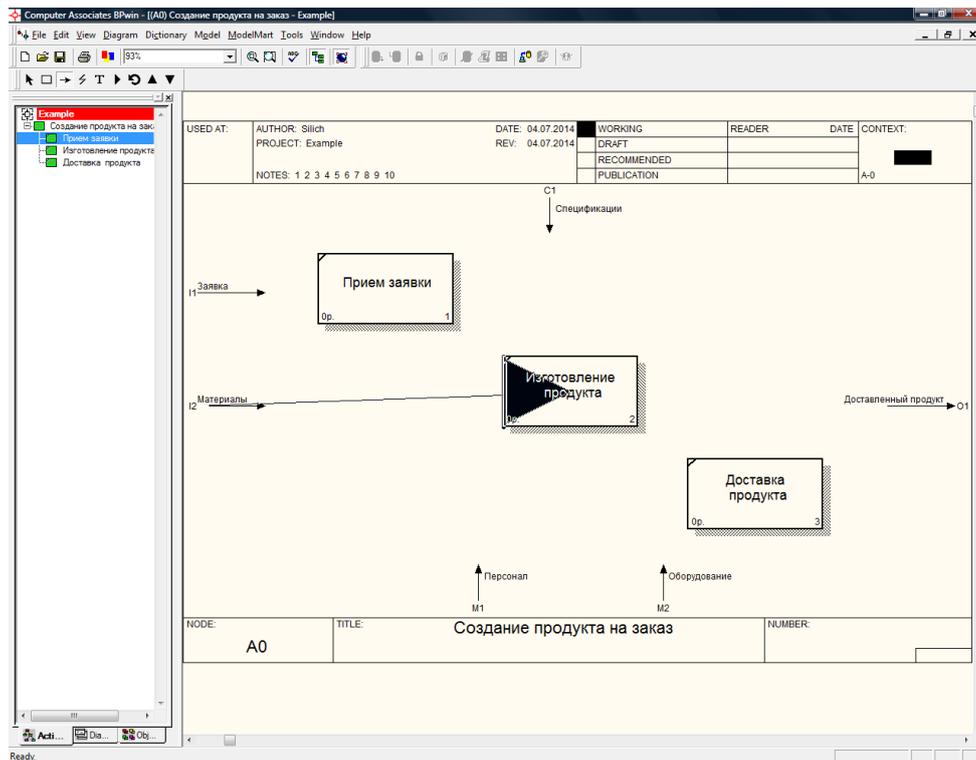


Рис 1.15. Соединение граничных стрелок на декомпозиционной диаграмме

Может возникнуть потребность добавить на диаграмму декомпозиции связь с окружением, которой не было на родительской диаграмме. В этом случае не нужно переходить на родительскую диаграмму, можно добавить граничную стрелку сразу на декомпозиционной диаграмме. Такая стрелка будет иметь знак туннелирования в виде квадратных скобок возле начала стрелки. Это означает, что она не перенесена с диаграммы верхнего уровня. Например, на декомпозиционную диаграмму мы добавили стрелку «Инструкции» (рис. 1.16).

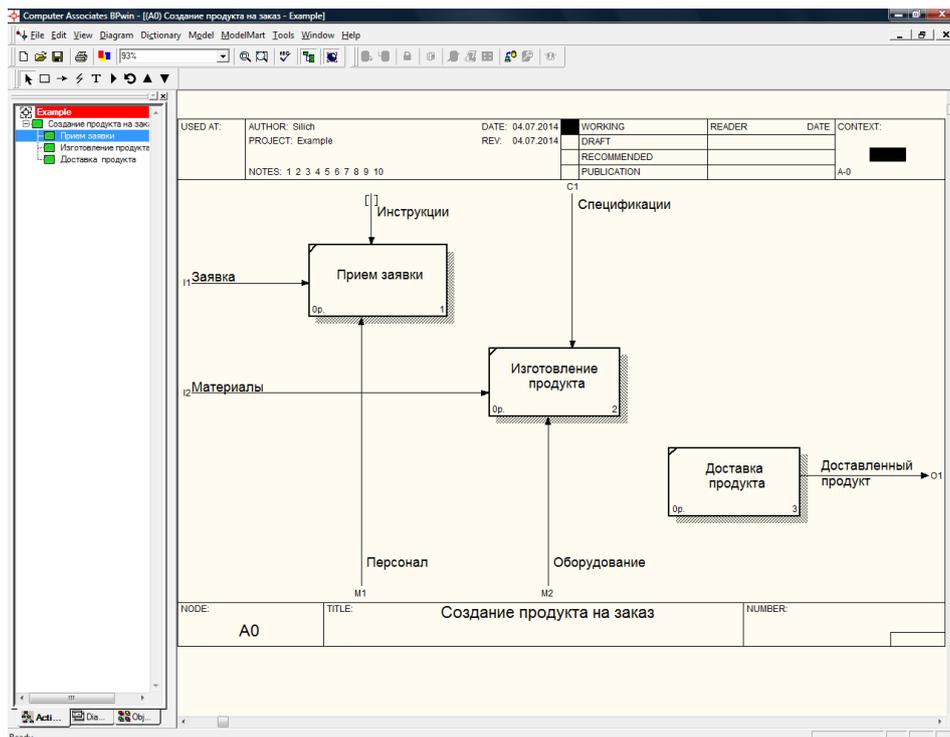


Рис. 1.16. Декомпозиционная диаграмма с добавленной граничной стрелкой

Может иметь место и противоположная ситуация, когда некоторые стрелки, унаследованные от контекстной диаграммы, могут оказаться не нужными на диаграмме декомпозиции. Такие стрелки можно удалить. На диаграмме верхнего уровня они будут отмечены знаком туннелирования в виде квадратных скобок возле окончания стрелки. Это означает, что стрелка не переносится на диаграмму нижнего уровня.

8. Рисование внутренних стрелок

Для связи функций между собой используются внутренние стрелки, т. е. стрелки, которые не касаются границы диаграммы, начинаются у одного и кончаются у другого блока.

В IDEF0 различают пять типов связей блоков (рис. 1.17).

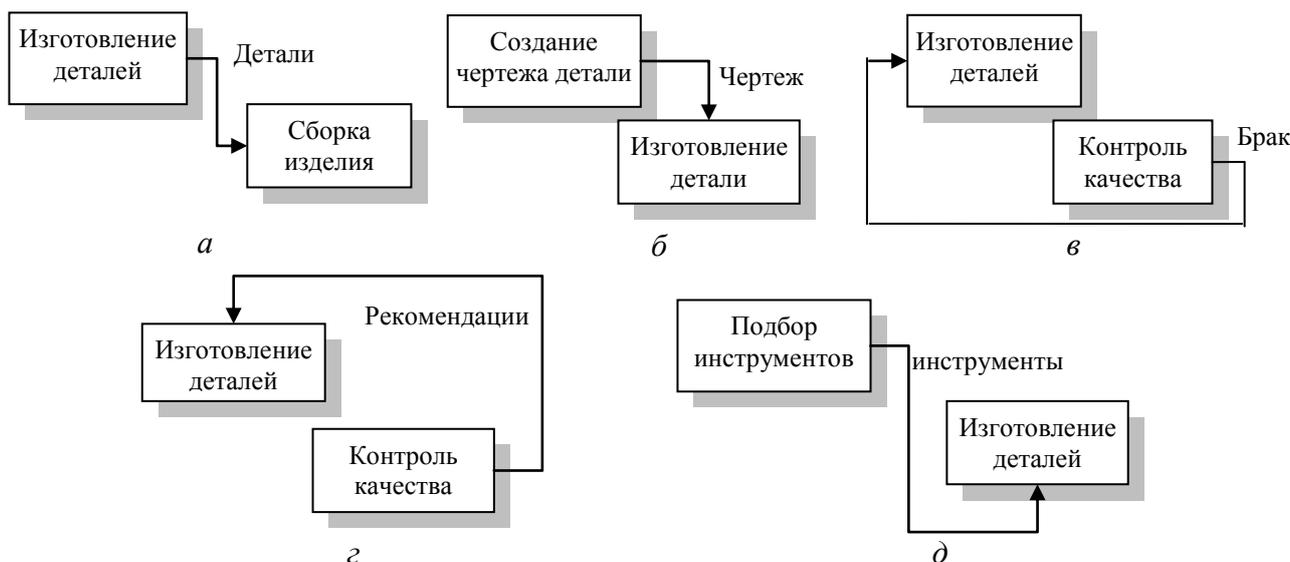


Рис. 1.17. Типы связей блоков:

а – связь по входу; б – связь по управлению; в – обратная связь по входу;
г – обратная связь по управлению; д – связь выход-механизм

Связь по входу (output-input), когда стрелка выхода предыдущего блока направляется на вход следующего, нижестоящего, блока. Например, на рис. 1.17 а стрелка "Детали" связывает работы "Изготовление деталей" и "Сборка изделия".

Связь по управлению (output-control), когда выход вышестоящего блока направляется на управление нижестоящей. На рис. 1.17 б стрелка "Чертеж" связывает функции "Создание чертежа детали" и "Изготовление детали", при этом чертеж не претерпевает изменений в процессе изготовления деталей.

Обратная связь по входу (output-input feedback), когда выход нижестоящего блока направляется на вход вышестоящего. Такая связь, как правило, используется для описания циклов. На рис. 1.17 в стрелка "Брак" связывает функции "Изготовление деталей" и "Контроль качества", при этом выявленный на контроле брак направляется на вторичную переработку.

Обратная связь по управлению (output-control feedback), когда выход нижестоящего блока направляется на управление вышестоящего (стрелка "Рекомендации", рис. 1.17 г). Обратная связь по управлению часто используется для корректировки управления (регулирования) по результатам контроля.

Связь выход-механизм (output-mechanism), когда выход одного блока направляется на механизм другого. Эта взаимосвязь используется реже остальных и показывает, что одна функция подготавливает ресурсы, необходимые для проведения другой (стрелка "Инструменты" на рис. 1.17 д).

Для рисования внутренней стрелки необходимо в режиме рисования стрелок щелкнуть по нужной стороне (например, выхода) одного блока и затем по нужной стороне (например, входа) другого. Пример диаграммы с внутренними стрелками приведен на рис. 1.18.

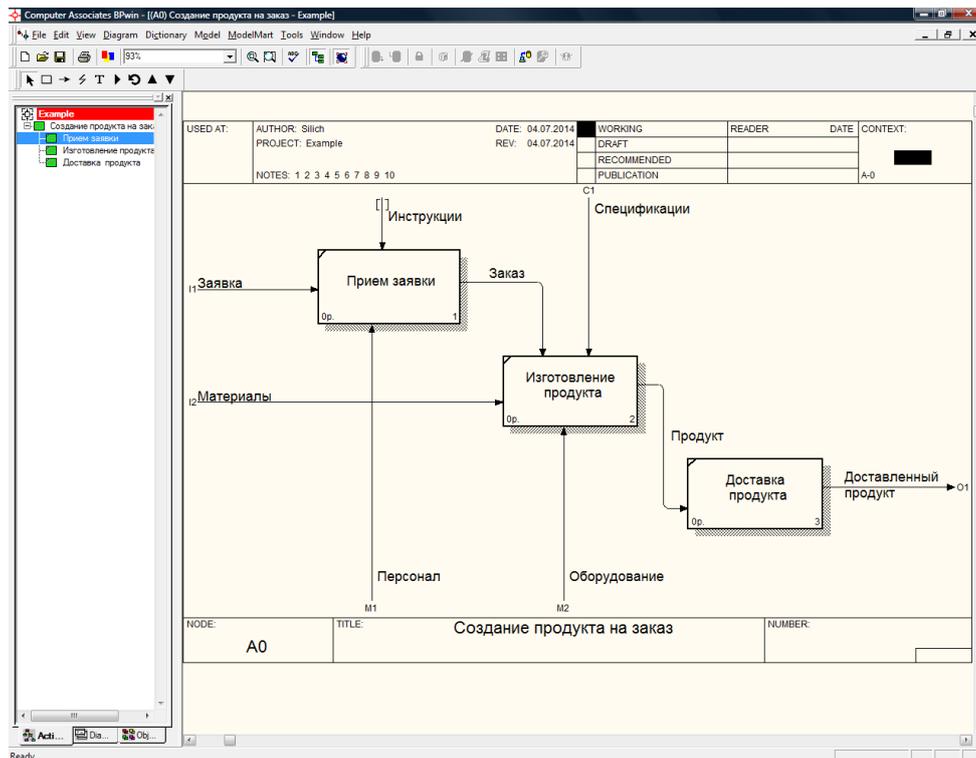


Рис. 1.18. Декомпозиционная диаграмма с внутренними стрелками

9. Разветвление и слияние стрелок

В некоторых случаях одну и ту же стрелку необходимо соединить более чем с одним функциональным блоком. Одни и те же данные или объекты, порожденные одной функцией, могут использоваться сразу в нескольких других функциях. С другой стороны, стрелки выхода разных функций могут представлять собой одинаковые или однородные данные или объекты, которые в дальнейшем используются или перерабатываются в одном месте. Для моделирования таких ситуаций в IDEF0 используются разветвляющиеся и сливающиеся стрелки.

Для разветвления стрелки нужно в режиме редактирования стрелки щелкнуть на стрелку, которую необходимо разветвить, а потом щелкнуть на соответствующей стороне того функционально блока, который получает ответвление.

Для слияния двух стрелок выхода нужно в режиме редактирования стрелки сначала щелкнуть по правой стороне блока, выход которого необходимо слить с уже имеющейся стрелкой выхода другого блока, а затем по самой стрелке.

Пример диаграммы с разветвленными стрелками приведен на рисунке 1.19.

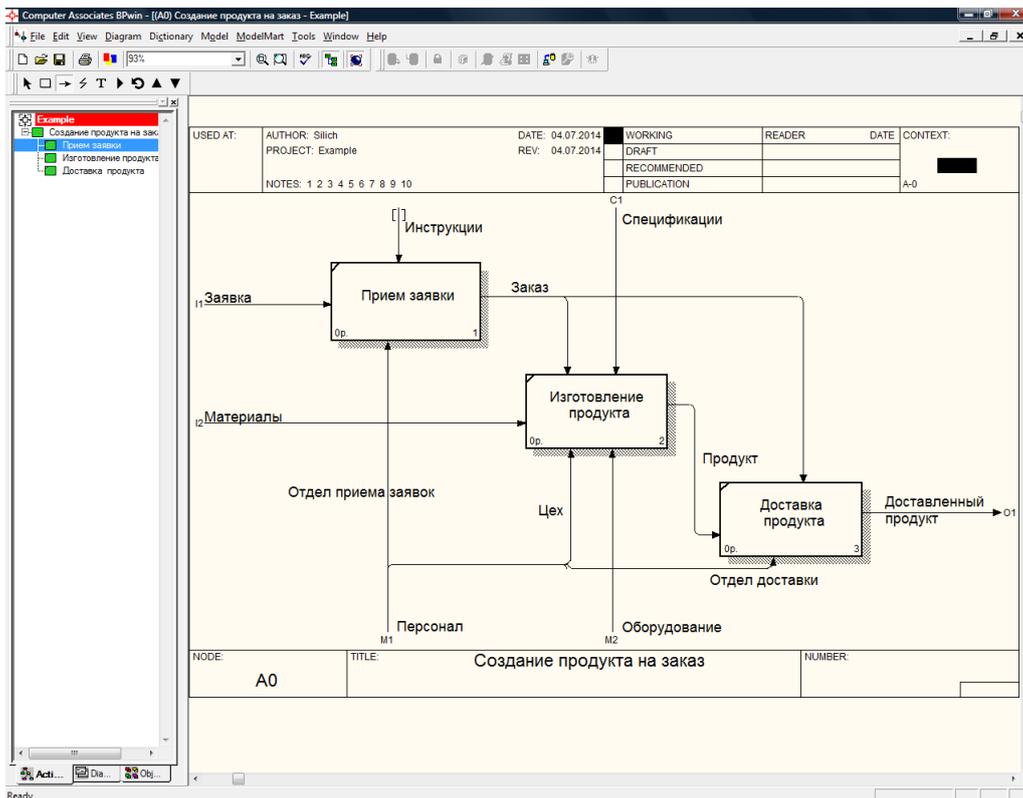


Рис. 1.19. Декомпозиционная диаграмма с разветвленными стрелками

Для именования отдельной ветви разветвляющихся и сливающихся стрелок следует выделить на диаграмме только одну ветвь, после этого вызвать редактор имени и присвоить имя стрелке. Это имя будет соответствовать только выделенной ветви. Если стрелка именована до разветвления, а после разветвления ни одна из ветвей не именована, то подразумевается, что каждая ветвь моделирует те же данные или объекты, что и ветвь до разветвления. На рис. 1.19 граничная стрелка «Персонал» была разветвлена на три ветви, каждая из которых имеет свое имя – «Отдел приема заявок», «Цех», «Отдел доставки».

10. Создание диаграммы дерева узлов

Диаграмма дерева узлов показывает иерархию функций в модели и позволяет рассмотреть всю модель целиком, но не показывает взаимосвязи между блоками (стрелки) (рис. 1.20).

Для создания диаграммы дерева узлов следует выбрать в меню пункт Diagram/Add Node Tree. Вызывается диалог Node Tree Wizard, в котором необходимо ввести имя диаграммы дерева узлов, узел верхнего уровня и глубину дерева – Number of Levels (по умолчанию 3). Второй диалог позволяет задать свойства диаграммы дерева узлов. По умолчанию нижний уровень декомпозиции показывается в виде списка, остальные функции – в виде прямоугольников. Для отображения всего дерева в виде прямоугольников следует выбрать опцию Bullet Last Level. Группа Connection Style позволяет выбрать стиль соединительных линий – диагональные (по умолчанию) или ортогональные.

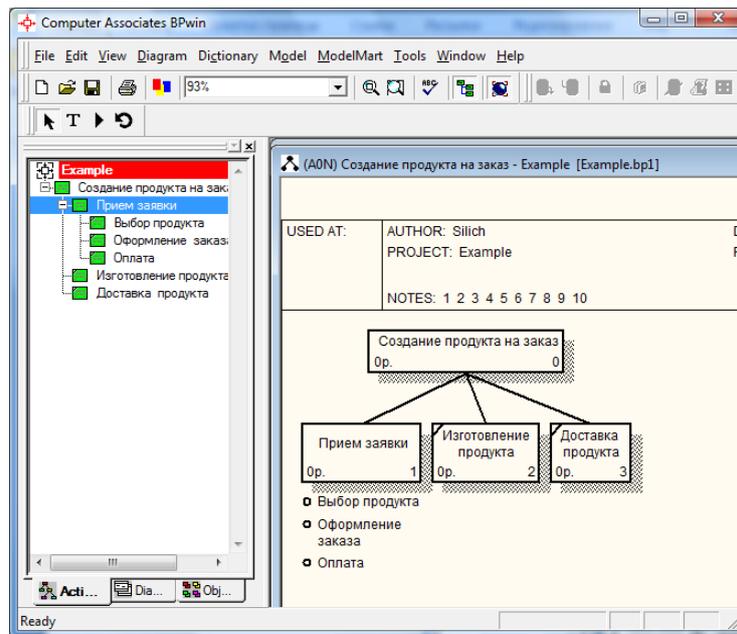


Рис. 1.20. Диаграмма дерева узлов

11. Доработка модели

Завершите создание IDEF0-модели для бизнес-процесса, выбранного вами на шаге 1 в качестве индивидуального задания. Законченная модель должна содержать, как минимум 4-5 диаграмм: контекстную, декомпозиционную диаграмму первого уровня, одну или несколько декомпозиционных диаграмм второго уровня и диаграмму дерева узлов. Все стрелки на диаграммах должны быть поименованы.

Некоторые рекомендации по составлению IDEF0-модели:

- блоки на диаграмме должны располагаться по диагонали (отношение доминирования);
- неконтекстные диаграммы должны содержать количество блоков от 3 до 6;
- следует максимально увеличить расстояние между работами, поворотами и пересечениями стрелок;
- если две стрелки проходят параллельно (начинаются из одной и той же грани одной работы и заканчиваются на одной и той же грани другой работы), то по возможности следует их объединить и назвать единым термином;
- обратные связи по входу рисуются "нижней" петлей, обратная связь по управлению – "верхней";
- следует минимизировать число пересечений, петель и поворотов стрелок.

Лабораторная работа №2 «Создание IDEF3-модели бизнес-процесса»

Цель работы: Получить практические навыки в построении IDEF3-модели бизнес-процесса средствами пакета BPWin.

Порядок выполнения работы.

1. Выбор задания.

Выберите бизнес-процесс, для которого будете формировать модель. Вы можете выбрать один из вариантов процессов, описанных в приложении, или предложить свой вариант. Можно выбрать тот же процесс, для которого на предыдущей лабораторной работе строилась IDEF0-модель, или часть этого процесса. При выборе учтите, что процесс обязательно должен иметь разветвления в технологии его выполнения и параллельно выполняемые работы.

2. Знакомство с основами методологии IDEF3.

IDEF3-методология предназначена для описания потоков работ (workflow или process flow). Она широко используется для документирования технологических процессов, особенно в случаях, когда в модели необходимо показать последовательность выполнения процесса. Как и методология IDEF0, IDEF3 построена на принципах декомпозиции и иерархического упорядочения: контекстная диаграмма отражает процесс в целом, а диаграммы декомпозиции — процесс в виде совокупности более мелких работ. Однако IDEF3-диаграммы, в отличие от IDEF0-диаграмм, позволяют описать логику процесса — всевозможные варианты ветвления и слияния потоков работ.

Основными элементами диаграмм являются единицы работ, отображающие действия, процессы, функции, этапы выполнения работ. Это аналоги функциональных блоков IDEF0-модели. На IDEF3-диаграммах, как и на IDEF0-диаграммах, отображаются связи между работами, однако смысл этих связей другой — это не объекты или информация, а переход от одной работы к другой. Стрелка, идущая от одной работы к другой, означает, что после окончания первой работы запустится вторая. Сторона блока, из которой выходит стрелка и в которую входит, не играет никакой роли. Для отображения разветвлений, параллельных потоков работ, используются специальные элементы диаграмм — перекрестки типа И, ИЛИ, Исключающего ИЛИ.

Но объекты, связанные с работами, тоже можно отобразить в IDEF3-модели. Во-первых, каждая работа должна иметь ассоциированный документ, который включает текстовое описание компонентов работы: объектов и фактов, связанных с работой, ограничений, накладываемых на работу, и дополнительное описание работы. Кроме того, имеется специальный вид связей, моделирующих потоки объектов, т.е. передачу материальных объектов или данных от работы к работе. Имеются также объекты ссылок, используемые для комментариев к элементам модели, для описания циклических переходов, ссылок на другие диаграммы.

IDEF3-диаграммы может дополнять IDEF0-модель. BPWin допускает построение смешанных моделей, в которых диаграммы декомпозиции отдельных функциональных блоков IDEF0-модели могут быть построены в нотации IDEF3. смешанная модель наилучшим образом описывает все стороны деятельности предприятия.

3. Создание контекстной IDEF3-диаграммы.

Для создания диаграммы в нотации IDEF3 необходимо выбрать в системном меню пункт File/New. В диалоге, который изображен на рисунке 2.1, необходимо определить имя модели и используемый метод (IDEF3).

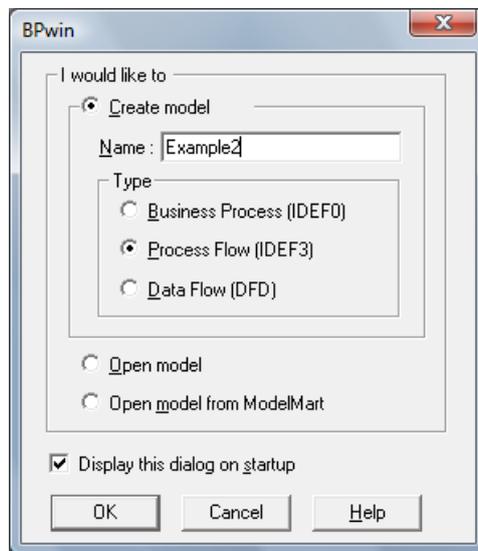


Рис. 2.1. Определение метода при создании диаграммы

После нажатия на кнопку Ok появляется диалоговое окно Properties, в котором необходимо определить автора модели. После нажатия на кнопку Ok появляется окно диаграммы с контекстной диаграммой, содержащей единственный блок (работу верхнего уровня). Введите имя блока, выбрав в контекстном меню пункт Name (см. рис. 2.2).

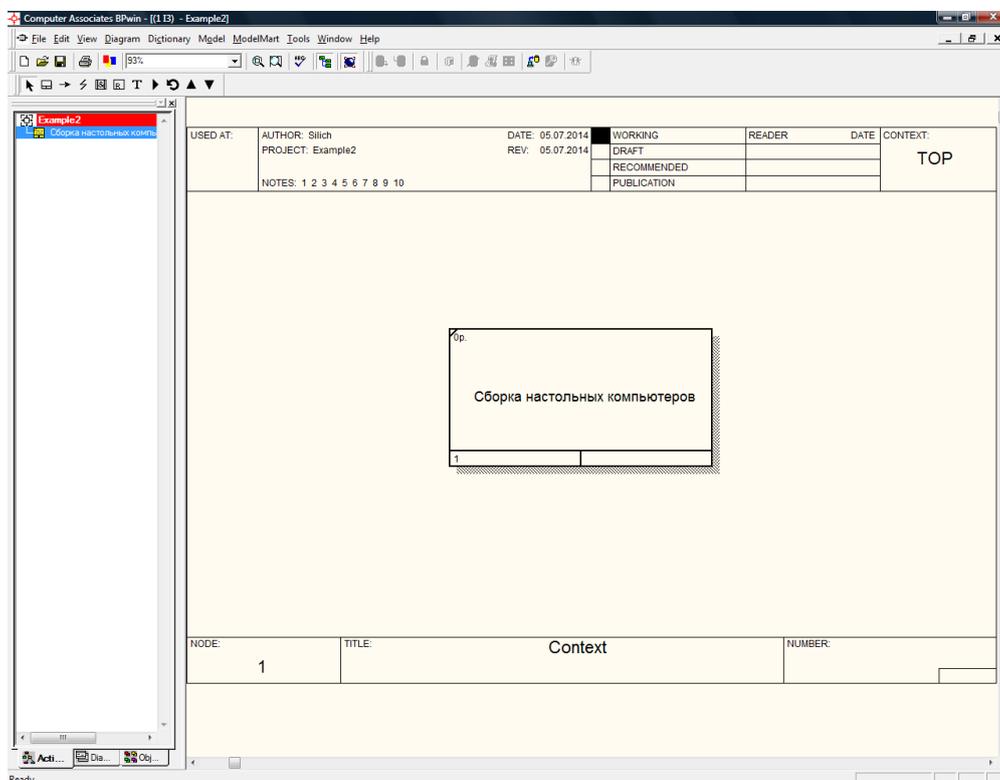


Рис 2.2. Контекстная диаграмма в нотации IDEF3

В IDEF3 вместо понятия «функциональный блок» используется понятие «единица работы (Unit of Work, UOW)» или «работа (activity)». UOW изображаются прямоугольниками с прямыми углами и имеют имя, выраженное отглагольным существительным, обозначающим процесс действия. Обычно номер работы состоит из номера родительской работы и порядкового номера на текущей диаграмме.

Работа в IDEF3 требует более подробного описания, чем работа в IDEF0. Каждая UOW должна иметь ассоциированный документ, который включает текстовое описание компонентов работы: объектов (Objects) и фактов (Facts), связанных с работой, ограничений (Constraints), накладываемых на работу, и дополнительное описание работы (Description). Эта информация заносится в том же диалоге Activity Properties, в котором вносилось имя работы, но на вкладке UOW (см. рис. 2.3).

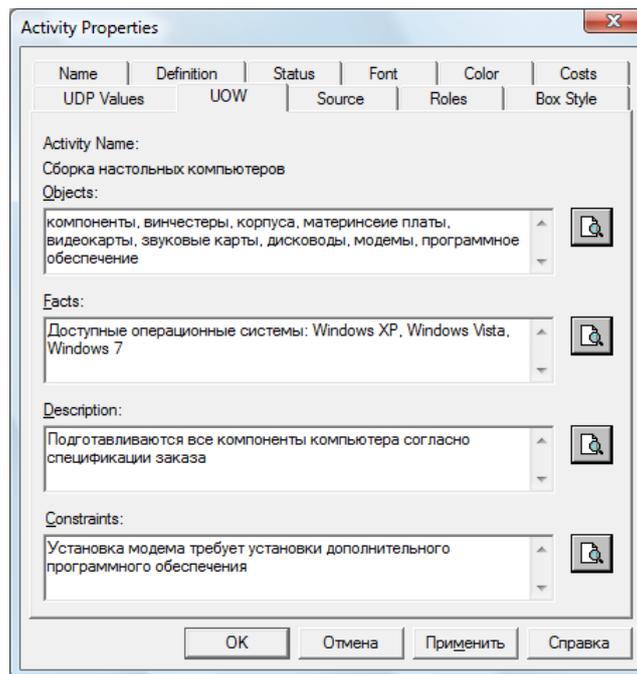


Рис. 2.3. Определение свойств работы

Можете добавить на контекстную диаграмму цель моделирования и точку зрения. Граничных стрелок рисовать не нужно, т.к. IDEF3-диаграммы показывают только последовательности работ и поэтому не используют граничные стрелки

4. Создание декомпозиционной IDEF3-диаграммы.

Контекстную диаграмму можно декомпозировать аналогично декомпозиции в нотации IDEF0, т.е. с помощью инструмента , но в диалоге указывается нотация IDEF3. На декомпозиционной диаграмме необходимо разместить и именовать все блоки, например, как на рисунке 2.4.

В отличие от IDEF0-диаграмм блоки обычно располагаются не по диагональной схеме, а по линейной – справа налево, так, чтобы работа, которая выполняется первой, была на левом краю диаграммы, а выполняемая последней – на правом. Потoki работ, выполняемые параллельно, а также альтернативные потоки работ, располагают друг над другом. Иногда работы располагают в порядке следования сверху вниз.

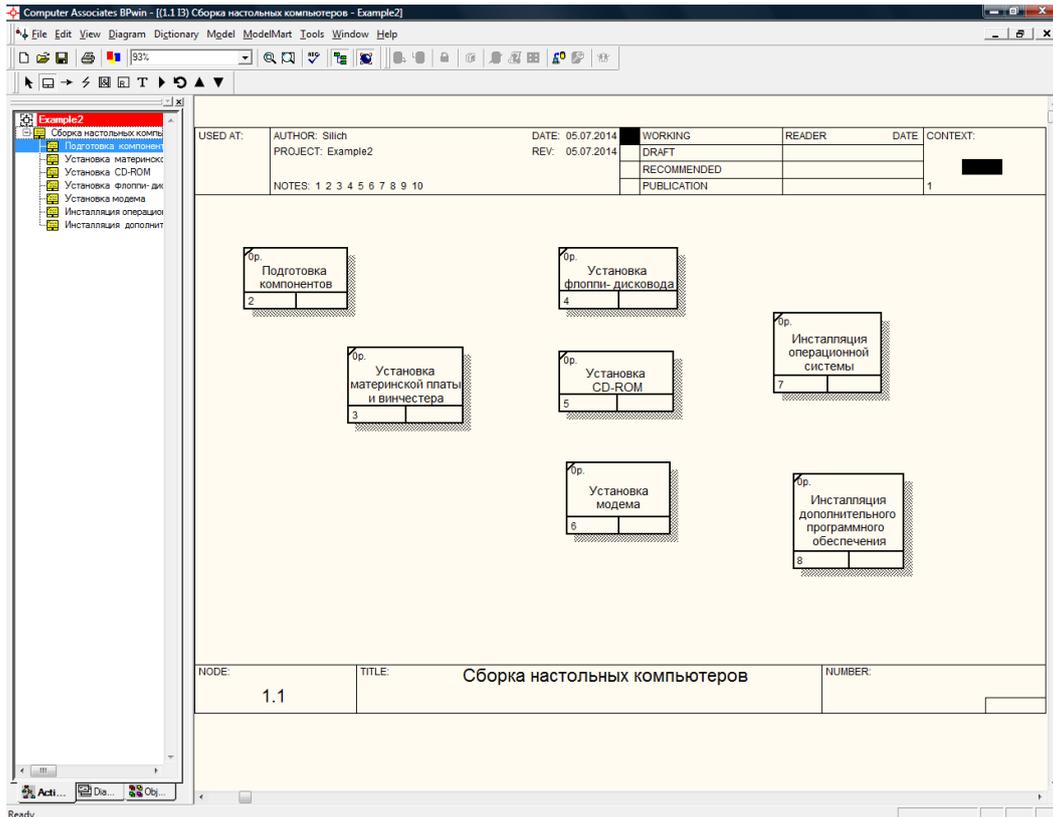


Рис. 2.4. Декомпозиционная диаграмма с функциональными блоками

Для связывания блоков используются стрелки. Все связи в IDEF3 однонаправленные и могут быть направлены куда угодно, но обычно диаграммы IDEF3 стараются построить так, чтобы связи были направлены слева направо. В IDEF3 различают три типа стрелок, изображающих связи, стиль которых устанавливается во вкладке Style диалога Arrow Properties (пункт Style контекстного меню).

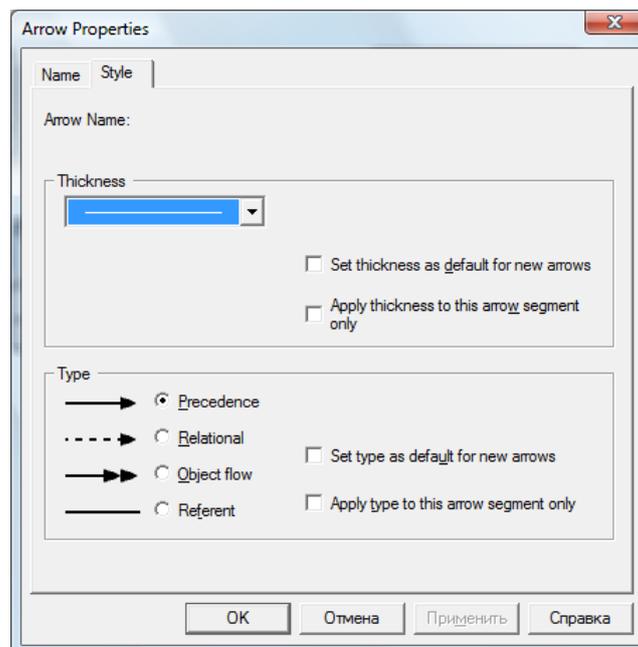


Рис. 2.5. Вкладка Style диалога Arrow Properties

Стрелка приоритета (Precedence) – сплошная линия, связывающая единицы работ (UOW). Рисуеться слева направо или сверху вниз. Показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется.

Стрелка отношения (Relational Link) – пунктирная линия, используемая для изображения связей между единицами работ (UOW), а также между единицами работ и объектами ссылок.

Потоки объектов (Object Flow) – стрелка с двумя наконечниками, применяется для описания того факта, что объект используется в двух или более единицах работы, например когда объект порождается в одной работе и используется в другой.

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

Различают перекрестки слияния (Fan-in Junction) и разветвления (Fan-out Junction) стрелок. Перекресток не может использоваться одновременно для слияния и для ветвления. Все перекрестки на диаграмме нумеруются, каждый номер имеет префикс J. Можно редактировать свойства перекрестка при помощи диалога Junction Properties (вызывается из контекстного меню). В отличие от IDEF0 в IDEF3 стрелки могут сливаться и разветвляться только через перекрестки.

Для внесения перекрестка служит кнопка  в палитре инструментов. В диалоге junction Type Editor необходимо указать тип перекрестка. Смысл каждого типа приведен в табл. 2.1.

Таблица 2.1

Типы перекрестков

Знак	Наименование	Смысл для перекрестка слияния	Смысл для перекрестка ветвления
	Асинхронное И Asynchronous AND	Выходной процесс запустится, если завершились все входные процессы	После завершения входного процесса запустятся все выходные процессы
	Синхронное И Synchronous AND	Выходной процесс запустится, если завершились одновременно все входные процессы	После завершения входного процесса запустятся одновременно все выходные процессы
	Асинхронное ИЛИ Asynchronous OR	Выходной процесс запустится, если завершился один или несколько входных процессов	После завершения входного процесса запустятся один или несколько выходных процессов
	Синхронное ИЛИ Synchronous OR	Выходной процесс запустится, если завершились один или несколько входных процессов, причем завершились одновременно	После завершения входного процесса запустится один или несколько выходных процессов, причем запустятся одновременно
	Исключающее ИЛИ Exclusive OR (XOR)	Выходной процесс запустится, если завершился только один входной процесс	После завершения входного процесса запустится только один выходной процесс

Пример декомпозиционной диаграммы приведен на рис. 2.6.

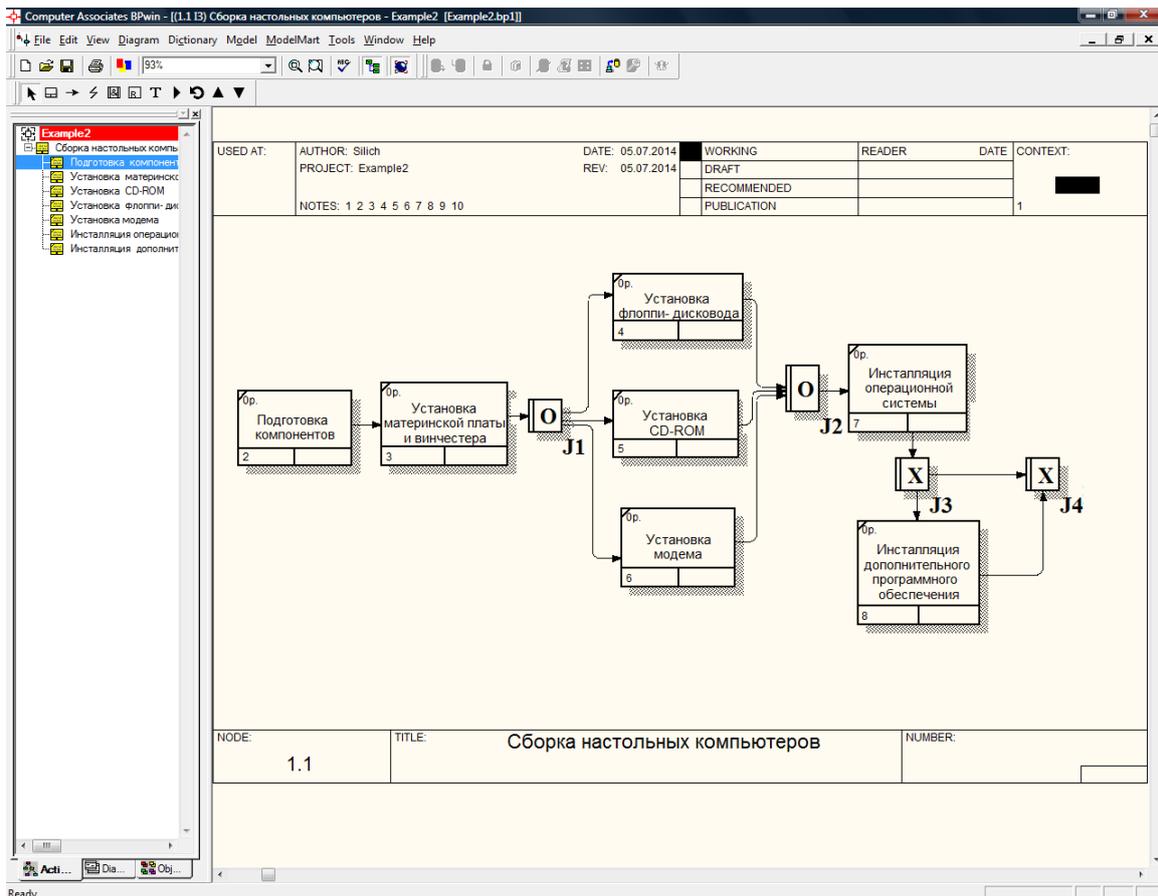


Рис. 2.6. Декомпозиционная диаграмма с перекрестками

Поясним использование стрелок и перекрестков на примере диаграммы, представленной на рис. 2.6. После выполнения работы «Подготовка компонентов» запускается работа «Установка материнской платы и винчестера». Следом идут работы по установке различных устройств (флорпи-дисквода, CD-ROM, модема), причем может быть установлено только одно устройство или несколько (или все) в зависимости от того, что указано в спецификации заказа. Поэтому используется перекресток разветвления типа ИЛИ. Он должен быть асинхронным, т.к. работы по установке устройств могут выполняться не одновременно. Для запуска работы «Инсталляция операционной системы» требуется завершение (не обязательно одновременное) запущенных работ по установке устройств. Затем либо выполняется работа «Инсталляция дополнительного программного обеспечения» (если это предусмотрено спецификацией заказа), либо процесс может завершиться. Поэтому здесь логично использовать перекрестки для слияния и разветвления типа исключающего ИЛИ.

Правила создания перекрестков:

- каждому перекрестку слияния должен предшествовать перекресток ветвления;
 - перекресток слияния «И» не может следовать за перекрестком ветвления типа синхронного, асинхронного или исключающего «ИЛИ»;
 - перекресток слияния типа исключающего «ИЛИ» не может следовать за перекрестком ветвления типа «И»;
 - перекресток, имеющий одну стрелку на одной стороне, должен иметь более одной стрелки на другой;
 - перекресток не может быть одновременно перекрестком слияния и ветвления.
- Если нужно одновременно осуществить слияние и разветвление потоков работ, вводится каскад перекрестков.

5. Использование объектов-ссылок.

Объект ссылки (Referent) в IDEF3 — это объекты, используемые для комментариев к элементам модели. Кроме того, они могут служить для описания циклических переходов, ссылок на другие диаграммы. Имя ссылки задается именем существительным, номер — числом.

Для внесения объекта ссылки служит кнопка . Объект ссылки изображается в виде прямоугольника, похожего на прямоугольник работы. Имя объекта ссылки задается в диалоге Referent (рис. 2.7), вызываемом через контекстное меню (пункт Name).

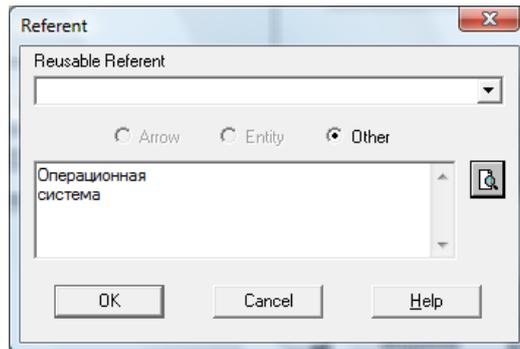


Рис. 2.7. Определение имени объекта-ссылки

Чтобы связать ссылку с работой, выберите инструмент , щелкните на ссылке (на любой стороне прямоугольника) и на работе (на любой стороне). Появится стрелка. Чтобы изменить ее стиль, выберите в контекстном меню пункт Style и в появившемся диалоге Arrow Properties (см. рис. 2.5) выберите тип (Type) Relational Link. Стрелка изменит свой вид и будет пунктирной линией. Пример диаграммы с объектами-ссылками приведен на рис. 2.8.

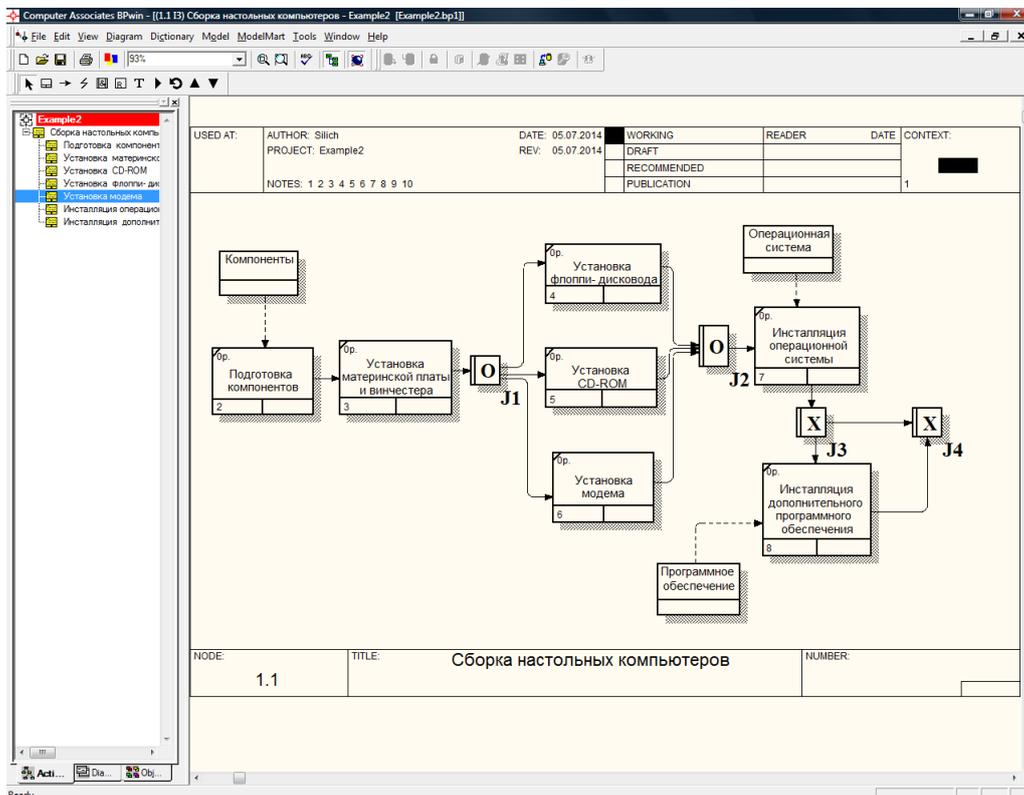


Рис. 2.8. Декомпозиционная диаграмма с объектами-ссылками

6. Создание диаграммы Swim Lane

Диаграмма **Swim Lane** является разновидностью диаграммы IDEF3, позволяющей явно описать роли и ответственности исполнителей в конкретной технологической операции. Эта диаграмма разделена на горизонтальные полосы, каждая полоса соответствует определенному исполнителю (точнее, его роли) и все работы, выполняемые этим исполнителем, помещаются на эту полосу. Помимо UOW полоса может содержать и другие объекты диаграммы IDEF3 (перекрестки и объекты ссылок), относящиеся к соответствующей роли.

Ролью (Role) может быть должность или позиция конкретного исполнителя. Примеры ролей: *Сборщик, Продавец, Технолог, Маркетолог, Разработчик, Дизайнер*. Роли объединяются в **группы ролей (Role Group)**. В качестве значения группы ролей может быть название предприятия, отдела, цеха или название региона, города и т. д. Одной роли может соответствовать несколько групп ролей. Конкретный исполнитель, связанный с определенной ролью и группой ролей, называется **ресурсом (Resource)**. В качестве значения ресурса можно использовать фамилию и имя сотрудника. Например, ресурс *Иванов В.П.* связан с ролью *Продавец* и группой *Продажа*.

Прежде чем создавать диаграмму Swim Lane, необходимо создать словарь ролей и групп ролей. Откройте словарь Role Group Dictionary (меню Dictionary/Role Group, рис. 2.9), создайте несколько групп ролей. Для каждой группы ролей может быть внесено описание (Definition), указано изображение (предварительно импортированное в словарь изображений Bitmap Dictionary) и важность группы (Importance).

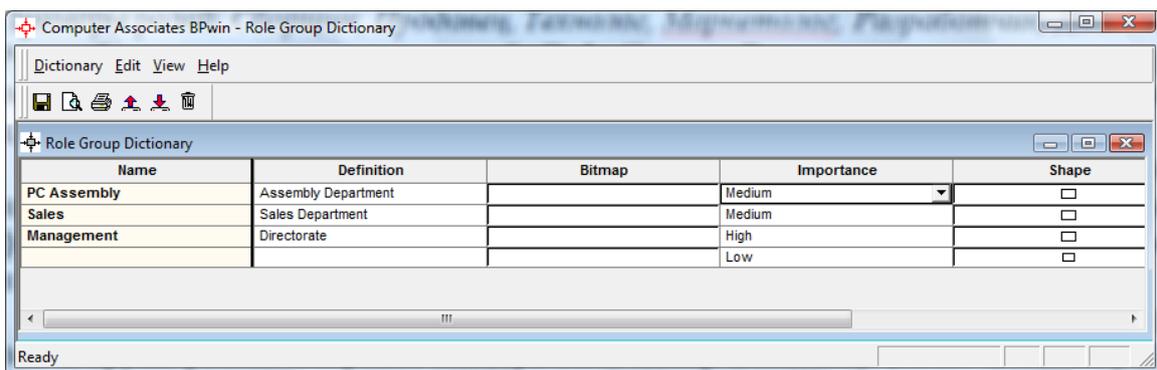


Рис. 2.9. Словарь групп ролей

Словарь ролей вызывается из меню Dictionary/Role (рис. 2.10). Для каждой роли можно внести определение, указать группу ролей, связать роль с изображением (Bitmap) и геометрической фигурой (Shape), указать важность роли.

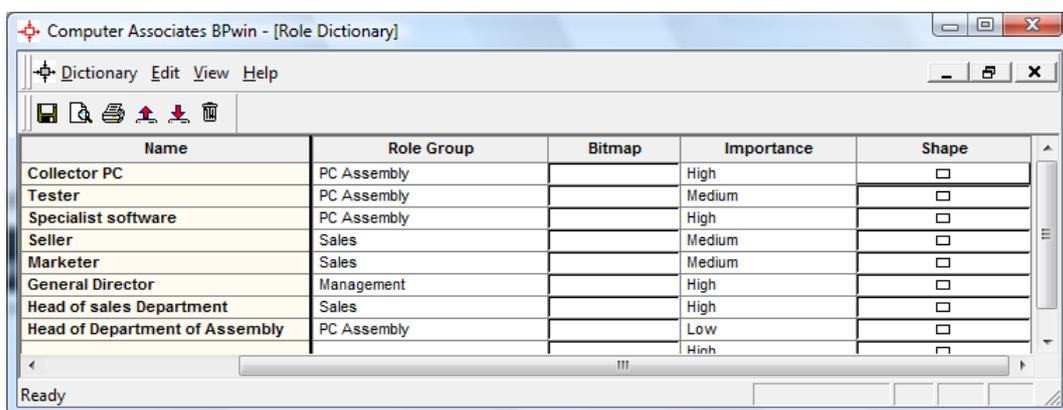


Рис. 2.10. Словарь ролей

Для создания диаграммы Swim Lane следует выбрать меню Diagram/Add Swim Lane diagram. Появляется гид Swim Lane diagram Wizard. В первом диалоге гида (рис. 2.11) следует внести название и имя автора диаграммы, выбрать имя и номер диаграммы IDEF3, на основе которой будет построена диаграмма, и группу ролей, из которой можно будет выбрать роли, связанные с диаграммой.

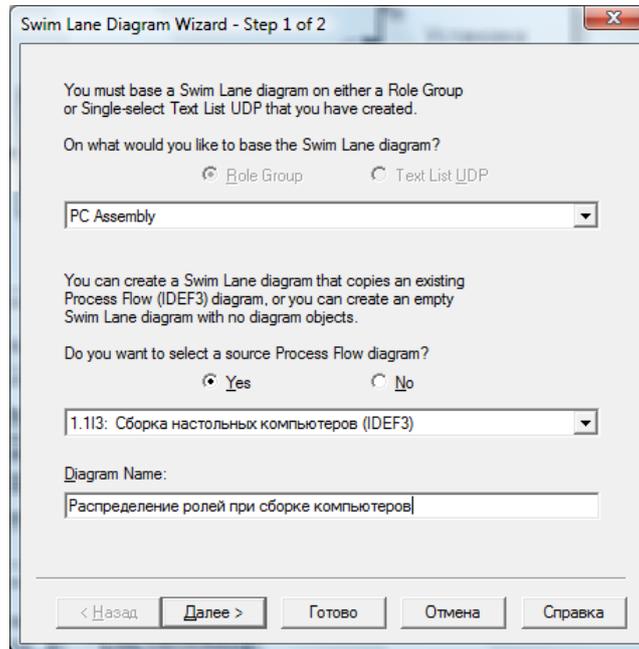


Рис. 2.11. Первый диалог гида Swim Lane Diagram Wizard

Во втором диалоге гида следует выбрать роли, на основе которых будет создана диаграмма (рис. 2.12). Диаграмма будет разделена на количество полос, указанных в колонке Display Swim Line.

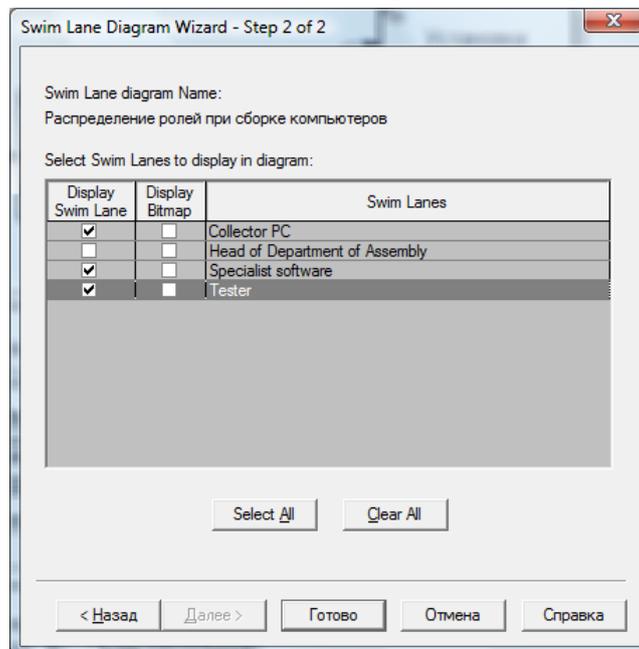


Рис. 2.12. Первый диалог гида Swim Lane Diagram Wizard

После щелчка по кнопке *Готово* создается новая диаграмма, все объекты которой расположены произвольно. Расположить объекты на полосах, соответствующих ролям, следует вручную (рис. 2.13).

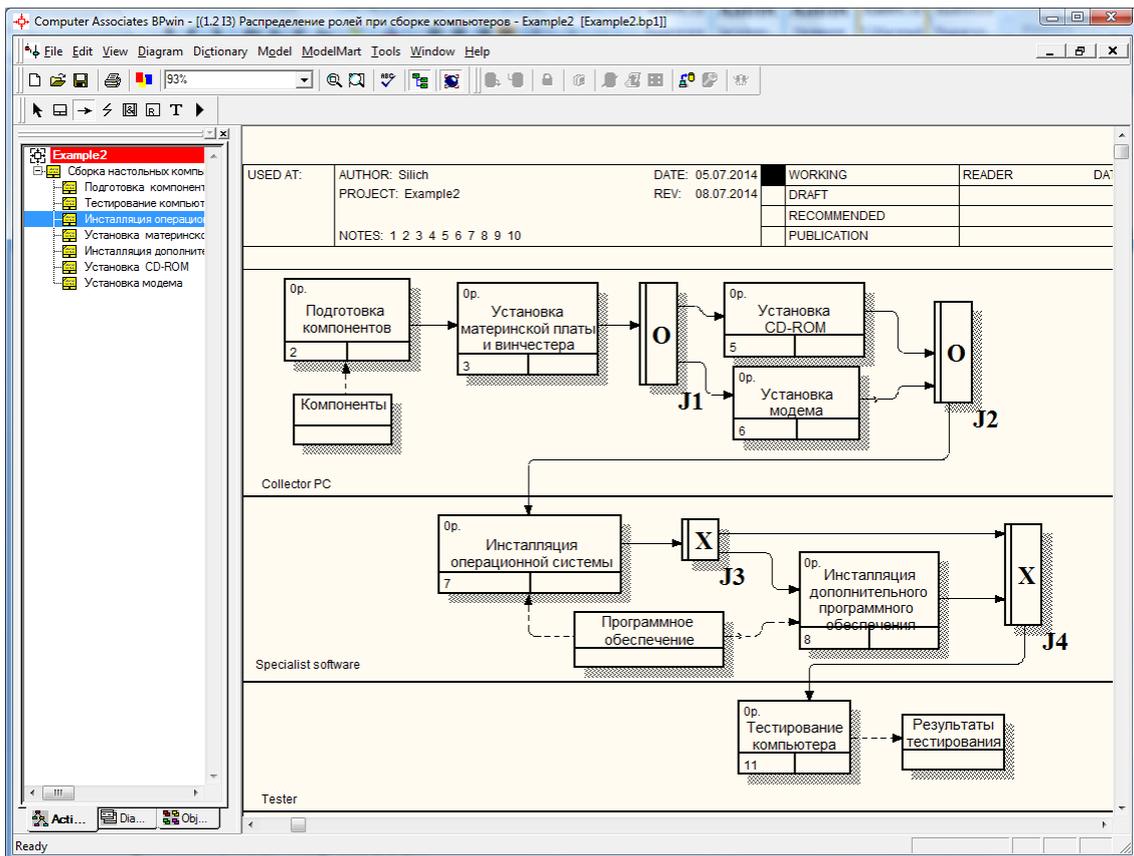


Рис. 2.13. Диаграмма Swim Lane

7. Доработка IDEF3-модели

Завершите создание IDEF3-модели для бизнес-процесса, выбранного вами на шаге 1 в качестве индивидуального задания. Законченная модель должна содержать, как минимум 3-4 диаграммы: контекстную, одну или несколько декомпозиционных диаграмм, диаграмму Swim Lane. Диаграммы декомпозиции должны содержать перекрестки (желательно использовать перекрестки нескольких типов), а также объекты-ссылки.

Лабораторная работа №3 «Создание DFD-модели бизнес-процесса»

Цель работы: Получить практические навыки в построении DFD-модели бизнес-процесса средствами пакета BPWin.

Порядок выполнения работы.

1. Выбор задания.

Выберите бизнес-процесс, для которого будете формировать модель. Вы можете выбрать один из вариантов процессов, описанных в приложении, или предложить свой вариант. Можно выбрать часть процесса, который моделировался на предыдущих лабораторных работах. При выборе учтите, что процесс обязательно должен предусматривать обработку информации, лучше, чтобы это была автоматизированная обработка с использованием одной или нескольких информационных систем.

2. Знакомство с основами методологии DFD.

Диаграммы потоков данных (Data flow diagramming, DFD) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации.

DFD описывает:

- процессы обработки информации (работы);
- потоки данных (стрелки, arrow), которые могут моделировать и потоки материальных объектов (изделия, документы);
- внешние ссылки (external references), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- хранилища данных (data store).

Как и в IDEF0, основными элементами DFD-диаграмм являются функциональные блоки, которые называются процессами или работами. Они преобразуют входы в выходы (чаще всего это преобразование входных данных в выходные). Блоки соединяются стрелками. В отличие от стрелок IDEF0, которые представляют собой жесткие ограничения на работу блоков, стрелки DFD показывают, как объекты (как правило, данные) двигаются от одной работы к другой. Это представление потоков совместно с хранилищами данных и внешними сущностями делает модели DFD более похожими на описание физических характеристик системы – движения объектов (data flow), хранения объектов (data stores), поставки и распространения объектов (external entities).

При построении диаграмм потоков данных наиболее часто используют две нотации: Йордана и Гейна-Сарсона. Обе нотации имеют одинаковый по названиям и значению элементный состав, но имеют различное его графическое изображение. В BPwin для построения диаграмм потоков данных используется нотация Гейна - Сарсона.

3. Создание контекстной DFD-диаграммы.

Контекстная DFD-диаграмма создается так же, как и аналогичная диаграмма в нотации IDEF0 или IDEF3 (пункт меню File/New), но в диалоге нужно указать тип модели – Data Flow (DFD). Определите автора модели в диалоговом окне Properties.

Появится окно с контекстной диаграммой, содержащее работу (процесс обработки информации) верхнего уровня.

Работы в DFD представляют собой функции системы, преобразующие входы в выходы. Хотя работы изображаются прямоугольниками со скругленными углами, смысл их совпадает со смыслом работ IDEF0 и IDEF3. Так же как работы IDEF3, они имеют входы и выходы, но не поддерживают управления и механизмы, как IDEF0. Работа на контекстной диаграмме обычно именуется по названию системы, например "*Система обработки информации*".

Определите цель, область и точку зрения на моделируемую систему так же, как и при создании контекстных диаграмм на предыдущих лабораторных работах.

В отличие от IDEF0-диаграмм, DFD-диаграмма не должна иметь граничных стрелок. Чтобы показать связь системы с окружением, субъекты окружения (внешние сущности) явно отображаются на диаграмме, и соединяются с системой (работой) стрелками.

Внешние сущности изображают входы в систему и/или выходы из системы. Как правило, они представляют собой материальный предмет или физическое лицо, например: *Заказчик, Пользователь, Персонал, Поставщик, Клиент, Банк*. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы. Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах. Обычно такой прием используют, чтобы не рисовать слишком длинных и запутанных стрелок.

Чтобы добавить внешнюю сущность выберите кнопку  (добавить внешнюю ссылку) в палитре инструментов и щелкните по свободному месту на диаграмме. Задайте имя сущности аналогично заданию имени объекта ссылки при создании IDEF3-диаграммы (см. рис. 2.7). Номера внешним сущностям присваиваются автоматически.

Если нужно, добавьте другие внешние сущности.

Необходимо показать связи между работой и внешними сущностями, которые моделируются с помощью стрелок.

Стрелки (потoki данных) описывают движение объектов из одной части системы в другую, например, от одной работы к другой или от внешней сущности к работе, или из хранилища данных к работе и т.д. Обычно это передача информационных объектов (документов, сообщений), но допустимо с помощью стрелок моделировать и материальные потоки. Поскольку в DFD каждая сторона работы не имеет четкого назначения, как в IDEF0, стрелки могут подходить выходить из любой грани прямоугольника работы. В DFD также применяются двунаправленные стрелки для описания диалогов типа «команда-ответ» между работами, между работой и внешней сущностью и между внешними сущностями.

Стрелки рисуются аналогично рисованию стрелок на IDEF0- или IDEF3- диаграммах. Используя инструмент , нарисуйте связи работы с внешними сущностями. Для каждой связи задайте имя, отражающее содержание соответствующего потока. Чтобы сделать стрелку двунаправленной, в диалоге Arrow Properties (пункт Style контекстного меню) на вкладке Style выберите тип (Type) Bidirectional.

Пример контекстной DFD-диаграммы приведен на рисунке 3.1.

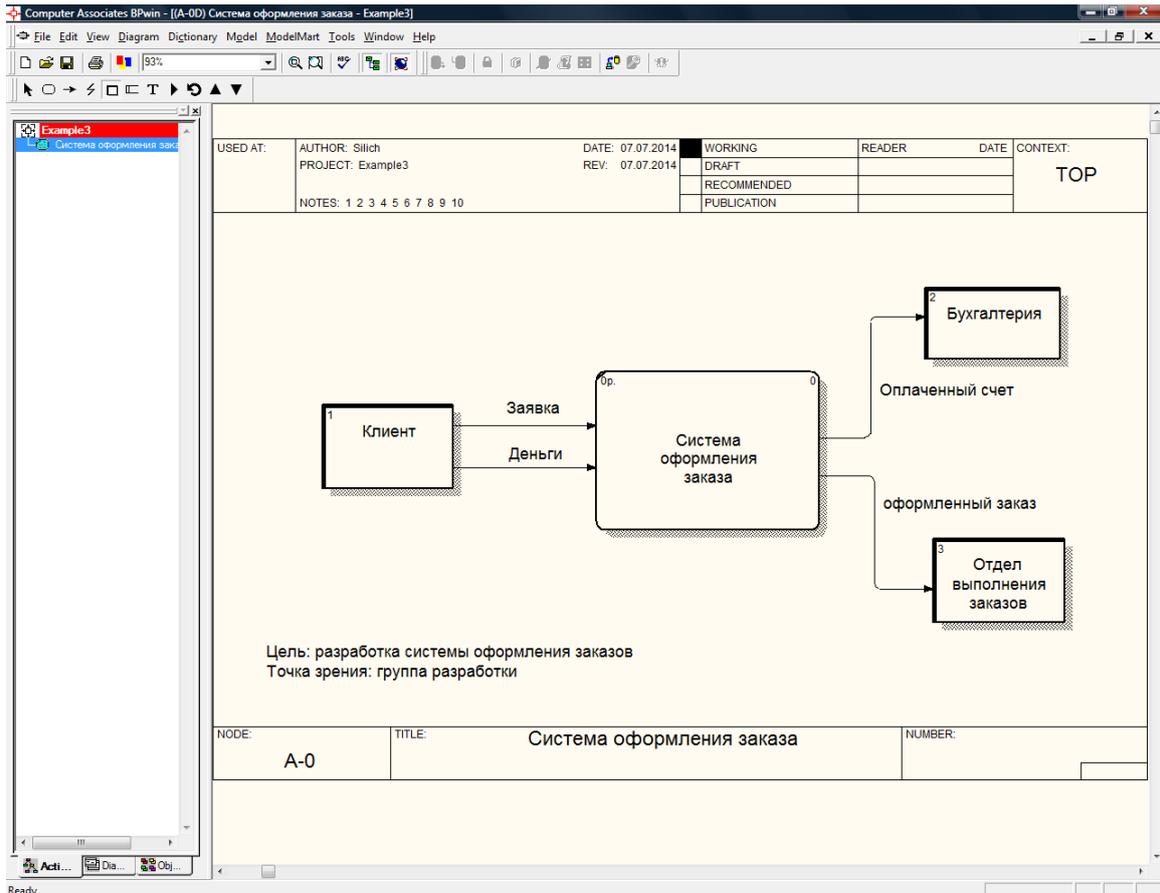


Рис. 3.1. Контекстная диаграмма в нотации DFD

4. Создание декомпозиционной DFD-диаграммы.

Выделите работу на контекстной диаграмме и декомпозируйте ее с помощью инструмента . Например, систему оформления заказа можно декомпозировать на три работы: «Консультирование клиента», «Оформление заказа» и «Прием оплаты».

Расположение блоков на диаграмме, как и для диаграммы IDEF3, может быть любым, но обычно их располагают слева направо в порядке выполнения соответствующих работ. Для каждого блока работы задайте имя. Обычно в имени используется глагол или отглагольное существительное.

Номер работы присваивается автоматически. Настроить параметры нумерации работ, а также внешних сущностей и хранилищ можно во вкладке Numbeing диалога Model Properties (меню Model /Model Properties).

На диаграмму декомпозиции с контекстной диаграммы будут перенесены стрелки входа и выхода родительской работы. Они будут представлены в виде граничных стрелок (см. рис. 3.2). Согласно нотации DFD диаграмма не должна иметь граничных стрелок – все стрелки должны начинаться и заканчиваться на работах, хранилищах данных или внешних сущностях. Поэтому следует удалить все граничные стрелки, создать соответствующие внешние сущности и вместо граничных провести внутренние стрелки, связывающие внешние сущности и работы. При этом на контекстной диаграмме стрелки входа и выхода родительской работы, соответствующие удаленным граничным стрелкам, будут иметь знак туннелирования в виде квадратных скобок.

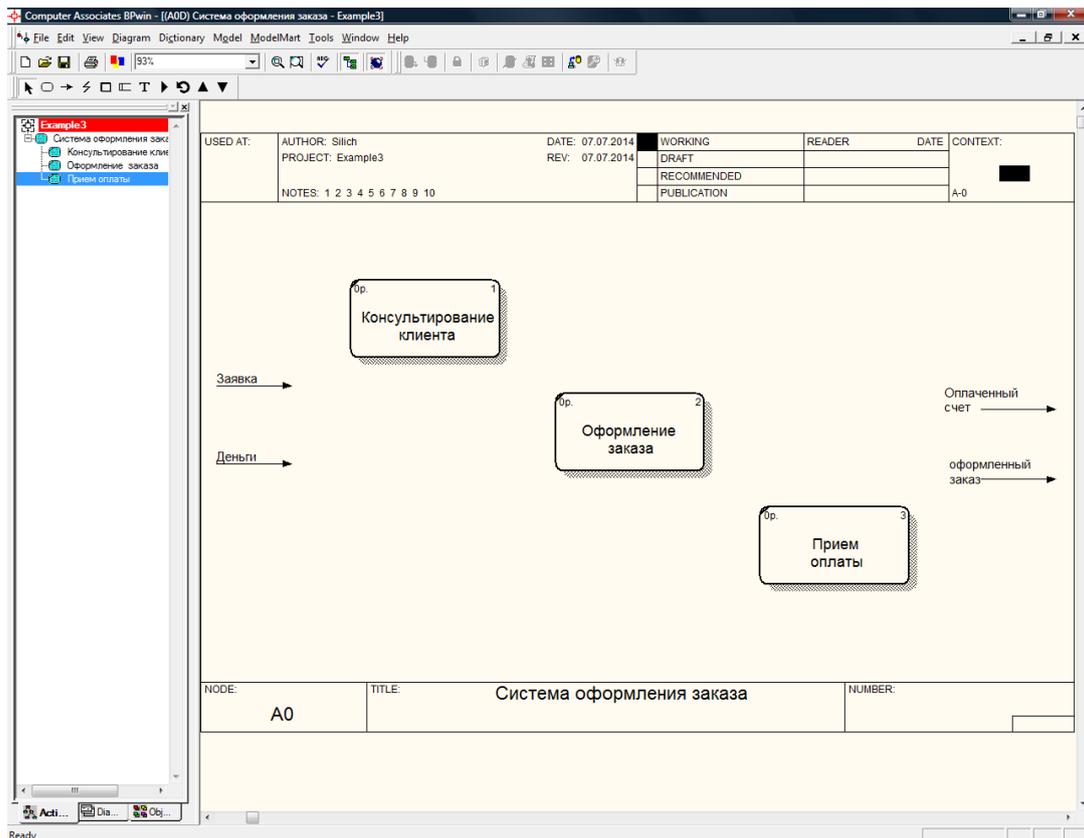


Рис 3.2. Декомпозиционная DFD-диаграмма с граничными стрелками

Поместите на диаграмму внешние сущности (такие же, как на контекстной диаграмме). Соедините их с работами внутренними стрелками, задайте имя для каждой связи. При этом в диалоге Arrow Properties Вы можете выбирать имя из списка ранее введенных имен.

Некоторые данные (объекты), которые обрабатываются в блоках работ, поступают не извне системы (от внешних сущностей) и не от других работ, а из хранилищ. Результаты работ также могут поступать в хранилища.

Хранилища данных (Data store) представляют собой собственно данные, к которым осуществляется доступ. Эти данные также могут быть созданы или изменены работами. В отличие от потоков данных, описывающих данные в движении, хранилища данных отображают данные в покое, т. е. данные, которые сохраняются в памяти между последующими работами. Информация, которую содержит хранилище данных, может использоваться в любое время после её определения. При этом данные могут выбираться в любом порядке. Примеры хранилищ: *База данных, Репозиторий, Карточка, Архив, Журнал.*

В материальных системах, в которых обрабатываются материальные объекты, а не информационные, хранилища моделируют места, где объекты ожидают обработки и в которые поступают после обработки, например: *Склад.*

Чтобы добавить хранилище данных выберите кнопку  (добавить Data store) в палитре инструментов и щелкните по свободному месту на диаграмме. Каждое хранилище данных имеет уникальный номер, который присваивается автоматически. Задайте имя хранилища данных, отражающее его содержание.

Соедините стрелками (потоками данных) работы, внешние сущности и хранилища данных.

В DFD стрелки могут сливаться и разветвляться, что позволяет описать декомпозицию стрелок. Каждый новый сегмент сливающейся или разветвляющейся стрелки может иметь собственное имя.

Пример декомпозиционной DFD-диаграммы приведен на рисунке 3.3.

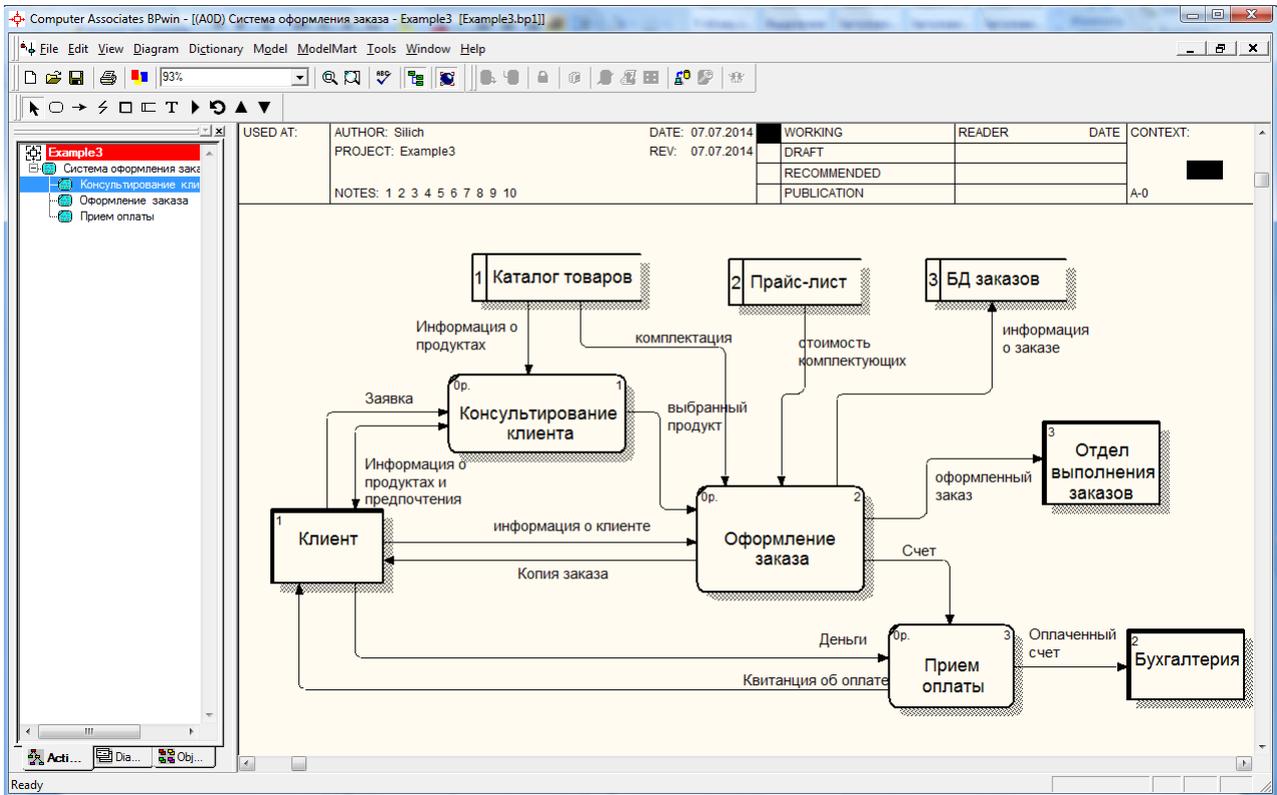


Рис 3.2. Декомпозиционная DFD-диаграмма

В заключение желательно документировать работы на вкладке Definition диалого Activity Properties (пункт Definition/Note контекстного меню).

5. Доработка модели

Завершите создание DFD-модели для бизнес-процесса, выбранного вами на шаге 1 в качестве индивидуального задания. Диаграммы обязательно должны содержать внешние сущности и хранилища данных. Все стрелки (потоки данных) должны быть обязательно поименованы.

Лабораторная работа №4 **«Создание прецедентной UML-модели бизнес-процесса»**

Цель работы: Получить практические навыки в построении прецедентной UML-модели бизнес-процесса средствами пакета Rational Rose.

Порядок выполнения работы.

1. Выбор задания.

Выберите бизнес-процесс, для которого будете формировать модель. Вы можете выбрать один из вариантов процессов, описанных в приложении, или предложить свой вариант. Можно выбрать один из процессов, для которого на предыдущих лабораторных работах строилась модель по одной из структурных методологий. Желательно, чтобы процесс имел различные версии, т.е. альтернативные потоки событий.

2. Знакомство с основами языка моделирования UML.

Унифицированный язык моделирования UML (Unified Modeling Language) предназначен для описания, визуализации и документирования бизнес-систем на базе объектно-ориентированного подхода с целью последующего использования моделей бизнес-процессов для реализации их в виде программного обеспечения.

Бурное развитие объектно-ориентированных языков программирования, сопровождающееся возрастанием сложности прикладных программ, вызвало потребность в создании объектно-ориентированного языка для формирования предварительной модели предметной области, для которой разрабатывается программа. Такая модель необходима заказчикам, программистам и менеджерам проекта по созданию информационной системы для того, чтобы они могли выработать общий взгляд на цели и функции системы. И хотя модели предметной области, формируемые с помощью UML, предназначены, прежде всего, для последующей реализации в виде программного обеспечения, они имеют и самостоятельную ценность, т.к. позволяют наглядно отобразить функции и процессы бизнес-системы, объекты, участвующие в бизнес-процессах, их отношения, а также динамику выполнения процессов.

Начало работ над созданием унифицированного объектно-ориентированного языка моделирования относится к середине 1990-х годов. К тому времени уже было разработано более 50 различных языков объектно-ориентированного моделирования. Авторы наиболее распространенных языков – Г. Буч, Д. Румбах и А. Джекобсон, – собравшись «под крылом» компании Rational Software Corporation, начали работу над унифицированным методом. Ими был создан ряд версий унифицированного метода, который они назвали Unified Modeling Language (UML). В настоящее время большинством производителей информационных систем и такими комитетами по стандартам, как ANSI и OMG, язык UML был признан в качестве стандарта.

В технологии реинжиниринга бизнес-процессов, пожалуй, впервые UML стали применять не только и не столько для создания информационных систем (ИС), сколько для анализа и перепроектирования бизнеса. Вместо моделей процессов, реализуемых информационной системой, строятся модели бизнес-процессов, даже если они и не будут подвергнуты автоматизации, вместо объектов ИС (программных объектов) в моделях отражаются объекты бизнеса (исполнители, продукция, услуги и т.д.), вместо окружения ИС (пользователей ИС) моделируется окружение бизнеса (поставщики, партнеры, клиенты).

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций (схем, графов), получивших название диаграмм. Предполагается, что никакая единственная модель не может с достаточной степенью адекватности описывать различные аспекты сложной системы. Таким образом, модель сложной системы состоит из некоторого числа диаграмм, каждая из которых отражает некоторый аспект поведения или структуры системы. В языке UML определены следующие виды диаграмм:

- диаграмма вариантов использования (Use case diagram);
- диаграмма состояний (State diagram);
- диаграмма деятельности (Activity diagram);
- диаграмма последовательности (Sequence diagram);
- диаграмма кооперации (Collaboration diagram);
- диаграмма классов (Class diagram);
- диаграмма компонентов (Component diagram);
- диаграмма развертывания (Deployment diagram).

Диаграмма вариантов использования представляет собой наиболее общую концептуальную модель системы, которая является исходной для построения всех остальных диаграмм. Представление вариантов использования детализируется с помощью диаграмм состояний, деятельности, последовательности и кооперации. Диаграммы классов используются для представления логической структуры информационной системы, диаграммы компонентов и диаграммы развертывания – для представления физических компонентов информационной системы.

3. Знакомство с основными возможностями Rational Rose

Для автоматизации проектирования и создания сложных информационных систем используются так называемые CASE-средства. Термин CASE расшифровывается как Computer-Aided Software Engineering – компьютерная поддержка проектирования программного обеспечения. Современные CASE-средства поддерживают весь жизненный цикл создания информационной системы – от создания графических моделей автоматизируемой предметной области (моделей бизнес-процессов) до проектирования баз данных и автоматического генерирования программного кода. Однако в настоящее время акцент стал смещаться с проектирования компонент ИС на анализ автоматизируемой предметной области, на моделирование сложных систем широкого назначения. Неслучайно аббревиатура CASE все чаще стала расшифровываться как Computer Aided System Engineering – компьютерная поддержка проектирования систем.

Среди CASE-средств, ориентированных на объектно-ориентированные методы, наиболее популярным является средство Rational Rose компании Rational Software Corporation. Rational Rose позволяет строить объектные модели проектируемой системы в различных нотациях (UML, нотация Буча, метод ОМТ Румбаха) и генерировать на основе полученной модели приложения на языках программирования C++, Visual Basic, Power Builder, Java, Ada, Smalltalk и др. Rational Rose содержит все диаграммы UML.

В рамках Rational Rose существуют различные программные инструментари. Базовым средством, используемым для построения диаграмм, является Rational Rose Enterprise Edition. После запуска этой программы открывается главное окно, показанное на рис. 4.1.

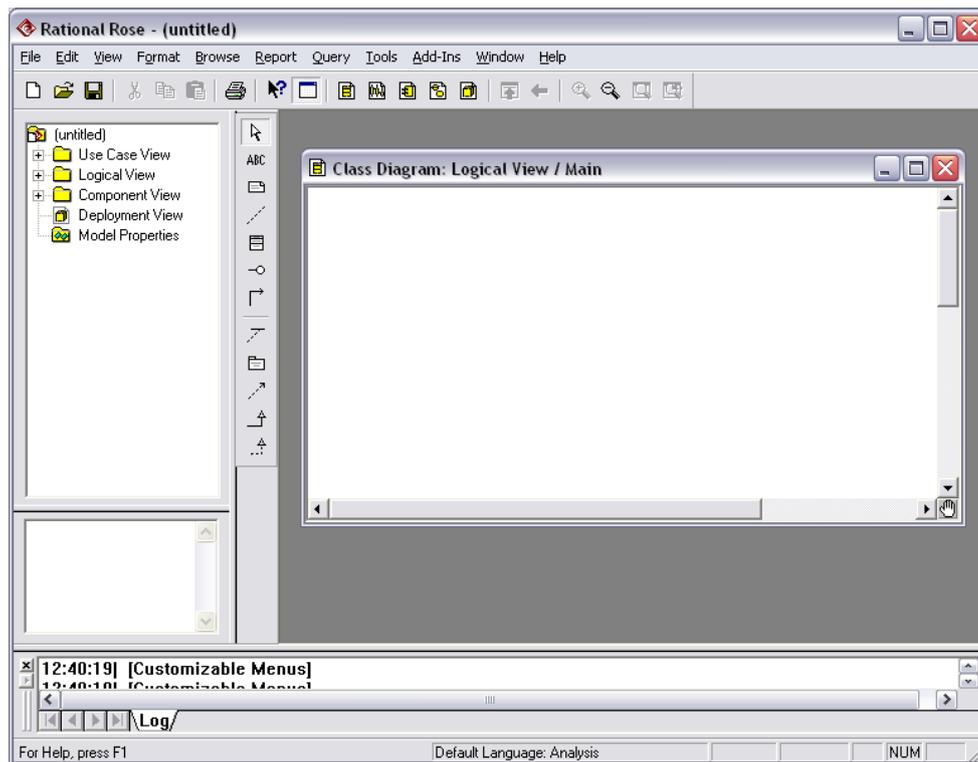


Рис. 4.1. Главное окно Rational Rose

В верхней части окна находится меню и стандартная панель инструментов (Tool Bar). В левой части главного окна находится окно браузера (Browser), предназначенное для быстрого доступа к диаграммам. Любой элемент, который разработчик добавляет в модель, сразу отображается в этом окне. Оно позволяет легко перемещаться по дереву диаграмм, буксировать диаграммы мышкой и изменять структуру модели по своему усмотрению.

В правой части главного окна находятся окна диаграмм (Diagram), открытых в текущий момент. Обычно это поле называется рабочим столом Rational Rose. Внизу рабочего стола находится окно протокола (Log). В нем Rational Rose постоянно фиксирует все действия, произведенные над диаграммами. Между окном браузера и окном диаграммы находится панель инструментов текущей диаграммы (специальная панель), вид которой зависит от типа выбранной диаграммы.

В нижней левой части главного окна, под окном браузера находится окно документации (Documentation). В этом окне можно записывать самую различную информацию о выделенном в текущий момент элементе диаграммы.

При запуске программы кроме главного окна на экране появится диалоговое окно Create New Model. В нем будет отражен список «каркасных» моделей, которые можно взять за основу при создании новой модели. Поскольку мы будем строить модель «с нуля», нажмите кнопку Cancel в этом окне. Автоматически будет создан новый проект.

В окне браузера в виде дерева будет отражена структура проекта, включающая три представления:

- представление вариантов использования (Use case view),
- логическое представление (Logical View),
- представление компонент (Component view).

Все эти представления отражены в виде папок (пакетов). Каждое из них, в свою очередь, содержит пустые диаграммы, созданные по умолчанию. Вы можете выбрать

любую папку, «раскрыть» ее (щелчком мыши на квадратике с '+'), активизировать уже имеющиеся элементы (визуализировать в окне диаграммы), добавить новые диаграммы (через всплывающее меню, активизируемое щелчком правой кнопки мыши).

По окончании сеанса работы над проектом выполненную работу необходимо сохранить в файле проекта с расширением mdl. Это можно сделать через меню File→Save. В дальнейшем в начале нового сеанса можно открыть этот проект для последующей модификации через меню File→Open.

4. Построение диаграммы прецедентов (Use case)

Представление вариантов использования показывает, как система должна выглядеть «извне», т.е. оно отражает функции системы и ее взаимодействие с внешним окружением. Основное внимание здесь уделяется представлению высокого уровня, отображающему, что система должна делать, а не как она будет делать это. Представление вариантов использования является исходным концептуальным представлением системы для последующей детализации.

В окне браузера щелкните мышью на квадратике с '+' рядом с пакетом Use case view, чтобы раскрыть пакет представления вариантов использования, и дважды щелкните на пиктограмме Main (главная).

При этом на рабочем столе появится окно диаграммы вариантов использования. Специальная панель инструментов примет вид, соответствующий данному виду диаграмм. Назначение инструментов панели поясняется в таблице 4.1.

Таблица 4.1

Инструмент	Наименование	Назначение
	Selection Tool	предоставляет возможность выделять объект
	Text Box	добавляет текст к диаграмме
	Note	добавляет к диаграмме примечание
	Anchor Note to Item	связывает примечание с объектом на диаграмме
	Package	помещает на диаграмму новый пакет
	Use case	помещает на диаграмму новый прецедент
	Actor	помещает на диаграмму нового актора
	Unidirectional Association	рисует направленную ассоциацию между актором и прецедентом
	Dependency or Instantiates	рисует отношение зависимости между элементами диаграммы
	Generalization	рисует отношение обобщения

Основными элементами диаграммы являются прецедент (вариант использования, use case) и актер (действующее лицо, actor).

Вариантом использования (Use Case) или прецедентом называется законченная совокупность действий моделируемой системы, начинающаяся при получении стимула извне и заканчивающаяся предоставлением некоторого продукта или сервиса актору – пользователю системы. При моделировании информационной системы вариант использования соответствует отдельному сервису, предоставляемому ИС пользователю. При моделировании бизнеса варианты использования ставятся в соответствие бизнес-процессам.

Чтобы поместить прецедент на диаграмму на панели инструментов выберите инструмент Use Case (см. табл. 4.1) и щелкните мышью в окне диаграммы на любом свободном месте. На диаграмме появится изображение прецедента с маркерами изменения его геометрических размеров и предложенным программой именем по умолчанию. Вы можете поменять имя: щелкните клавишей мыши на выделенном элементе и введите новое имя (например, «Продажа») в поле под элементом.

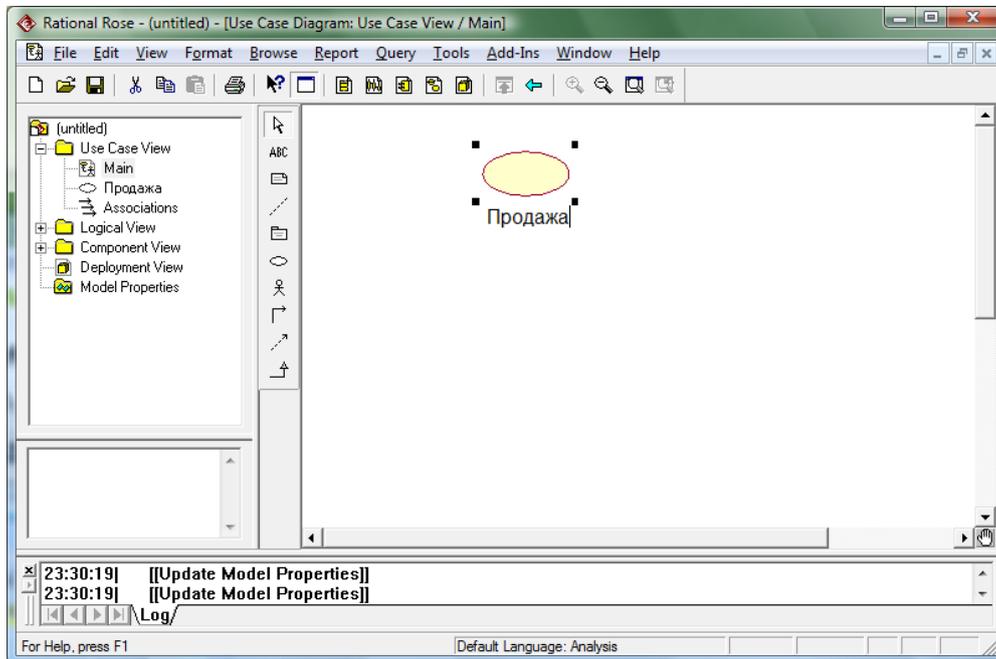


Рис. 4.2. Размещение прецедента

Переименовать можно и другим способом: щелкните на выделенном элементе двойным щелчком, чтобы вызвать окно спецификации, и введите новое имя в поле Name. Спецификация прецедента содержит также стереотип, краткое описание, перечень связанных с прецедентом поддиаграмм и документов. В поле Stereotype выберите из списка стереотип *business use case* (рис. 4.3) и щелкните ОК. Изображение прецедента на диаграмме несколько поменяется.

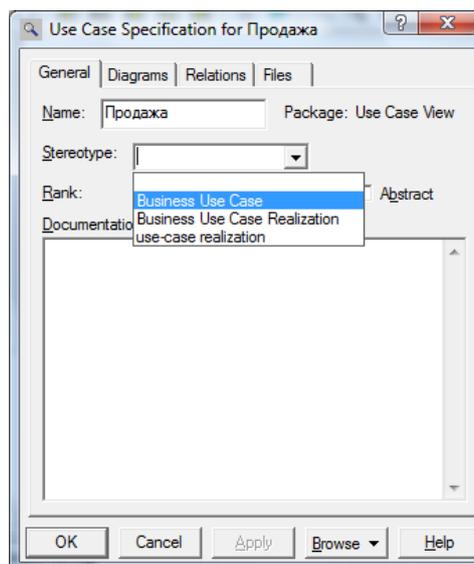


Рис. 4.3. Спецификация прецедента

Актор (Actor) или **действующее лицо** моделирует субъект окружения системы. Это любые сущности, взаимодействующие с системой. Примеры акторов бизнес-систем: *Клиент, Заказчик, Поставщик, Партнер* и т.д. Акторы взаимодействуют с системой через варианты использования.

Чтобы создать актора, выберите инструмент Actor и щелкните мышью в том месте диаграммы, куда Вы хотите его поместить. Имя актора задается так же, как имя варианта использования. В окне спецификации актора выберите стереотип *business actor*, можете ввести также его краткое описание.

После того, как Вы создадите элемент (прецедент, актора), его можно перемещать, удалять, изменять размеры. В любой момент Вы можете активизировать любой из созданных элементов диаграммы и производить с ним дальнейшие действия: перемещать с помощью мыши; «растягивать», потянув за маркер; удалять с помощью клавиши Delete или через всплывающее меню Edit/Delete. Все создаваемые Вами элементы будут отображаться также в окне браузера (будут помещены в дерево диаграммы Use case View/Main).

Взаимосвязи между вариантами использования и акторами называются **отношениями ассоциации**. К ним, в частности относятся отношения **коммуникации**. При моделировании бизнеса они отражают материальные и информационные потоки, заключающиеся в передаче вещества (сырья, инструментов, продукции и т.д.) и информации.

Чтобы установить отношение ассоциации между актором и прецедентом выберите инструмент Unidirectional Association и, не отпуская кнопки мыши, перемещайте указатель от одного элемента к другому. Щелкните двойным щелчком на выделенной ассоциации и в появившемся диалоговом окне введите имя ассоциации в поле Name. Можно (не обязательно) указать стереотип *communicate*.

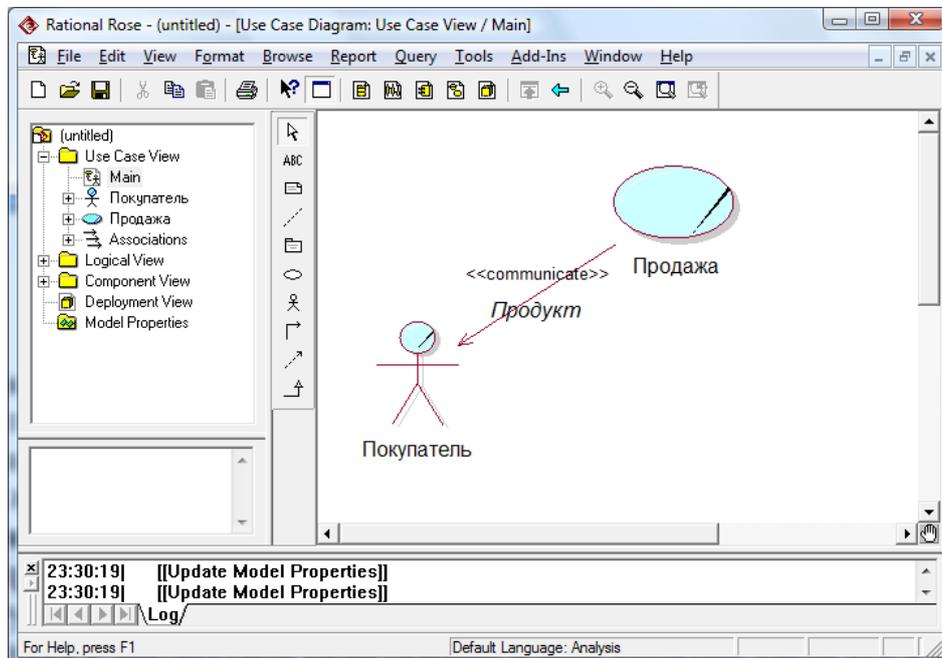


Рис. 4.4. Установление ассоциации между прецедентом и актором

Между вариантами использования отношения ассоциации недопустимы, т.к. каждый из них описывает самостоятельный законченный процесс. Между акторами также ассоциации обычно не указываются, так как с точки зрения бизнес-системы они не представляют интереса.

Диаграмма может содержать несколько вариантов использования и акторов.

Если моделируемый бизнес-процесс существует в нескольких *версиях*, то каждой из них нужно сопоставить отдельный прецедент. Например, прецедент «Продажа» имеет две версии: «Продажа готового продукта» и «Продажа заказного продукта». В этом случае на диаграмму нужно поместить соответствующие прецеденты и установить отношения обобщения между прецедентом «Продажа» и каждым из прецедентов-версий (см. рис. 4.5).

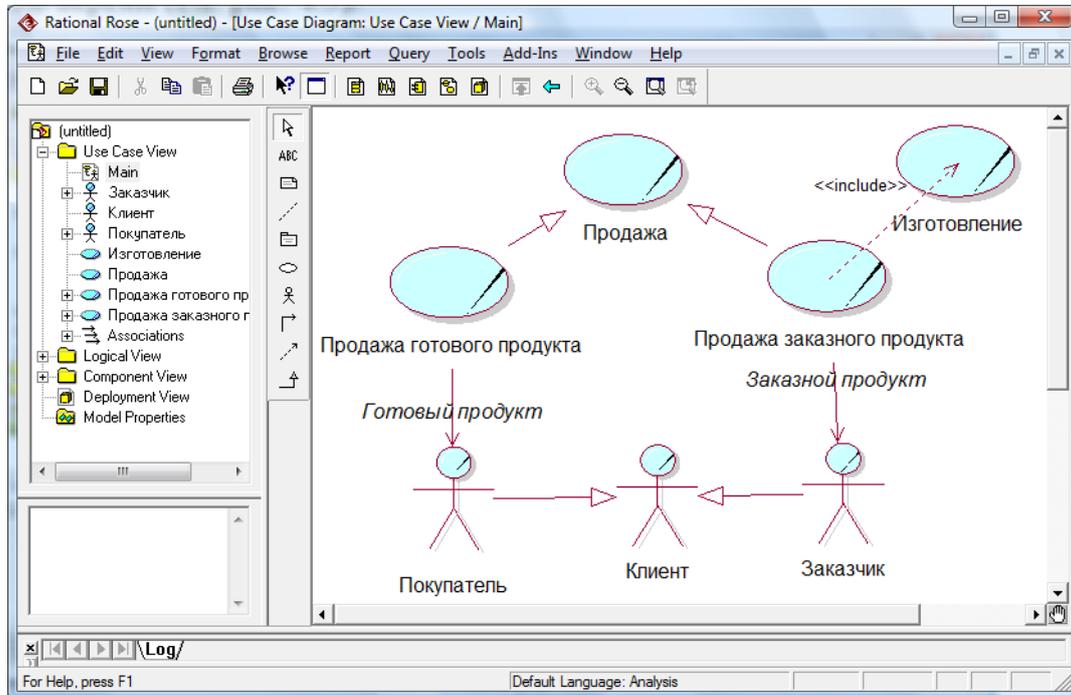


Рис. 4.5. Установление отношений обобщения и включения между прецедентами

Отношение обобщения между прецедентами означает, что один из процессов является частным случаем другого, более общего процесса. Отношения обобщения могут быть установлены и между акторами.

Чтобы установить отношение обобщения выберите на панели инструментов инструмент Generalization и, не отпуская кнопки мыши, перемещайте указатель от одного элемента (прецедента-версии) к другому (обобщенному прецеденту).

Отношения обобщения могут быть установлены и между акторами. Например, с прецедентом «Продажа готового продукта» взаимодействует актер *Покупатель*, а с прецедентом «Продажа заказного продукта» – актер *Заказчик*, можно добавить обобщенный актер *Клиент*, обозначающий более широкое понятие, чем покупатель или заказчик.

Между вариантами использования могут устанавливаться и другие отношения, в частности, **отношения зависимости**, означающие, что один из них некоторым образом зависит от другого. Разновидностями отношений зависимости являются **отношение включения (include)**, означающее, что один процесс является частью другого, и **отношение расширения (extend)**, означающее, что один процесс расширяет (дополняет) другой. Отношения обобщения, включения, расширения называют отношениями структурирования прецедентов, т.к. они позволяют упростить описание сложного процесса с множеством разветвлений в ходе событий, сведя его к нескольким более простым прецедентам.

Если моделируемый бизнес-процесс содержит относительно *самостоятельный фрагмент*, то этот фрагмент можно выделить в отдельный прецедент, связанный с исходным прецедентом отношением включения или расширения. Чтобы установить между прецедентами отношение включения выберите инструмент Dependency or Instantiates (отношение зависимости) и, не отпуская кнопки мыши, перемещайте указатель от исходного прецедента к прецеденту-фрагменту. Затем щелкните двойным щелчком на выделенном отношении зависимости, в появившемся диалоговом окне в поле Stereotype впишите или выберите из выпадающего списка стереотип *include* и закройте окно по кнопке ОК. Аналогично устанавливается отношение расширения, только в этом случае указывается стереотип *extend*.

Последним этапом создания диаграммы является *документирование* объектов диаграммы. Документация (поясняющий текст) вносится в окне Documentation, находящемся под окном браузера. Выделите некоторый объект диаграммы (вариант использования, актер, отношение) и в окне Documentation введите поясняющий текст (рис. 4.6). Каждый раз, когда Вы будете выделять этот объект, в окне документации будет отображаться введенный Вами текст. Документацию для любого объекта можно также внести в окне спецификации, вызываемом двойным щелчком мыши на выделенном объекте.

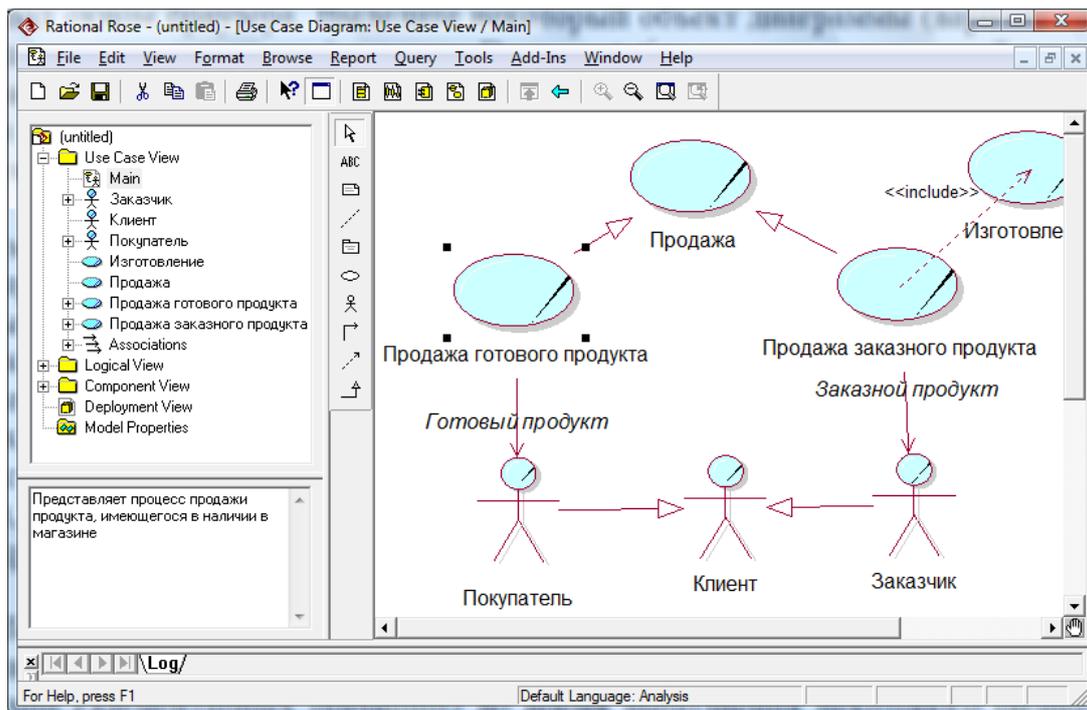


Рис. 4.6. Документирование прецедента

Вы можете также разместить на диаграмме примечания с помощью инструмента Note и привязать его к некоторому элементу с помощью инструмента Anchor Note to Item. Вы можете также создать произвольную надпись на диаграмме, не привязанную ни к какому элементу. Это можно сделать с помощью инструмента Text Box.

В результате Вы должны получить диаграмму, содержащую: прецеденты (не менее трех), акторы, отношения коммуникации между акторами и прецедентами, отношения структурирования (обобщения, включения или расширения) между прецедентами и, возможно, отношения обобщения между акторами.

Сохраните проект с помощью меню File→Save.

5. Построение диаграммы деятельности (Activity diagram)

Данный тип диаграмм может использоваться для отражения последовательности действий (элементарных операций) во время выполнения некоторого бизнес-процесса, представленного на диаграмме вариантов использования. Традиционно для этой цели использовались блок-схемы или структурные схемы алгоритмов.

Так как диаграмма последовательности раскрывает последовательность действий (событий) при выполнении некоторого прецедента, то будем создавать ее как поддиаграмму (Sub Diagram) этого прецедента.

В окне диаграммы Use case выберите прецедент, для которого будете создавать диаграмму деятельности. Установите курсор на этот прецедент и активизируйте всплывающее меню щелчком правой кнопки мыши. Выберите в меню Sub Diagrams → New Activity Diagram (рис. 4.7).

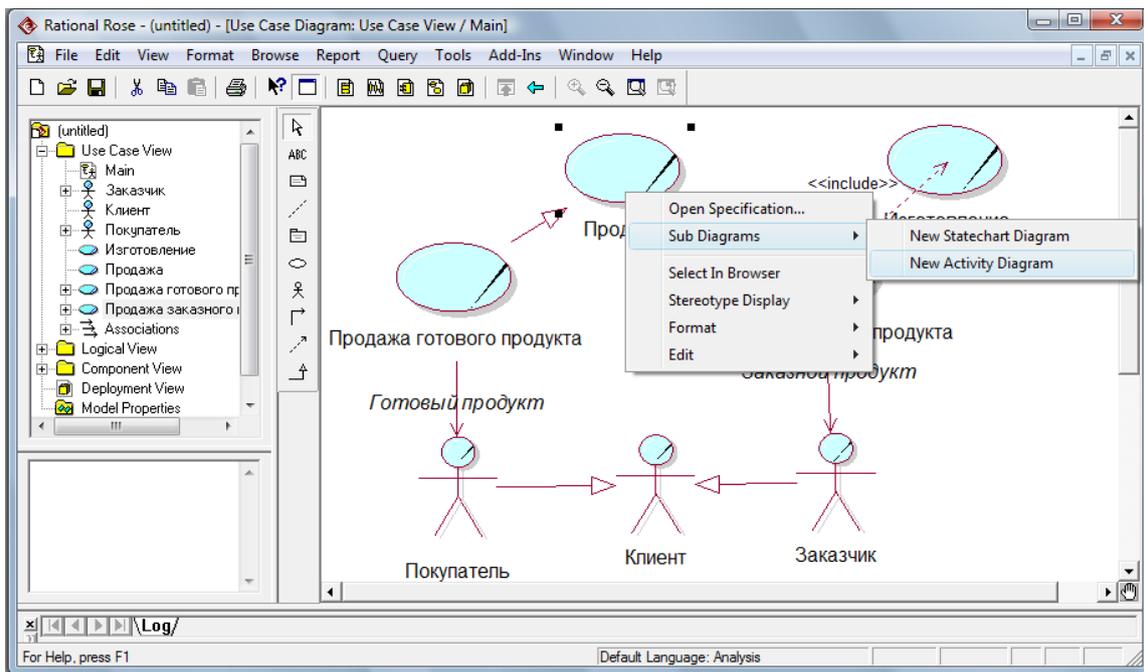


Рис. 4.7. Создание диаграммы деятельности

На рабочем столе появится пустое окно диаграммы деятельности. Специальная панель инструментов примет вид, соответствующий данному виду диаграмм. Назначение инструментов панели поясняется в таблице 4.2.

Пример диаграммы деятельности приведен на рис. 4.8. Процесс начинается с **начального состояния (Start State)** и переходит от одного **действия (Activity)** к другому, заканчиваясь **конечным состоянием (End State)**. Обычно действия располагают в порядке следования сверху вниз. Действия могут чередоваться с **состояниями (State)**. Состояния являются, с одной стороны, результатами определенных действий, с другой стороны, стимулами для выполнения следующих действий.

Переход (Transition) между действиями или между состоянием и действием изображается сплошной линией со стрелкой.

Если после выполнения некоторого действия процесс должен **разделиться** на альтернативные ветви в зависимости от некоторого условия, то ставится знак ветвления (**Desizion**) в виде ромба, имеющего одну входящую стрелку и две или более выходящих. Для каждой из выходящих стрелок указывается соответствующее условие, при котором выполняется данный переход.

Инструмент	Наименование	Назначение
	Selection Tool	предоставляет возможность выделять объект
	Text Box	добавляет текст к диаграмме
	Note	добавляет к диаграмме примечание
	Anchor Note to Item	связывает примечание с объектом на диаграмме
	State	помещает на диаграмму новое состояние
	Activity	помещает на диаграмму новое действие
	Start State	помещает на диаграмму новое начальное состояние
	End State	помещает на диаграмму новое конечное состояние
	Tranzition	рисует переход
	Tranzition to Self	рисует рефлексивный переход
	Horizontal Sinchronization	помещает на диаграмму новый символ горизонтальной синхронизации
	Vertical Sinchronization	помещает на диаграмму новый символ вертикальной синхронизации
	Desizion	помещает на диаграмму новое ветвление
	SwimLane	помещает на диаграмму новую дорожку

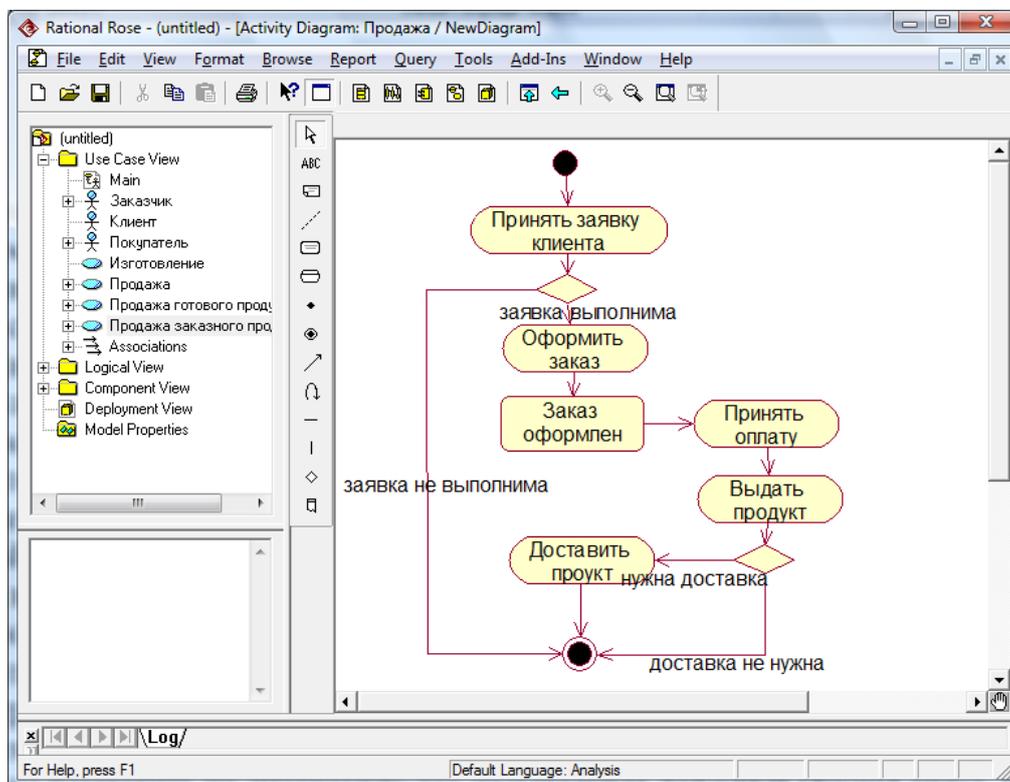


Рис. 4.8. Пример диаграммы деятельности

Для того чтобы поместить на диаграмму некоторый элемент (начальное или конечное состояние, действие, ветвление и т.д.), выберите на панели инструментов нужный инструмент и щелкните мышью в окне диаграммы в том месте, куда Вы хотите поместить элемент.

На диаграмме появится изображение элемента с маркерами изменения его геометрических размеров. Вы можете перемещать элемент, менять его размеры, удалить. Внутри элемента Activity или State нужно ввести описание действия или состояния.

Для того чтобы установить **переход** между элементами диаграммы, на панели инструментов выберите инструмент State Transition, затем, не отпуская кнопки мыши, перемещайте указатель от одного элемента к другому. Если необходимо нарисовать переход в виде ломаной линии, то рисовать нужно сегментами: проведите, не отпуская кнопки мыши, один сегмент (отрезок прямой), затем отпустите кнопку, опять нажмите и рисуйте следующий сегмент и т.д.

Стрелки перехода, выходящие из **ветвления** (Desizion), должны сопровождаться текстом, поясняющим условия, при которых выполняется тот или иной переход. Чтобы поместить текст возле стрелки перехода (условие перехода), щелкните на выделенной стрелке двойным щелчком и в открывшемся диалоговом окне в поле Event введите текст.

Для того чтобы показать, что разные исполнители могут параллельно выполнять действия, используется символ **синхронизации** в виде отрезка горизонтальной или вертикальной линии (**Horizontal Synchronization** или **Vertical Synchronization**). Он означает либо разделение на параллельные потоки (в этом случае в него входит один переход, а выходит – несколько), либо слияние параллельных потоков (в этом случае в него входят несколько переходов, а выходит – один). Пример использования синхронизации приведен на рис. 4.9.

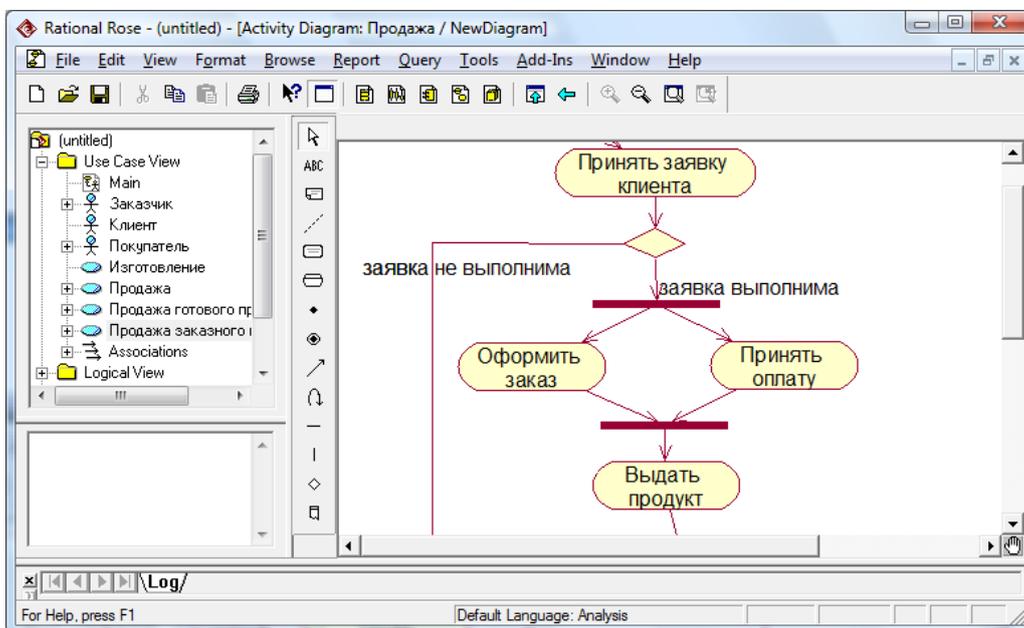


Рис. 4.9. Использование символа синхронизации

Можно поместить на диаграмму, так называемые **дорожки** (SwimLanes). Имеется в виду визуальная аналогия с плавательными дорожками в бассейне. Поле диаграммы разделено на «дорожки», соответствующие различным исполнителям. При этом все действия, выполняемые определенным исполнителем, помещаются на соответствующую дорожку (см. рис. 4.10).

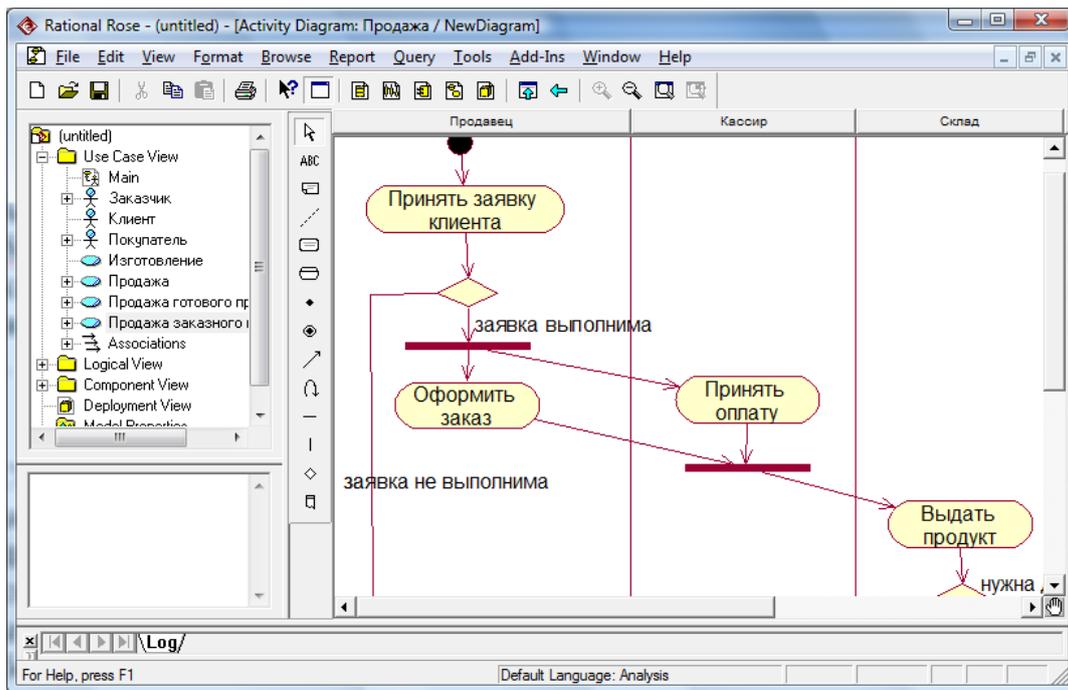


Рис. 4.10. Диаграмма с дорожками

Чтобы поместить на диаграмму дорожку выберите на панели инструментов инструмент SwimLane и щелкните мышью в окне диаграммы. Затем введите наименования дорожки в ее верхней части. Переместите на дорожку те действия, которые выполняются соответствующим исполнителем. Добавьте столько дорожек, сколько необходимо.

6. Связь диаграмм деятельности прецедентов, связанных отношениями обобщения или включения

На диаграмме Use case, могут быть представлены прецеденты, связанные друг с другом отношениями обобщения или включения. Их диаграммы деятельности, отражающие описание хода событий, должны быть связаны между собой.

Если для нескольких прецедентов, имеющих похожее поведение, был введен **обобщенный прецедент**, то диаграммы деятельности для прецедентов-потомков могут быть построены на основе диаграммы деятельности родительского (обобщенного) прецедента. В этом случае общее поведение описывается только один раз. Описания конкретных прецедентов (потомков) содержат только дополнительные шаги (или модифицированные шаги), которых нет в обобщенном описании.

Создайте диаграмму деятельности для обобщенного прецедента, включив в нее общие шаги, которые выполняются во всех прецедентах-потомках. Например, диаграмма обобщенного прецедента «Продажа», имеющего потомков «Продажа готового продукта» и «Продажа заказного продукта», содержит шаги, которые выполняются вне зависимости от того какой продукт (готовый или заказной) продается. Выделите с помощью левой кнопки мыши все элементы диаграммы и скопируйте выделенный фрагмент в буфер (выбрав в контекстном меню опцию Copy).

Перейдите на окно диаграммы деятельности для прецедента-потомка. Вставьте из буфера сохраненную диаграмму родительского прецедента (выбрав в контекстном меню опцию Paste). Добавьте дополнительные шаги, выполняемые только в соответствующей версии процесса. Пример диаграммы деятельности, построенной на основе диаграммы обобщенного прецедента, приведен на рис. 4.11.

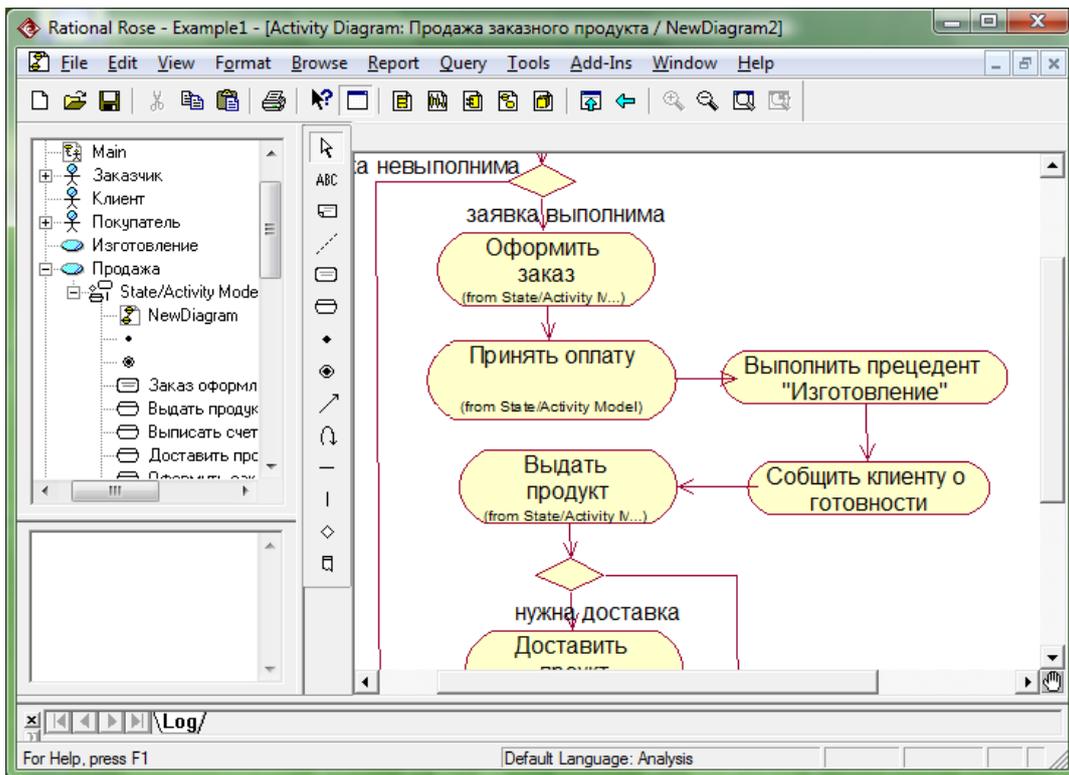


Рис. 4.11. Диаграмма деятельности прецедента, связанного с другими прецедентами отношениями обобщения и включения

Если фрагмент некоторого прецедента, представляющий относительно законченную последовательность событий, был выделен в отдельный прецедент, связанный с базовым прецедентом **отношением включения**, то в диаграмме деятельности базового прецедента должен быть вызов включенного прецедента. Поток событий включенного прецедента «встраивается» в поток событий базового прецедента, т. е. когда экземпляр базового прецедента в процессе своего выполнения достигает точки включения, выполняется последовательность шагов включенного прецедента, после чего продолжается выполнение исходного прецедента. Например, если прецедент «Продажа заказного продукта» включает прецедент «Изготовление», то все шаги, выполняемые в ходе изготовления продукции в соответствии с требованиями заказчика, помещаются во включенный прецедент.

Создайте диаграммы деятельности для обоих прецедентов, связанных отношением включения. В диаграмме для базового прецедента в том месте, где должен вызываться включенный прецедент, добавьте новое действие (Activity), внутри которого напишите: «Выполнить прецедент <Имя включенного прецедента>» (см. рис. 4.11).

7. Доработка модели

Завершите создание прецедентной модели бизнес-процесса, выбранного вами в качестве индивидуального задания. Модель обязательно должна содержать помимо основного прецедента, прецеденты, связанные с основным отношениями обобщения и/или включения. Для всех прецедентов должны быть созданы диаграммы деятельности.

Лабораторная работа №5 «Создание объектной UML-модели бизнес-процесса»

Цель работы: Получить практические навыки в построении объектной UML-модели бизнес-процесса средствами пакета Rational Rose.

Порядок выполнения работы.

1. Выбор задания.

В качестве индивидуального задания используется тот же бизнес-процесс, который Вы выбрали при выполнении предыдущей лабораторной работы. Объектная модель строится на основе созданной Вами прецедентной модели.

Откройте проект, созданный во время выполнения предыдущей лабораторной работы, через меню File→Open.

2. Построение диаграммы последовательности (Sequence diagram)

Прецедентная модель, включающая диаграмму вариантов использования и диаграммы деятельности, показывает основные процессы бизнес-системы и последовательность их выполнения. Однако для полного понимания системы такого описания недостаточно. Необходима модель, показывающая как, за счет чего реализуются процессы, т.е. какие объекты участвуют в выполнении процесса и как они взаимодействуют между собой и с окружением системы. Для этого служат диаграммы взаимодействия, к которым относятся диаграмма последовательности и диаграмма кооперации.

Так как диаграмма последовательности раскрывает последовательность взаимодействия объектов при выполнении одного из вариантов использования, то будем создавать ее в пакете представления вариантов. В окне браузера установите курсор на соответствующем прецеденте, откройте всплывающее меню щелчком правой кнопки мыши и выберите пункт New → Sequence Diagram (см. рис. 5.1). Дайте новой диаграмме имя. Дважды щелкнув на диаграмме в браузере, откройте ее.

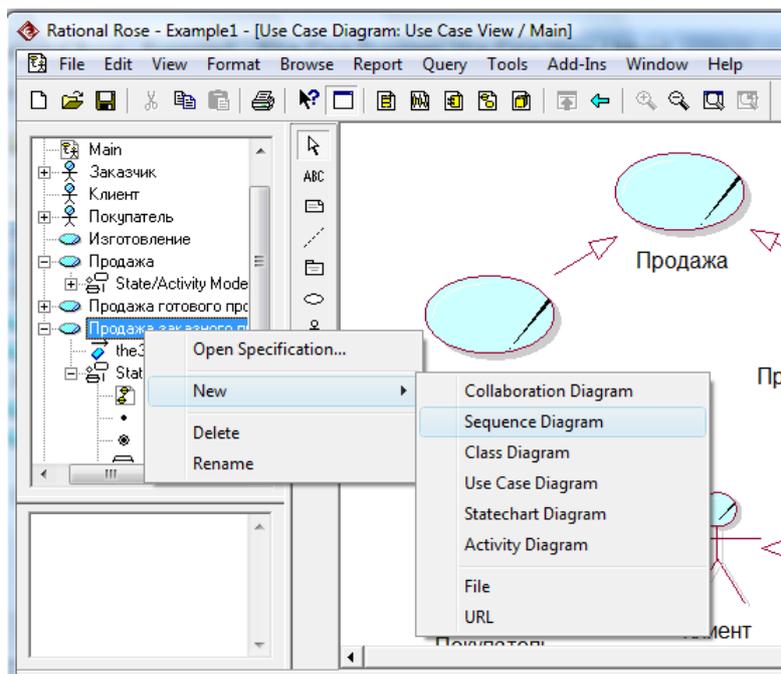


Рис. 5.1. Создание диаграммы последовательности

На рабочем столе появится окно диаграммы последовательности. Специальная панель инструментов примет вид, соответствующий данному виду диаграмм. Назначение инструментов панели поясняется в таблице 5.1.

Таблица 5.1.

Инструмент	Наименование	Назначение
	Selects or deselects an item	предоставляет возможность выделять объект
	Text Box	добавляет к диаграмме текст
	Note	добавляет к диаграмме примечание
	Anchor Note to Item	связывает примечание с элементом на диаграмме
	Object	помещает на диаграмму новый объект
	Object Message	рисует сообщение между двумя объектами
	Message to Self	Рисует рефлексивное сообщение (сообщение самому себе)

Построение диаграммы последовательности сводится к добавлению или удалению отдельных объектов и сообщений. Прежде всего, необходимо в верхней части диаграммы расположить в ряд всех активных участников выполнения прецедента. Например, для прецедента «Продажа заказного продукта» участниками могут быть: *Заказчик*, *Продавец*, *Кассир*, *Изготовитель* (бригада, занимающаяся изготовлением продукта), *Склад*, *Отправитель* (бригада, осуществляющая доставку).

Чтобы поместить на диаграмму последовательности **актера** (например, *Заказчика*), его следует перенести с диаграммы вариантов использования. Для этого в окне браузера раскройте диаграмму Main, чтобы показать элементы, включенные в эту диаграмму. Выберите нужный элемент (актера) и отбуксируйте его (оставляя нажатой левую кнопку мыши) в окно диаграммы. Расположите его в верхней левой части диаграммы.

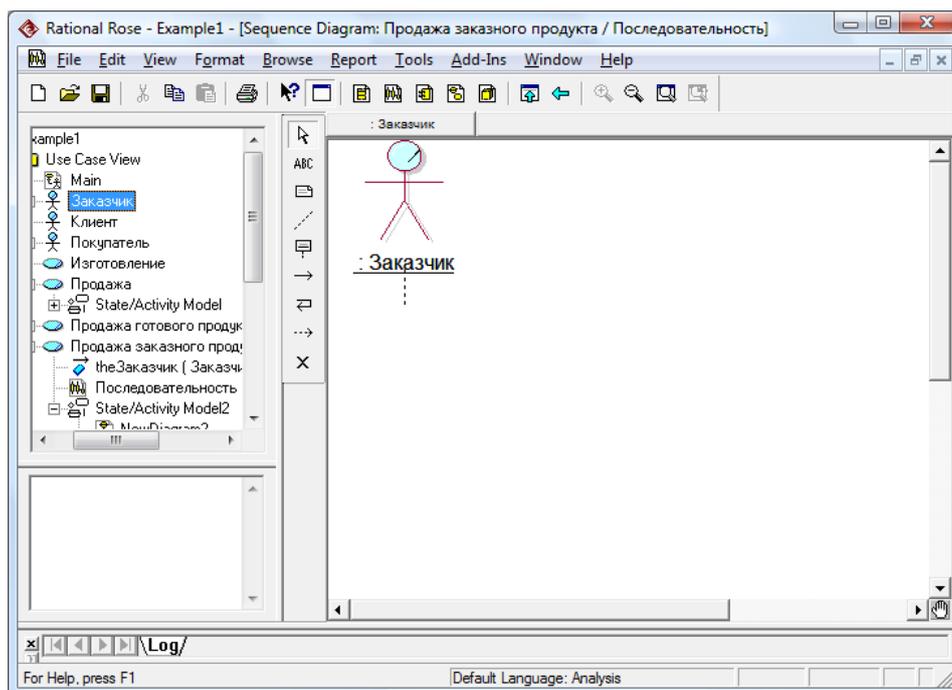


Рис. 5.2. Размещение актора на диаграмме последовательности

«Внутренние» участники бизнес-процесса называются **объектами (Object)**. Нужно помнить, что на диаграмме последовательности отображаются объекты, выполняющие конкретный экземпляр прецедента, например, конкретный продавец (его можно назвать *Продавец1*), являющийся представителем класса *Продавец*. Графически каждый объект изображается в виде прямоугольника, внутри которого записывается имя объекта и через двоеточие имя класса, к которому принадлежит объект (имя подчеркивается, что является признаком объекта). Чтобы создать объект, на панели инструментов выберите инструмент Object и щелкните мышью в верхней части диаграммы правее уже помещенных объектов (актеров). Щелкните на выделенном объекте и введите имя внутри прямоугольника (рис. 5.3).

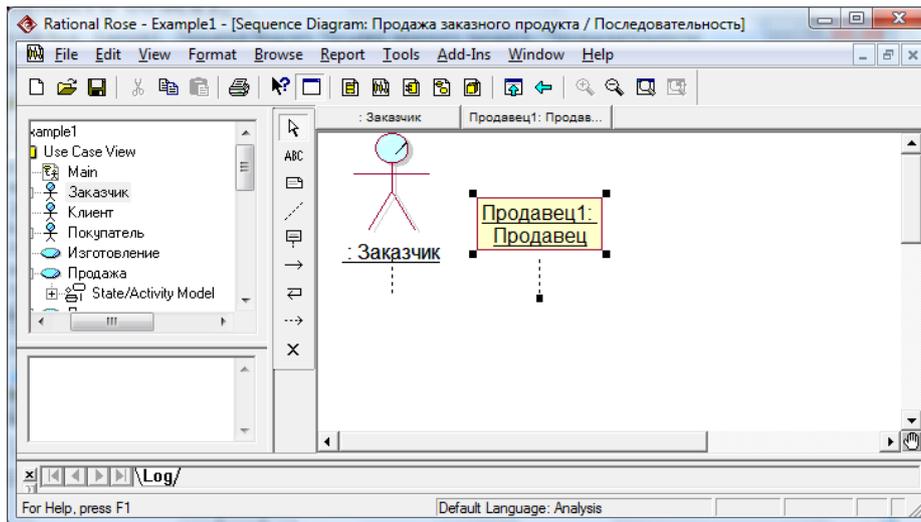


Рис. 5.3. Размещение объекта на диаграмме последовательности

Задать имя можно и другим способом: щелкните двойным щелчком на выделенном объекте и в появившемся диалоговом окне введите имя в поле Name. Кроме того, Вы можете задать класс объекта. Для этого в поле Class в выпадающем списке выберите подходящее имя класса или выберите <New> и в появившемся диалоговом окне введите имя класса. Там же можно задать стереотип класса - *business worker*. (при этом изображение объекта на диаграмме изменится).

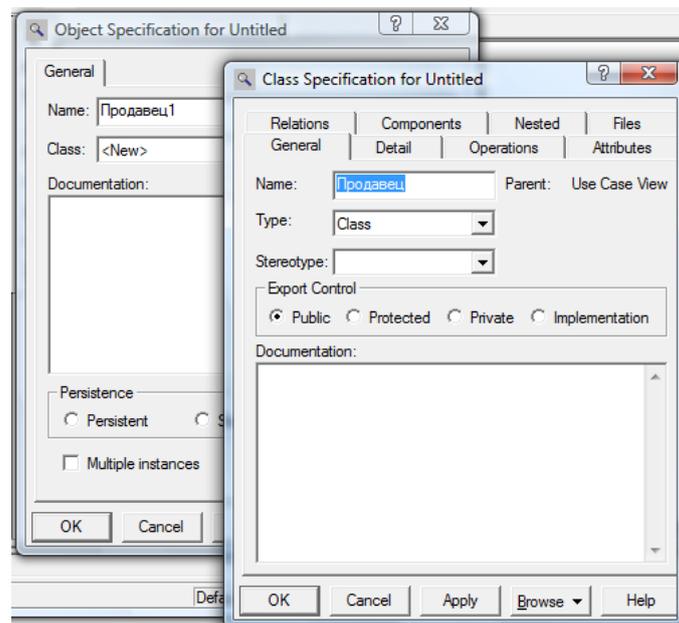


Рис. 5.4. Спецификации объекта и класса

Объекты и акторов следует располагать на диаграмме в порядке их активности: крайним слева изображается объект (актор), который является инициатором первого сообщения, правее – получатель сообщения и т.д. Здесь необходимо пояснить, что в UML взаимодействие объектов рассматривается, прежде всего, как коммуникация, т.е. передача сообщений (информации). Однако взаимодействие можно рассматривать и как передачу вещественных объектов (продукции, деталей, материалов), понимая под сообщением не только информацию, но и передачу некоторого пассивного объекта (объекта-сущности).

Передача сообщения осуществляется от объекта-инициатора к другому объекту. При этом сообщение инициирует выполнение определенных действий тем объектом, которому это сообщение отправлено. Поэтому объект, получающий сообщение, называют иногда исполнителем. При этом в разных ситуациях одни и те же объекты могут выступать и в качестве инициаторов, и в качестве исполнителей. Кроме того, объект может отправлять сообщение самому себе, т.е. он одновременно является и отправителем, и получателем сообщения.

От каждого активного объекта (или актора) вниз идет пунктирная линия, называемая *линией жизни* объекта. Передача **сообщения (Object Message)** изображается в виде горизонтальной стрелки с именем сообщения от линии жизни объекта-инициатора к линии жизни объекта-получателя (см. рис. 5.5). Последовательность взаимодействий отражается через последовательное их расположение на диаграмме сверху - вниз. Другими словами, сообщения, расположенные на диаграмме выше, инициируются раньше тех, которые расположены ниже.

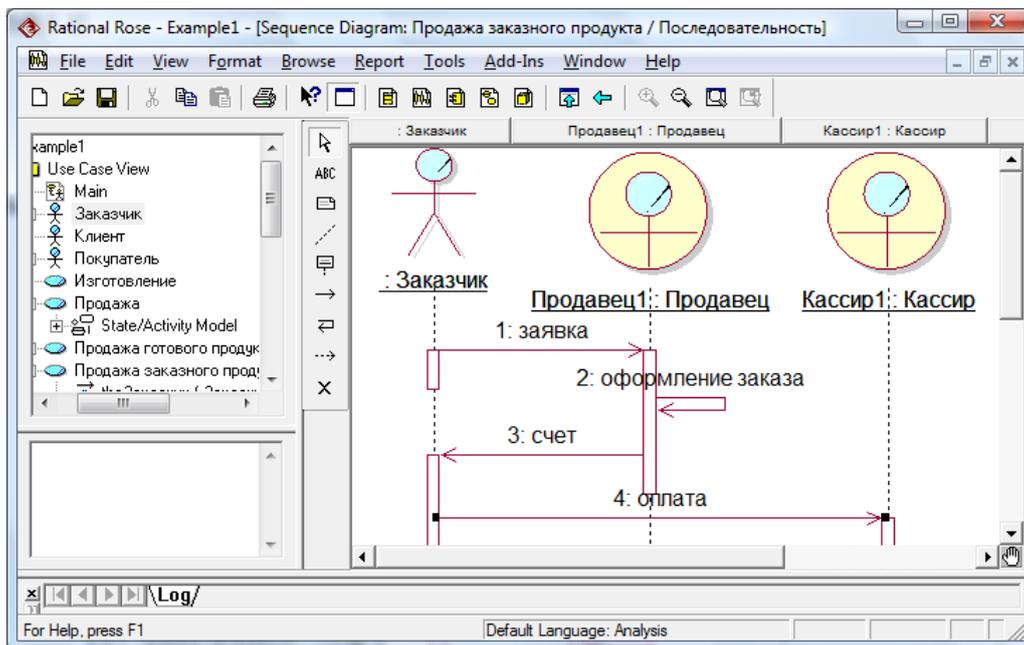


Рис. 5.5. Отображение взаимодействий между объектами

Для того чтобы отобразить взаимодействие между объектами, необходимо выполнить следующие действия:

- на панели инструментов выберите инструмент Message (сообщение);
- установите курсор на линии жизни объекта – инициатора сообщения в нужном месте, соответствующем последовательности передачи сообщения;
- не отпуская кнопки мыши, перемещайте указатель к линии жизни объекта-получателя сообщений.

Чтобы задать имя сообщения, щелкните двойным щелчком на выделенной линии сообщения и в открывшемся диалоговом окне введите имя. В дальнейшем можно переименовать сообщение: щелкнуть на нем мышью и ввести имя в поле над линией сообщения.

Чтобы отобразить сообщение, посылаемое объектом самому себе, нужно выбрать на панели инструментов инструмент Message to Self и щелкнуть мышью на линии жизни объекта в нужном месте, соответствующем последовательности передачи сообщения.

При необходимости можно изменить порядок следования сообщений. Для этого достаточно перетащить сообщение на новое место. При изменении порядка следования сообщений они автоматически перенумеровываются. Если Вы удалите ранее нарисованные сообщения, оставшиеся сообщения также будут автоматически перенумерованы.

В процессе работы с диаграммой Вам может понадобиться удалить некоторые объекты. При этом Rational Rose автоматически удалит также все сообщения, которые начинаются или заканчиваются на объекте, и переименует оставшиеся сообщения.

В процессе функционирования системы одни объекты могут находиться в активном состоянии, непосредственно выполняя определенные действия или в состоянии пассивного ожидания сообщений от других объектов. Чтобы явно выделить активность объектов, в UML применяется понятие «фокус управления». Фокус управления изображается в форме вытянутого узкого прямоугольника вдоль линии жизни, верхняя сторона которого обозначает начало активности, а нижняя – окончание активности.

Последним этапом создания диаграммы последовательности является документирование элементов диаграммы (объектов и сообщений). Документация (поясняющий текст) на активный элемент вносится в окне Documentation. На диаграмме могут также располагаться комментарии или примечания (Notes), которые могут ассоциироваться с отдельными объектами или сообщениями. Вы можете также разместить на диаграмме произвольный текст (с помощью инструмента Text Box).

Сохраните проект.

3. Построение диаграммы кооперации (Collaboration diagram)

Диаграмма кооперации, как и диаграмма последовательности, предназначена для визуализации взаимодействия участников процесса. Если диаграмма последовательности служит для отображения временных аспектов взаимодействия, то диаграмма кооперации акцентирует внимание на структурных аспектах взаимодействия объектов. Это статическая модель процесса, т. е. является мгновенным снимком объектов системы в некотором состоянии.

Так как диаграммы последовательности (Sequence) и кооперации (Collaboration) являются разными взглядами на одни и те же процессы, Rational Rose позволяет создавать из Sequence- диаграммы диаграмму Collaboration и наоборот, а также производит автоматическую синхронизацию этих диаграмм.

Находясь в окне с диаграммой последовательности, нажмите клавишу <F5>. Диаграмма кооперации будет создана автоматически на основе диаграммы последовательности. В дальнейшем с помощью клавиши <F5> Вы можете переключаться с одной диаграммы на другую. При этом изменения, вносимые в одну из диаграмм, будут автоматически вноситься и на другую диаграмму.

Пример диаграммы кооперации приведен на рис. 5.6.

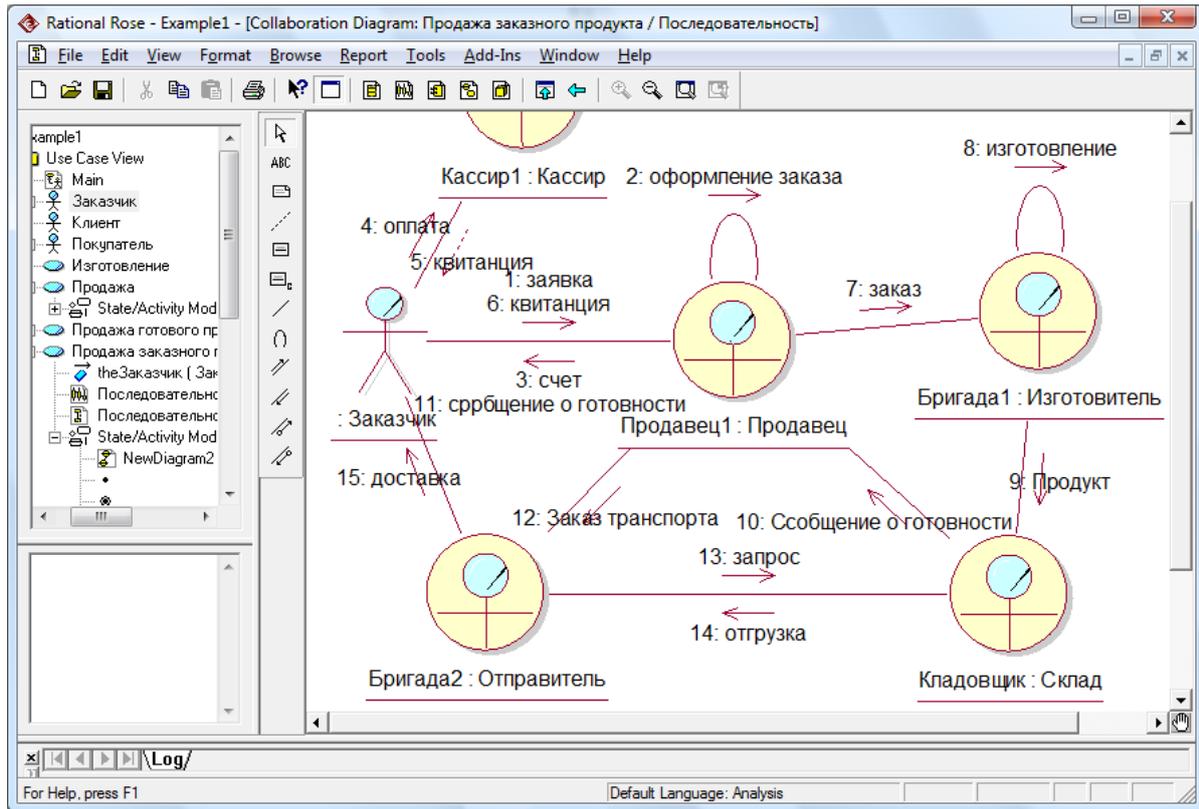


Рис. 5.6. Диаграмма кооперации

В окне диаграммы будут отражены объекты и сообщения, перенесенные с диаграммы последовательности. Можете активизировать любой объект и передвинуть его, чтобы диаграмма приобрела более красивый вид.

В отличие от диаграммы последовательности на диаграмме кооперации не указывается время в виде отдельного измерения. Поэтому последовательность взаимодействий может быть определена только с помощью порядковых номеров.

4. Построение диаграммы классов (Class diagram)

Диаграмма классов позволяет отразить классы объектов, участвующих в выполнении бизнес-процесса, и отношения между ними. При этом под классом понимается структура описания (набор характеристик) и шаблон поведения (набор операций) для некоторого множества реальных объектов, которые определяются на основе данного шаблона. Класс — это некоторая абстракция реального мира. Когда эта абстракция принимает конкретное воплощение, она называется объектом. Например, класс «Продавец» имеет набор характеристик (атрибутов), таких как «фамилия, имя, отчество», «стаж работы», «рабочее место» и т.д., а также набор операций, таких как «прием заявки клиента», «оформление заказа» и т.д. Описание конкретного продавца создается на базе данного класса и содержит конкретные значения атрибутов, характерные именно для этого продавца.

Создайте новую диаграмму классов для выбранного прецедента аналогично тому, как вы создавали диаграмму последовательности (см. п.2, рис. 5.1), только во всплывающем меню выберите пункт New → Class Diagram. Дважды щелкнув на диаграмме в браузере, откройте ее.

На рабочем столе появится окно диаграммы классов. Специальная панель инструментов примет вид, соответствующий данному виду диаграмм. Назначение некоторых инструментов панели поясняется в таблице 5.2.

Таблица 5.2.

Инструмент	Наименование	Назначение
	Class (Класс)	Добавляет на диаграмму новый класс
	Interface (Интерфейс)	Добавляет новый интерфейсный класс
	Association (Ассоциация)	Добавляет ненаправленную ассоциацию
	Aggregation (Агрегация)	Добавляет отношение агрегации
	Link Attribute (Атрибут отношения)	Связывает класс с отношением ассоциации
	Package (Пакет)	Добавляет на диаграмму новый пакет
	Dependency or instantiates (Зависимость или наполнение)	Добавляет отношение зависимости
	Generalization (Обобщение)	Добавляет отношение обобщения
	Realize (Реализация)	Добавляет отношение реализации
	Unidirectional Association (Однонаправленная ассоциация)	Добавляет однонаправленную ассоциацию

Чтобы создать новый **класс (Class)**, на панели инструментов выберите инструмент Class и щелкните мышью в любом месте окна диаграммы. Будет создан новый класс с именем NewClass. Замените это имя на новое, например, «Заказ». Если Вы при построении диаграммы последовательности создавали объекты и задавали для них класс, то можно перенести эти классы на диаграмму классов. Для этого в окне браузера раскройте диаграмму последовательности, выберите нужный объект и отбуксируйте его (оставляя нажатой левую кнопку мыши) в окно диаграммы.

Всю информацию о классе можно задать в окне спецификации класса. Вы можете переименовать класс, задать его стереотип, добавить атрибуты и операции. Для вызова окна щелкните двойным щелчком на выделенном классе. Пример окна спецификации класса приведен на рис. 5.7.

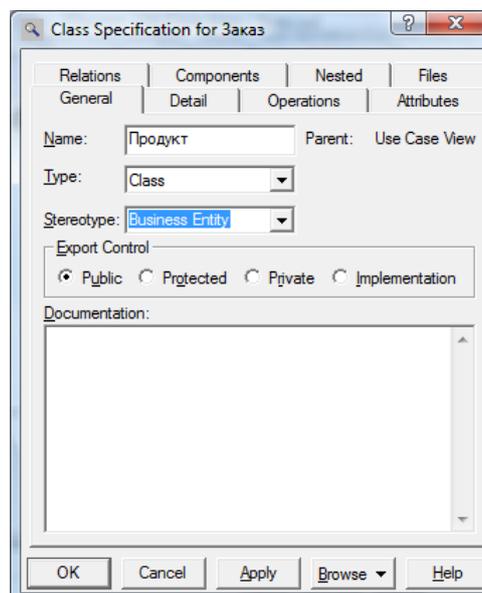


Рис. 5.7. Окно спецификации класса

Класс изображается как прямоугольник, разделенный на три части (класс *Заказ* на рис. 5.8): в верхней части записывается название класса, в середине — атрибуты, в нижней части — операции. Однако класс может иметь и другое изображение, связанное с его стереотипом. В модели бизнеса для классов используются два основных стереотипа: *business worker* (класс *Продавец* на рис. 5.8) и *business entity* (класс *Продукт* на рис. 5.8). Стереотип *business worker* используется для обозначения классов исполнителей, т.е. активных объектов, выполняющих прецеденты (например, *Продавец*, *Исполнитель*, *Склад*, *Отправитель*). Стереотип *business entity* используется для объектов-сущностей — пассивных объектов, используемых в ходе выполнения прецедента (например, *Заказ*, *Продукт*).

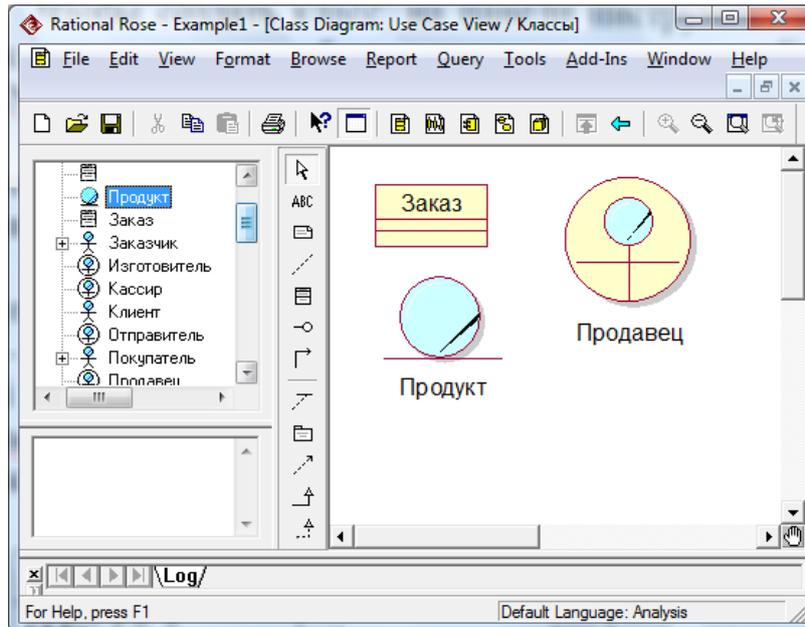


Рис.5.8. Различные изображения классов

Поскольку класс — это множество объектов, связанных общностью свойств и поведения, все объекты одного класса имеют схожие свойства. Свойства объектов описываются с помощью характеристик, называемых *атрибутами*. При этом состав атрибутов одинаков для всего класса, различные экземпляры одного класса отличаются лишь набором конкретных значений атрибутов. Например, класс *Продавец* может иметь атрибуты: *ФИО*, *Стаж*, *Квалификация* и т.д.

Для добавления атрибута щелкните правой кнопкой мыши на классе. В открывшемся меню выберите пункт *New Attribute* (новый атрибут) и введите имя атрибута. Можно вводить и другим способом. Откройте окно спецификации класса двойным щелчком мыши на выделенном классе. Перейдите на вкладку *Attributes (Атрибуты)*. Если у класса уже имеются атрибуты, они будут перечислены на этой вкладке. Щелкните правой кнопкой мыши где-нибудь внутри области атрибутов. В открывшемся меню выберите пункт *Insert* (Вставить). Введите имя нового атрибута в колонке *Name*. Можно ввести также тип данных (*Type*), например, *Integer* — целое число, *String* — строка, *Date* — дата. В колонке *Initial* можно ввести значение по умолчанию (рис. 5.9).

Удалить атрибут можно, выделив его имя и выбрав во всплывающем меню (по щелчку правой кнопки мыши) пункт *Delete*.

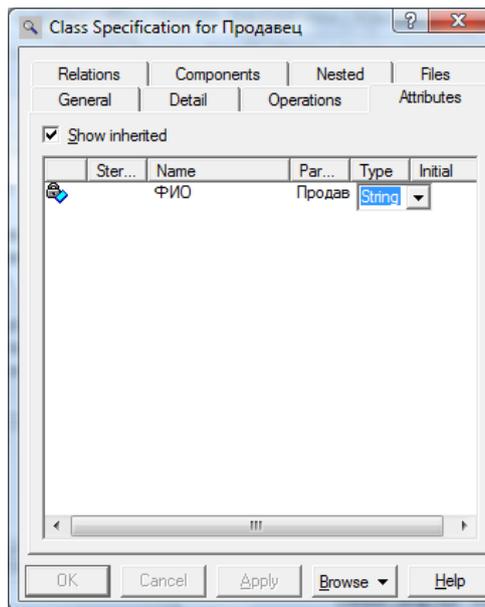


Рис.5.9. Добавление атрибута в окне спецификации класса

Операции класса вводятся аналогично. Перейдите на вкладку **Operations (Операции)** окна спецификации. Щелкните правой кнопкой мыши где-нибудь внутри области операций. В открывшемся меню выберите пункт Insert (Вставить). Введите имя новой операции в колонке Operation.

Идентифицируя операции, изучите все сообщения на диаграммах последовательности и кооперации. Анализ этих диаграмм помогает выявить все операции активного объекта. Так, из диаграммы, представленной на рис. 5.6, можно определить, что к обязательствам объекта *Продавец* относятся: *Прием заявки*, *Оформление заказа*, *Выдача счета*, *Передача заказа Изготовителю*, *Прием сообщения о готовности продукта*, *Сообщение клиенту о готовности продукта*, *Заказ транспорта*. Данные операции должны быть внесены в спецификацию соответствующего класса *Продавец*.

У классов объектов-сущностей, как правило, операции отсутствуют.

Поместите также на диаграмму классов акторов («отбуксировав» их из браузера), введите для них атрибуты и операции.

Для того чтобы атрибуты и операции класса (или актора) были видны на диаграмме, выделите класс, щелкните правой кнопкой мыши и во всплывающем меню выберите Options, Show All Attributes и Show All Operations.

Следующим этапом построения диаграммы классов является установление **отношений** между классами (рис. 5.10).

Между классами исполнителей (business worker) устанавливаются отношения **коммуникации** (ассоциации со стереотипом *communicate*). На диаграмме классов не нужна такая детализация взаимодействий, как на диаграммах последовательности и кооперации, достаточно лишь указать наличие ассоциации (коммуникации), т.к. здесь отображаются только статические устойчивые связи между объектами.

Чтобы установить отношение коммуникации выберите инструмент Association (Ассоциация) на панели инструментов. Проведите мышью линию ассоциации от одного класса исполнителей к другому (чтобы сделать связь двунаправленной, проведите еще одну линию в обратном направлении). Откройте окно спецификации отношения двойным щелчком на выделенном отношении, на вкладке General в поле Stereotype выберите стереотип «communicate».

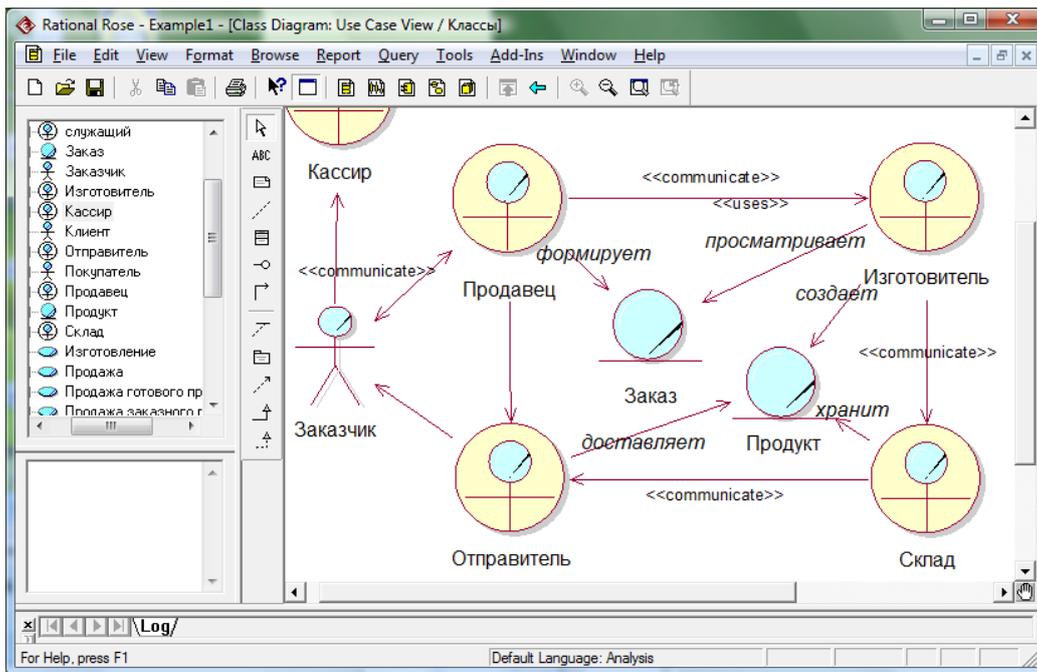


Рис.5.10. Отношения коммуникации и использования между классами

Между классами объектов-сущностей (business entity) отношения коммуникации не устанавливаются. Класс сущности может быть связан с классом исполнителя отношением **использования** (ассоциации со стереотипом *uses*) в случае, если исполнитель некоторым образом использует сущность. Например, Продавец *формирует* Заказ, Изготовитель *использует* Заказ для получения описания продукта, Изготовителем *создает* Продукт.

Чтобы установить отношение использования выберите инструмент Association (Ассоциация) на панели инструментов. Проведите мышью линию ассоциации от класса исполнителей к классу сущности. Откройте окно спецификации отношения, на вкладке General в поле Stereotype выберите (или введите) стереотип «uses». Введите имя отношения в поле имени (Name). При выборе имени отношения помните, что, как правило, это глагол или глагольная фраза, описывающая, зачем нужно отношение. Помимо (вместо) имени отношения можно ввести роли каждого из классов в этом отношении (см. рис. 5.11).

Рис.5.11. Введение роли для участников отношения

На диаграмме классов могут быть отражены также отношения структурирования – обобщения и включения (рис. 5.12). С помощью **обобщения** (*generalization*) показывают отношение наследования между двумя классами. Например, можно ввести абстрактный класс *Служащий*, являющийся обобщением исполнителей *Продавец*, *Кассир* и т.д. Обобщенный класс содержит атрибуты, общие для классов-потомков, например, «ФИО», «Стаж». Соответствующие атрибуты у классов-потомков можно удалить.

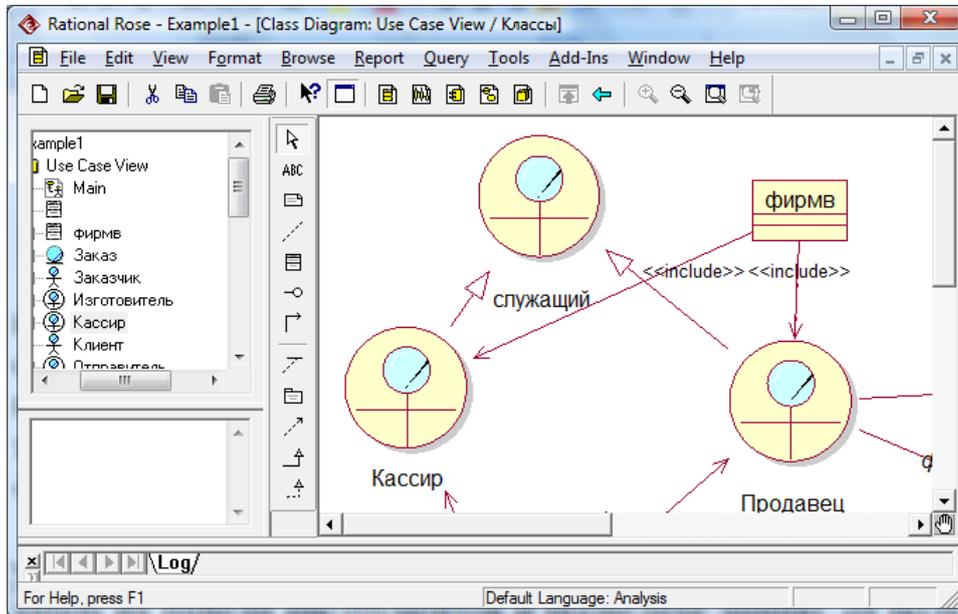


Рис.5.12. Отношения обобщения и включения между классами

Разместите на диаграмме обобщенный класс и введите для него атрибуты. Чтобы установить отношение обобщения между двумя классами, выберите кнопку *Generalization* на панели инструментов и проведите линию от подкласса (потомка) к обобщенному классу (предку).

Отношение **включения** (*include*) устанавливается между классами, один из которых является частью другого. Например, *Продавец* и *Кассир* являются частью *фирмы*, занимающейся изготовлением и продажей продукта (см. рис. 5.12).

В заключение можете документировать основные элементы диаграммы классов.

5. Доработка модели

Завершите создание объектной модели бизнес-процесса, выбранного вами в качестве индивидуального задания. Модель обязательно должна содержать диаграмму (несколько диаграмм) последовательности, диаграмму кооперации, диаграмму классов. Диаграммы последовательности и кооперации должны быть согласованы с диаграммами деятельности, построенными на предыдущей лабораторной работе. На диаграмме классов должны быть представлены классы исполнителей и объектов-сущностей, а также разнообразные отношения (коммуникации, использования, отношения структурирования).

Лабораторная работа №6 «Создание UML-модели информационной системы»

Цель работы: Получить практические навыки в построении UML-модели информационной системы поддержки бизнес-процесса средствами пакета Rational Rose.

Порядок выполнения работы.

1. Выбор задания.

Выберите бизнес-процесс, для поддержки которого Вы будете проектировать информационную систему (ИС). Бизнес-процесс должен включать операции, выполняемые в автоматизированном режиме. Модель ИС строится на основе объектной модели бизнес-процесса. В качестве индивидуального задания Вы можете использовать тот же бизнес-процесс, который Вы выбрали при выполнении предыдущих лабораторных работ, если этот процесс (или его фрагмент) может быть автоматизирован.

2. Построение объектной модели автоматизируемого бизнес-процесса

Создайте диаграмму последовательности для выбранного бизнес-процесса. Диаграмма обязательно должна содержать хотя бы один объект, относящийся к классу информационных систем. Это должен быть активный объект, связанный с другими участниками процесса отношениями сообщений, т.е. он должен получать и отправлять сообщения. Чем подробнее Вы опишете взаимодействие исполнителей бизнес-процесса с информационной системой, тем проще Вам будет построить ее модель. Пример диаграммы последовательности автоматизируемого процесса приведен на рис. 6.1.

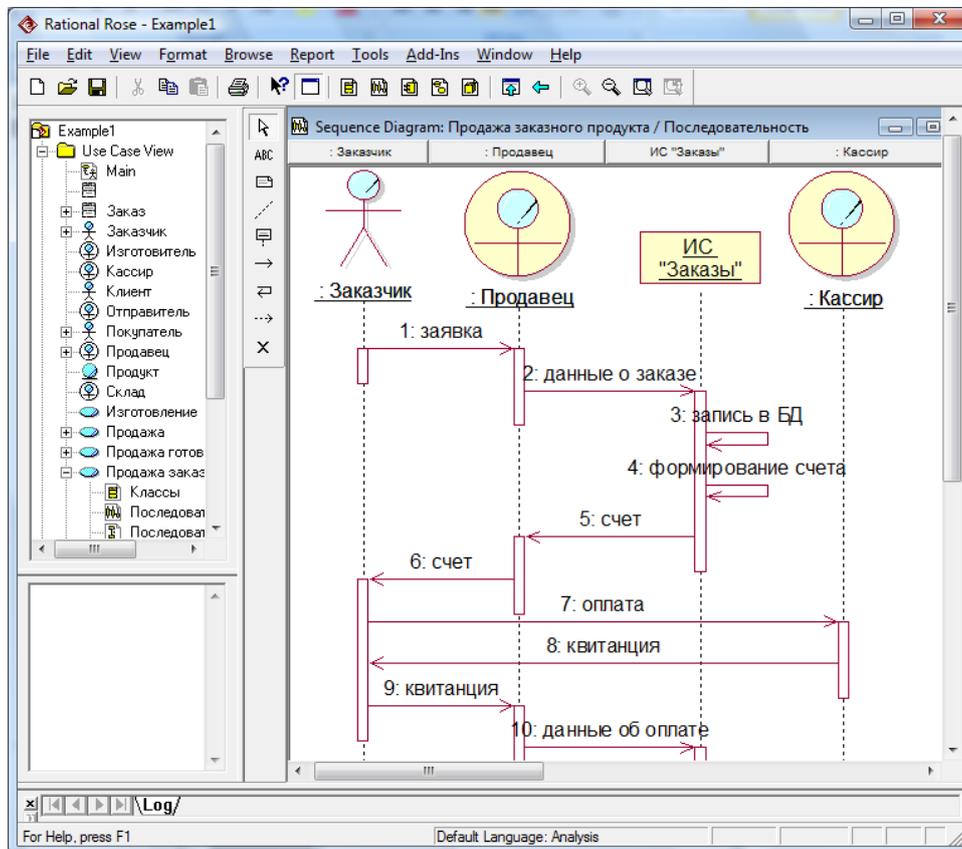


Рис.6.1. Диаграмма последовательности автоматизируемого бизнес-процесса

В бизнес-процессе может быть задействована не одна, а несколько информационных систем. В виде отдельных объектов могут быть представлены и отдельные модули ИС или отдельные приложения в составе пакета прикладных программ.

3. Создание диаграммы вариантов использования ИС

Создайте новую диаграмму use case, предназначенную для моделирования функций информационной системы. В окне браузера щелкните правой кнопкой мыши на папке Use case view (представление вариантов использования), выберите во всплывающем меню New → Use case Diagram и задайте имя новой диаграммы.

Прежде всего, необходимо определить **пользователей проектируемой ИС**. На основе анализа диаграммы последовательности бизнес-процесса выберите все активные объекты (или бизнес-акторы), использующие в своей деятельности информационную систему, т.е. взаимодействующие с соответствующим объектом. Каждому из таких объектов в модели информационной системы должен соответствовать актер — пользователь ИС. Поместите на диаграмму use case ИС соответствующих акторов (имя актора ИС не должно полностью совпадать с именем соответствующего объекта или бизнес-актора). Пример сопоставления объектам модели бизнес-процесса акторов модели информационной системы приведен на рис. 6.2.

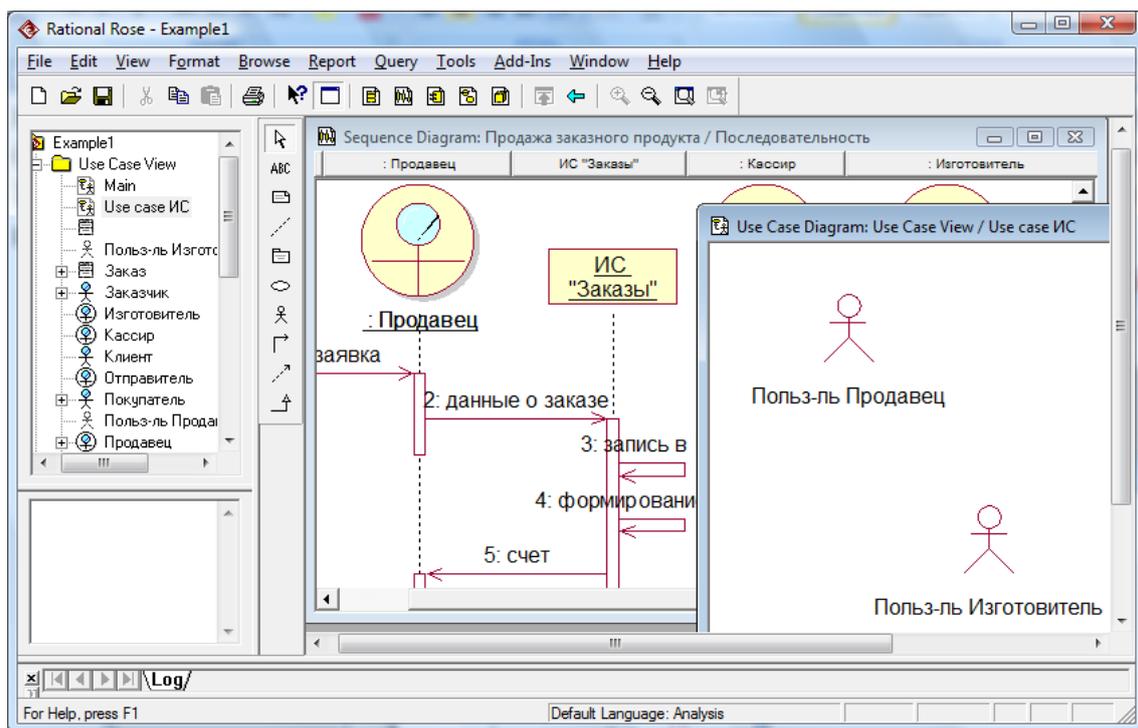


Рис. 6.2. Сопоставление объектам бизнес-процесса акторов ИС

В примере, представленном на рис. 6.2, *Продавец* и *Изготовитель* взаимодействуют с информационной системой, т.е. принимают или посылают сообщения объекту ИС «Заказы». Поэтому на диаграмму use case ИС помещены два актора - *Пользователь Продавец* и *Пользователь Изготовитель*.

Помимо участников бизнес-процесса на диаграмму use case ИС могут быть добавлены и другие акторы ИС, например, занимающиеся поддержкой и администрированием системы, в частности, пополнением базы данных, настройкой системы и т.д.

Следующим шагом является определение **функций информационной системы**. Вернитесь к диаграмме последовательности бизнес-процесса. Просмотрите всю «линию жизни» объекта информационной системы и выпишите все входящие и выходящие отношения сообщения (обязательства ИС по отношению к бизнесу). Занесите их в окно документации объекта ИС (см. рис. 6.3).

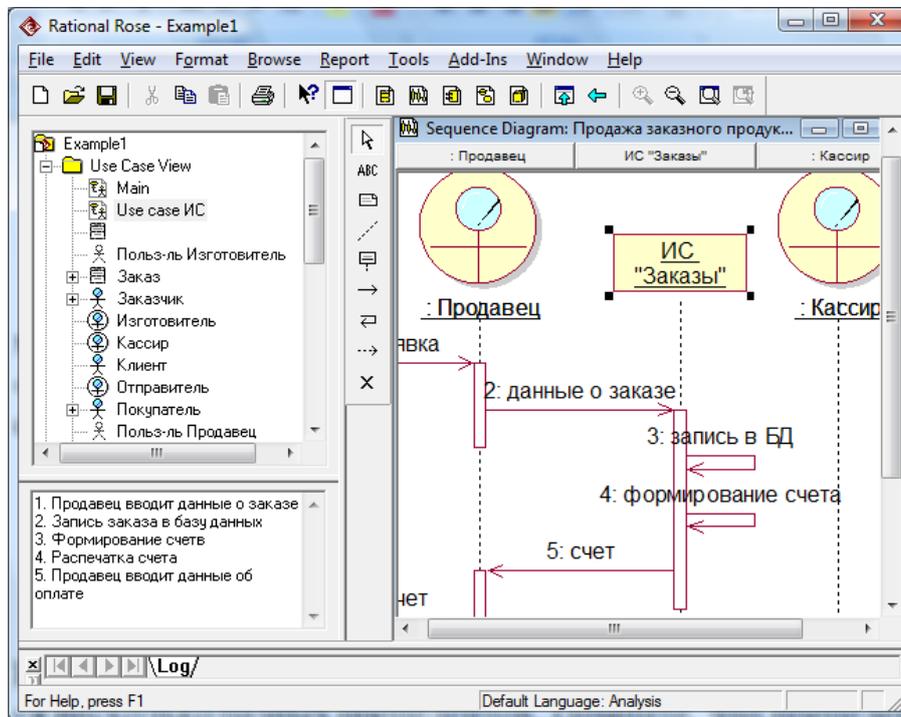


Рис. 6.3. Внесение обязательств объекта ИС в окно документации

После того, как выписаны все обязательства информационной системы, необходимо определить, какие **прецеденты ИС** могут реализовать эти обязательства. Подумайте, можно ли все функции выполнить с помощью одного прецедента или требуется несколько разных прецедентов. Помните, что прецедентом является относительно законченная последовательность действий ИС, заканчивающаяся предоставлением пользователю некоторого результата. Сгруппируйте все обязательства ИС в группы, соответствующие разным прецедентам (это можно сделать в окне документации). Возможно, Вам потребуется дополнить список обязательств ИС.

Перейдите на диаграмму use case ИС и поместите на нее требуемое количество прецедентов. Поименуйте прецеденты.

Вернитесь к диаграмме последовательности бизнес-процесса. Выберите объект ИС, в окне документации этого объекта выделите левой кнопкой мыши все обязательства ИС, относящиеся к одному из прецедентов, и скопируйте их в буфер. Перейдите на диаграмму use case ИС, выберите соответствующий прецедент и вставьте в окне документации обязательства из буфера (см. рис. 6.4).

По такой же схеме скопируйте обязательства информационной системы, относящиеся к другим прецедентам ИС, и внесите их в документацию прецедентов.

На диаграмме use case ИС соедините прецеденты с актерами ИС, участвующими в выполнении прецедентов (вводящими данные и/или получающими результаты), отношениями ассоциации, поименуйте эти связи (см. рис. 6.4).

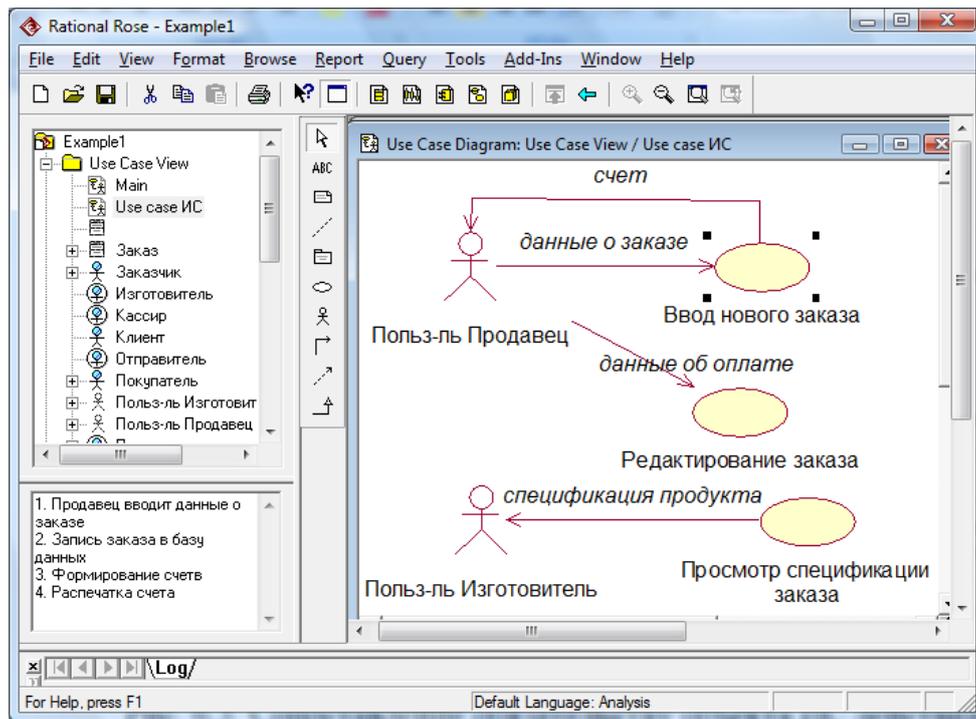


Рис. 6.4. Документирование прецедентов ИС,
установление связи между прецедентами и актерами ИС

На диаграмму use case ИС могут быть добавлены также прецеденты, напрямую не связанные с поддержкой бизнес-процесса. Например, прецеденты, выполняющие служебные функции, такие как настройка системы, формирование отчетов, предоставление справки, авторизация пользователей и т.д. Эти прецеденты могут выполняться как для конечных пользователей, являющихся участниками бизнес-процесса, так и для акторов, занимающихся администрированием системы.

Желательно структурировать выявленные прецеденты с помощью отношений обобщения, включения и расширения, например, выделить обобщенные прецеденты или выделить в отдельные прецеденты относительно законченные последовательности действий, или отдельно выделить последовательности действий, дополняющие основной ход событий при определенных условиях.

Процесс формирования диаграммы вариантов использования является творческим. Возможно, Вам потребуется несколько итераций прежде, чем Вы сформируете окончательный вариант.

4. Создание диаграмм деятельности для основных прецедентов ИС

Выберите на диаграмме use case ИС прецедент. Проанализируйте описание действий этого прецедента, внесенные на предыдущем этапе в окно документации. Добавьте в это описание новые действия, в частности, «внутренние» действия, не предусматривающие взаимодействие с пользователем. В результате должна получиться цепочка действий, в которой каждое действие «цепляет» следующее.

Создайте диаграмму деятельности (выбрав в контекстном меню Sub Diagrams → New Activity Diagram). Разделите поле диаграммы на две дорожки: одна должна соответствовать действиям пользователя, другая – действиям информационной системы. Пример диаграммы деятельности для прецедента ИС приведен на рис. 6.5.

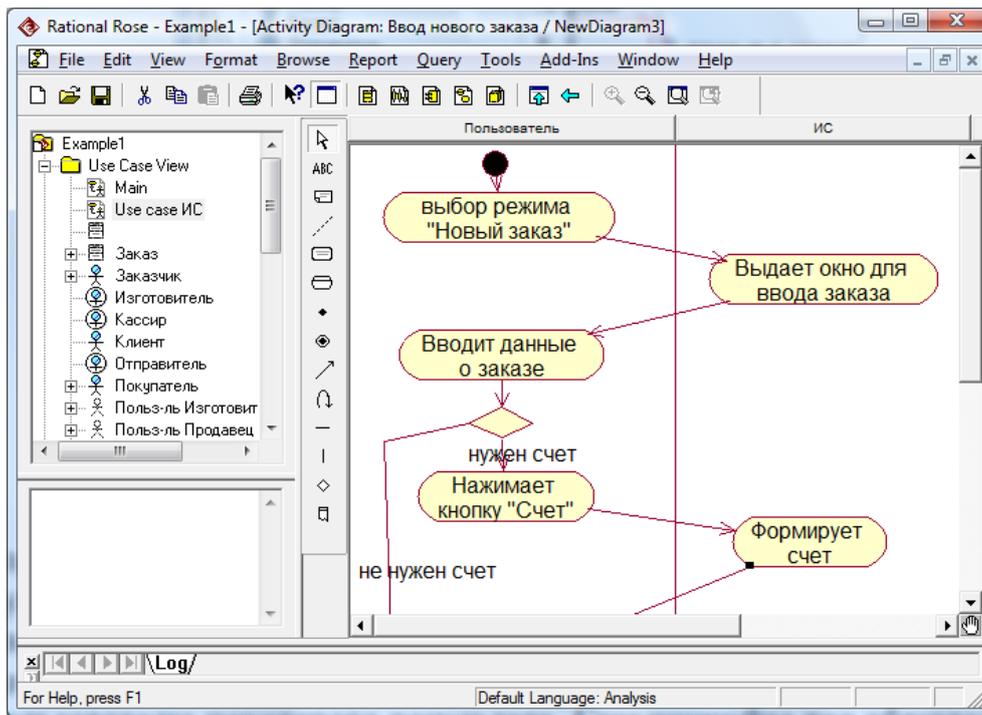


Рис. 6.5. Диаграмма деятельности прецедента ИС

5. Доработка модели

Завершите создание модели информационной системы (ИС) поддержки бизнес-процесса, выбранного вами в качестве индивидуального задания. Модель обязательно должна содержать диаграмму последовательности автоматизируемого бизнес-процесса, диаграмму прецедентов ИС, диаграмму деятельности для основных прецедентов ИС, а также, возможно (не обязательно) диаграмму классов ИС и диаграмму последовательности для прецедентов ИС.

Литература

1. Силич В.А., Силич М.П. Моделирование и анализ бизнес-процессов : учебное пособие. - Томск : Томск. гос. ун-т систем управления и радиоэлектроники, 2011. – 212 с.
2. Методология IDEF0. Стандарт. Русская версия. - М.: Метатехнология, 1993. – 107 с.
3. Калянов Г.Н. CASE-технологии: консалтинг в автоматизации бизнес-процессов. – М.: Горячая линия-Телеком, 2000. – 318 с.
4. Маклаков С.В. Моделирование бизнес-процессов с AllFusion Process Modeler (BPwin 4.1). – М.: ДИАЛОГ-МИФИ, 2003. – 240 с.
5. Федотова Д.Э., Семенов Ю.Д., Чижик К.Н. CASE-технологии: Практикум. - М.: Горячая линия-Телеком, 2005. – 160 с.
6. Леоненков А.В. Самоучитель UML. – СПб.: БХВ-Петербург, 2001. – 304 с.
7. Калашян А.Н., Калянов Г.Н. Структурные модели бизнеса: DFD-технологии; Под ред. Г.Н. Калянова. — М.: Финансы и статистика, 2003. – 256 с.
8. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М: Финансы и статистика, 1998. – 176 с.
9. Грекул В.И. Проектирование информационных систем. Курс лекций: Учебное пособие для вузов/ В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. - М.: Интернет-Университет Информационных Технологий, 2005. – 298 с.

...

Варианты индивидуального задания

Вариант 1. Продажа туристического продукта

Процесс продажи туристическим агентством турпродукта клиенту включает три основных этапа: прием заявки; подбор и предложение тура; оформление правоотношений и расчет с клиентом.

Первый этап начинается с приема клиента в офисе и определения его потребностей в выборе определенного типа отдыха. Клиенту предоставляется интересующая его информация и после того, как он определит свои потребности, оформляется заявка туриста на бронирование тура. Она содержит информацию о клиенте и сведения о намерении приобрести те или иные услуги (маршрут, начало и продолжительность тура, требования к проживанию и др.). Заявка подписывается клиентом и менеджером турагентства. Обе стороны поучают по экземпляру. Кроме того, клиент в подтверждение своих намерений оплачивает часть стоимости тура. Оплата производится в кассе агентства. Клиент получает квитанцию об оплате.

Второй этап предполагает бронирование тура, оформление визы, приобретение билетов на транспорт, страхование тура.

При бронировании тура менеджер турагентства заполняет предложенную туроператором форму заявки и отправляет ее на адрес туроператора (по факсу или по e-mail). Туроператор либо бронирует тур (при наличии свободных мест в отеле), либо отказывает в брони. В первом случае оформляется туристский ваучер – документ, гарантирующий туристу получение всех указанных в ваучере и оплаченных туристом услуг (проживание, питание, экскурсионное обслуживание и т.д.). Для принимающей стороны ваучер является гарантией получения от фирмы, направившей туриста, уплаченных им денежных средств.

В случае, если в туроператор отказал в брони, менеджер турагентства связывается с клиентом. Клиент корректирует свои требования и процедура бронирования повторяется.

Помимо брони, при необходимости, турагентство приобретает билеты на транспорт, а также страхует клиента. При выезде за рубеж, как правило, оформляется медицинский страховой полис, являющийся договором страхования между страхователем (туристом) и страховщиком (страховой компанией).

Для оформления визы клиент должен предоставить турагентству необходимые документы (загранпаспорт, фотографии, заполненную анкету и др.). Менеджер турагентства проверяет правильность заполнения анкеты, наличие всех необходимых документов и обращается в Консульство выбранной для турпоездки страны (или пересылает все документы по почте). Как правило, необходимо также предъявить подтверждение забронированного на весь срок поездки номера в отеле и уплатить консульский сбор. Оформление визы производится в срок от нескольких дней до нескольких месяцев, в зависимости от требований посольства той или иной страны.

На заключительном этапе осуществляется заключение договора на туристическое обслуживание, окончательный расчет с клиентом, а также выдача всех документов (туристической путевки, ваучера, билетов на транспорт, паспорта с визой, страхового полиса, памятки).

Договор на туристическое обслуживание является главным документом, на основании которого строятся взаимоотношения турагентства с клиентом. Туристская путевка формы ТУР-1 представляет собой бланк строгой отчетности, который используется при расчетах с туристом. Дата выписки (передачи) путевки и дата ее реализации отражаются на счетах бухгалтерского учета. При передаче путевки туристу отрывной талон остается у турфирмы в качестве документа, подтверждающего факт и дату реализации турпродукта.

Памятка туриста в обязательном порядке выдается туристу, выезжающему за рубеж. В ней описаны особенности посещаемой страны, возможные риски и опасности, запреты и рекомендации о том, как необходимо себя вести в этой стране, указывается вся необходимая информация о ней, которую туристу надо знать, выезжая в данную страну.

Вариант 2. Выпуск газеты

Процесс подготовки и выпуска номера печатного периодического издания делится на несколько этапов: подготовка информации, формирование номера и, наконец, его выпуск и распространение.

Подготовка журналистской информации, предназначенной для определенного номера газеты, начинается с отбора информации. Отбор производится в отделах редакции и в ее секретариате. Информацию может отбирать и ведущий редактор номера, отвечающий за его содержание. Отбор этот идет с использованием нескольких основных источников. Во-первых, это оперативная информация, доставляемая корреспондентами и собкорами редакции, а также информационными агентствами. Второй источник – запас готовых материалов, хранящихся в «портфелях» отделов и секретариата.

Отобранная информация должна быть подготовлена к публикации. Эта подготовка происходит, прежде всего, в процессе редактирования текстов. В редактировании каждого текста обычно участвуют: сам автор; его руководитель, приводящий текст в соответствие с планом отдела; сотрудник секретариата или ответственный секретарь, проверяющий правдивость и точность информации, содержащейся в тексте, и при необходимости сокращающий его размеры; ведущий редактор или сам главный редактор, оценивающий значение информации и определяющий ее место в номере; возможно, наконец, и корректор, сигнализирующий в секретариат и автору о смысловых и стилистических ошибках, замеченных в тексте.

Последний этап подготовки текста – техническое редактирование. Оно означает определение и указание объема текста (количества наборных строк), шрифта и формата набора, а также полосы, для которой предназначается текст.

Редактируют не только текст, но и оригиналы иллюстраций издания. Этим занимается бильд-редактор. Компьютеры и сканеры дают возможность кадрировать фотоснимки, уменьшать или увеличивать их размеры, удалять лишние детали и т.п.

После завершения подготовки всей журналистской информации формируется номер периодического издания в секретариате редакции. Осуществляется разделение всей массы материалов, отобранных для номера, на несколько частей. В каждой из них сосредоточиваются тексты и иллюстрации, объединяемые по какому-то общему признаку – теме, региону, времени описываемых событий. В каждой из таких групп публикаций производится дальнейшая их организация – выделение по значению, важности информации.

Переход к выпуску газетного номера совершается в процессе его макетирования. Макет номера – это его графический план, показывающий размещение текстов и иллюстраций на полосах, заголовков и авторских подписей, место и конфигурацию каждой из публикаций и т.д. Макетированием номера газеты руководят ответственный секретарь редакции и его заместители. При отсутствии у редакции электронной базы, позволяющей перейти к современному процессу выпуска газеты, подготовленный ими макет передается в типографию, где версткой номера руководит выпускающий – сотрудник секретариата. Процесс выпуска номера разделяется на этапы, соответствующие важнейшим производственным операциям – набору текстов, их верстки вместе с иллюстрациями в соответствии с макетами полос, их редакционной и корректорской вычитки, подписи каждой полосы главным редактором, подготовки печатных форм и других – вплоть до печатания тиража газеты.

Процесс подготовки и выпуска газеты завершается ее распространением. Отдел распространения передает тираж своим контрагентам, занимающимся распространением. В зависимости от используемых методов распространения (подписка, розничная продажа, бесплатная доставка) это могут быть отделения Роспечати, фирмы, имеющие сети лотков или киосков, частные распространители периодики и др.

Вариант 3. Кредитование владельцев частных предприятий

Процесс предоставления кредита начинается с подачи заявления. Клиент обращается в кредитную организацию в отдел регистрации с заявлением о выдаче ему кредита. После регистрации клиент с зарегистрированным заявлением обращается к специалисту отдела кредитования. Специалист отдела кредитования доводит до сведения клиента условия кредитования. В случае, если условия устраивают клиента, то специалист отдела кредитования составляет проект договора.

Далее клиент предает проект договора специалисту отдела экономической безопасности, который проверяет платежеспособность клиента и инвестиционные риски. Если клиент является неплатежеспособным, то ему сообщается об этом, и процесс после отметки этого факта в книге регистрации (в отделе регистрации) прекращается. Если клиент является платежеспособным, но существуют повышенные инвестиционные риски, то об этом сообщается клиенту, и в этом случае либо процесс прекращается, либо условия договора корректируются, после чего новый проект договора передается специалисту отдела кредитования.

После этого проект договора передается юристу для проверки юридической чистоты сделки. После проверки проект договора визируется у начальника юридического отдела и начальника отдела кредитования. Затем завизированный проект договора передается в отдел перевода.

В случае, если у клиента отсутствует открытый счет в банке, с которым сотрудничает кредитная организация, то он открывает счет. После открытия счета клиент сообщает номер счета в отдел перевода.

Затем договор визируется директором кредитной организации и регистрируется в отделе документооборота и в книге регистрации отдела регистрации. Один из экземпляров договора передается клиенту и на его счет переводятся запрашиваемые денежные средства.

Вариант 4. Предоставление доступа к местной телефонной сети

Бизнес-процесс начинается с подачи клиентом заявления на установку телефона в абонентский отдел Управления электросвязи. Сотрудник абонентского отдела регистрирует заявление и передает его в технический отдел для анализа технической возможности установки телефона.

Технический отдел, изучив все необходимые данные о задействованной линейной емкости телефонной сети, возвращает заявление в абонентский отдел либо с отрицательным решением (если нет технической возможности установить телефон), либо с положительным заключением и указанием бронируемых линейных данных (если возможность имеется). Абонентский отдел в случае отрицательного решения об установке телефона ставит клиента на очередь (записывает данные клиента в базу данных очередников) и извещает клиента о постановке его на очередь. В случае же положительного решения абонентский отдел уведомляет бухгалтерию о необходимости оформить счет новому абоненту на оплату установки телефона. Бухгалтерия информирует клиента о необходимости оплатить установку телефона, оформляет счет и получает плату от клиента, после чего информирует абонентский отдел об оплате.

Затем абонентский отдел выписывает наряд на установку телефона и согласует его с начальником цеха городской телефонной станции (ГТС), после чего передает наряд в отдел КРОСС цеха ГТС для выделения номера новому абоненту и далее в линейно-кабельный участок, где выполняется подключение абонента к линии и установка ему телефона.

После этого наряд с отметкой о подключении возвращается в технический отдел, где номер абонента заносится в базу данных по линейной емкости, и в абонентский отдел. Абонентский отдел заключает договор с абонентом об оказании услуг телефонной связи и информирует отдел по расчетам с населением о номере нового абонента и договоре с ним для внесения этих данных в базу данных по абонентам. На данном этапе абонент получает доступ к городской телефонной сети.

Вариант 5. Дипломирование студентов вузов

Период дипломирования является завершающим этапом выполнения дипломной работы. Он начинается сразу после окончания преддипломной практики и заканчивается защитой. Длительность периода дипломирования в соответствии с учебным планом составляет 16 недель.

На первом этапе осуществляется подготовка проекта приказа и утверждение приказа. Каждый студент-дипломник совместно со своим руководителем (назначенным или выбранным самим студентом) определяет тему дипломной работы. Сведения о своем руководителе диплома и тему работы студент подает руководителю дипломирования. Руководитель дипломирования готовит проект приказа. После обсуждения проекта с заведующим кафедрой и, возможно, корректировке тем, готовится окончательный вариант приказа, который визируется заведующим кафедрой, деканом и подписывается проректором университета по учебной работе.

Непосредственно выполнение дипломной работы включает в себя: выдачу руководителем диплома задания на дипломирование (при этом оформляется лист задания, подписываемый студентом, руководителем и заведующим кафедрой); разработку и написание студентом разделов работы; распечатку работы. После написания каждого раздела руководитель диплома проверяет написанное, делает замечания, и студент корректирует или дополняет содержание раздела.

Этап подготовки к защите начинается примерно за шесть недель до защиты. Руководитель дипломирования составляет график подготовки к защите, составляет список рецензентов, согласует его с заведующим кафедрой и проводит собрание дипломников, на котором доводит до сведения студентов всю информацию о порядке подготовки к защите.

Подготовка каждой дипломной работы включает в себя: проверку ее руководителем диплома, прохождение нормоконтроля, рецензирование, сдачу всех документов секретарю государственной аттестационной комиссии (ГАК), подготовку доклада, демонстрационных и раздаточных материалов.

Проверка работы руководителем диплома. Руководитель читает работу, ставит свою подпись на обложке, делает в зачетной книжке запись о допуске студента к защите и пишет отзыв руководителя на дипломную работу, в которой дает оценку выполненной работе. Если руководитель диплома не является сотрудником университета, то все его подписи (на обложке работы, в зачетной книжке и в отзыве) заверяются печатью предприятия, на котором он работает.

Прохождение нормоконтроля. Студент сдает работу нормоконтролеру кафедры, который проверяет правильность оформления работы и делает замечания. После исправления всех замечаний студент «сшивает» работу и приносит ее нормоконтролеру для окончательной проверки. Если замечаний нет, но нормоконтролер подписывает работу. Прием студентов нормоконтролер осуществляет в соответствии с графиком подготовки к защите.

Рецензирование. Студент получает направление на рецензию у руководителя дипломирования. Рецензенту передаются дипломная работа, памятка по составлению рецензии, бланк для оплаты. Памятку и бланк оплаты студент получает у секретаря ГАК. Рецензент пишет рецензию, в которой дается оценка работе, подписывает ее, заверяет печатью и передает студенту вместе с дипломной работой и заполненным бланком оплаты.

Сдача документов. Студент представляет секретарю ГАК следующие документы: дипломную работу; зачетную книжку с отметкой деканата и допуском руководителя; отзыв руководителя; рецензию; заполненные рецензентом и руководителем бланки на оплату. Секретарь ГАК дает дипломные работы на подпись заведующему кафедрой и составляет график защиты.

Подготовка доклада, демонстрационных и раздаточных материалов. Студент обсуждает с руководителем диплома план выступления на защите, готовит доклад на семь минут, компьютерную презентацию и раздаточные материалы. Презентацию необходимо передать секретарю ГАК или ответственному за компьютерные презентации.

Вариант 6. Аттестация муниципальных служащих

Периодическая аттестация муниципальных служащих проводится в соответствии с федеральными законами и нормативными актами органов местного самоуправления для определения уровня профессиональной подготовки служащих и соответствия их занимаемым должностям. Процесс аттестации предусматривает три основных этапа – подготовительный, основной и заключительный.

Подготовительный этап начинается с разработки методики проведения аттестации, выполняемой отделом социально-трудовых отношений (СТО) при участии мэра. Затем выполняется подготовка и издание нормативных документов: о формировании аттестационной комиссии; об утверждении графика проведения аттестации; об утверждении списков муниципальных служащих, подлежащих аттестации. Документы готовятся секретариатом мэра и утверждаются мэром.

Отдел СТО занимается подготовкой материалов для аттестации (бланков, форм, анкет), а также обучением и консультированием всех, вовлеченных в аттестационный процесс – членов аттестационной комиссии, руководителей структурных подразделений, в которых проводится аттестация, самих аттестуемых.

На каждого аттестуемого служащего, формируются следующие документы:

1. Анкета аттестуемого служащего
2. Отзыв руководителя.
3. Сведения о служебной деятельности.
4. Характеристика муниципального служащего
5. Психологическое резюме и/или экспертная оценка.

Первый документ (анкета) представляет сам аттестуемый, следующие два документа (отзыв и сведения) представляет непосредственный руководитель аттестуемого, остальные – отдел социально-трудовых отношений. В ходе формирования документов проводятся тестирования, экспертизы, анализ кадровых данных и другие мероприятия с участием экспертов, психологов, работников кадровых служб. Все документы (кроме анкеты) подаются в аттестационную комиссию не позднее, чем за 2 недели до начала аттестации. Анкета подается в день проведения аттестации. Каждый аттестуемый служащий должен быть заранее (не менее, чем за неделю до начала аттестации) ознакомлен с представленными документами.

Основной этап заключается собственно в проведении аттестации в соответствии с графиком аттестации. На заседание комиссии приглашаются аттестуемые и их непосредственные руководители. Члены комиссии заслушивают сообщения аттестуемого и его руководителя, задают вопросы, рассматривают представленные документы. Затем проводится обсуждение, в ходе которого высказываются мнения, замечания и предложения, касающиеся профессиональных и личностных качеств служащего, эффективности его деятельности, направлений развития и т.д. С учетом обсуждений и в отсутствие аттестуемого комиссия открытым голосованием простым большинством принимает одно из следующих решений: аттестуемый соответствует занимаемой должности; соответствует при условии выполнения рекомендаций; не соответствует. Решение и рекомендации комиссии заносятся в аттестационный лист и сообщаются аттестуемому непосредственно после голосования. Лист и протокол заседания подписывается председателем и членами комиссии.

На заключительном этапе принимаются кадровые решения по итогам аттестации. Материалы аттестации муниципальных служащих и не позднее чем через семь дней после ее проведения подаются мэру, а также руководителям департаментов и служб, в которых работают прошедшие аттестацию служащие. В течение месяца материалы рассматриваются, изучаются, сравниваются с материалами предыдущей аттестации и принимаются решения (о переводе на другую должность, о включении в кадровый резерв, изменении условий оплаты труда, направлении на переподготовку или повышение квалификации и т.д.). Принятые решения оформляются документально (издаются соответствующие приказы и распоряжения) и доводятся до служащего.

Вариант 7. Организация выставки-ярмарки

Процессы организации выставок-ярмарок осуществляются деловым центром «Технопарк» в соответствии с годовым планом выставочно-ярмарочных мероприятий. За каждой выставкой закрепляется директор выставки.

Начало работ по подготовке выставки начинается за 6 – 7 месяцев до начала ее проведения. Основанием является распоряжение о проведении выставки. Согласно распоряжению утверждается состав организационного комитета выставки. Подготовительный этап начинается с разработки оргкомитетом концепции, описывающей цели, задачи и направления (разделы) выставки. В соответствии с концепцией разрабатывается деловая и научная программа выставки (проведение семинаров, конференций, крупных презентаций, круглых столов).

Затем формируется клиентская база потенциальных участников (экспонентов, участников научной и деловой программы) с использованием информационной системы. Исходной информацией являются данные о предприятиях, представленных на рынке, соответствующем тематике выставки (данные могут храниться в базах данных). Директор выставки составляет информационные письма - приглашения на участие в выставке. Информационный отдел рассылает письма потенциальным участникам выставки, осуществляет прием поступивших заявок на участие в выставке, их учет и ведение списка участников. Заявка – Договор на участие в выставке-ярмарке – содержит информацию об услугах, предоставляемых Технопарком, об условиях обслуживания и стоимости услуг. При оформлении заявки бухгалтерией выписывается и отправляется счет на оплату участия в выставке. После оплаты счета в списке участников ставится отметка об оплате.

С учетом поступивших заявок на участие в выставке, а также на основе концепции, деловой и научной программы оргкомитет составляет сводную программу мероприятий выставки-ярмарки и план экспозиции. Кроме того, он разрабатывает положение о конкурсе и формирует конкурсную комиссию. Конкурс проводится среди участников выставки-ярмарки в ходе ее проведения.

На заключительной стадии подготовительного этапа административно-хозяйственный отдел осуществляет подготовку выставочного павильона (оформление выставочного зала, информационного стенда и др.), размещение участников, подготовку мест проживания и питания иногородних участников, обеспечение транспортом.

Параллельно отдел рекламы изготавливает рекламно-информационную продукцию (каталог выставки, листовки, афиши, пригласительные билеты, значки и т.д.), совместно с информационным отделом осуществляет расклейку афиш, распространение листовок, рассылку приглашений, размещение рекламы в СМИ.

Этап проведения выставки включает: сопровождение выставки; проведение деловой и научной программы; проведение конкурсов. Сопровождение выставки выполняется, в основном, силами административно-хозяйственного отдела и включает организацию работы гардероба, буфета, Информбюро, радиорубки, обеспечение транспортом. Мероприятия деловой и научной программы (семинары, конференции, круглые столы) проводятся оргкомитетом выставки или сторонней организацией. Проведение конкурсов осуществляется конкурсной комиссией и включает в себя: сбор информации об экспонатах, принимающих участие в конкурсе; проведение заседаний комиссии, на которых производится оценка экспонатов, выбор победителей конкурса и распределение мест между победителями; вручение медалей, дипломов и удостоверений лауреатам конкурса.

На заключительном этапе осуществляется подведение итогов выставки-ярмарки. Оргкомитет выставки подготавливает материалы – список участников выставки, список представленных товаров и услуг, информацию о заключенных в ходе проведения выставки контрактах между участниками и посетителями (как, в количественном, так и в денежном выражении), список лауреатов конкурса и т.д. Информация анализируется директором выставки и членами оргкомитета, формулируются выводы и рекомендации и определяется содержание отчета. Написание отчета и его обсуждение на заседании совета Технопарка завершает процесс организации выставки-ярмарки.

Вариант 8. Ремонт квартиры

Клиент обращается в фирму, специализирующуюся на ремонте квартир. В отделе приема заявок консультант знакомит клиента со всей интересующей его информацией (виды работ, выполняемые фирмой, расценки, сроки выполнения и т.д.). Если клиента решает воспользоваться услугами фирмы, то он подает заявку и договаривается о дате и времени осмотра квартиры. В назначенное время мастер и дизайнер делают осмотр квартиры и обговаривают с клиентом требования к ремонту. Если требования клиента выполнимы, то клиент заключает договор на выполнение работ. Кроме того, клиент вносит задаток, т.е. делает предоплату в кассе и получает квитанцию об оплате.

Дизайнер выполняет дизайн-проект и согласовывает его с клиентом и мастером. При необходимости, в проект могут быть внесены изменения. Если проект устраивает все стороны, то мастер определяет перечень необходимых материалов и составляет план работ, которые также согласовываются с клиентом. Затем мастер передает перечень материалов в отдел закупок, который выписывает счет на приобретение материалов и сообщает клиенту о необходимости оплатить счет. После оплаты счета отдел закупок выписывает необходимые материалы со склада и/или приобретает их в строительных магазинах (возможно и поэтапное приобретение материалов). Материалы доставляются на объект ремонта.

Бригада строительных рабочих осуществляет подготовительные работы (выносит мебель, удаляет со стен обои, при необходимости снимает паркет, снимает старые окна, двери и т.д.). Затем выполняются ремонтные работы в соответствии с планом работ, например, покраска потолка, наклейка обоев, настиление полов и т.д. По окончании каждого этапа мастер (а также, возможно, клиент и дизайнер) проверяет качество выполненных работ, и, при наличии замечаний, все выявленные дефекты исправляются. На заключительном этапе выполняется окончательная приемка работы клиентом. Мастер определяет объем выполненных работ и выписывает счет с учетом задатка. Клиент оплачивает счет.

Вариант 9. Изготовление шкафа-купе на заказ

Процесс изготовления встроенного шкафа-купе начинается со звонка клиента оператору компании. Клиент сообщает параметры шкафа, в том числе размер комнаты и высоту потолков. Менеджер производит предварительный расчет стоимости проекта шкафа, основываясь на цене базовой модели шкафа-купе (выполненной из стандартных материалов и имеющей стандартную, усредненную начинку). После этого менеджер связывается с клиентом и записывает дату и время проведения замера — для уточнения заказа.

В указанное время в квартиру к клиенту приходит мастер-консультант. Он производит точный замер места установки шкафа, помогает клиенту выбрать конструкцию шкафа (наличие пола, потолка, задней стенки, антресолей), количество дверей и материал, из которого они будут выполнены, цветовое решение, а также начинку шкафа. При этом клиенту предоставляются каталоги с цветными иллюстрациями и ценами. После того, как клиент сделал выбор, составляется эскиз шкафа-купе с указанием точных размеров и расположения всех элементов и составляется спецификация на элементы шкафа-купе. На основании спецификации и прайс-листа осуществляется точный расчет стоимости. Затем мастер-консультант оформляет счет. Клиент производит оплату и получает чек.

Мастер-консультант передает оплаченный счет и деньги в бухгалтерию, а эскиз и спецификацию — мастеру производственного цеха. На основании спецификации в производственном цехе на специальном оборудовании «раскраивают» детали шкафа, «изолируют» срезы ламината, нарезают металлические профили каркаса шкафа, а также его декоративные элементы. По окончании работ мастер сообщает оператору о готовности к сборке шкафа, и оператор уточняет у клиента время доставки.

Последний этап — сборка шкафа-купе на месте. Все необходимые детали упаковываются, загружаются в грузовой автомобиль, доставляются и разгружаются рабочими-сборщиками. Затем ими осуществляется установка и сборка шкафа-купе на месте. Бригадир и клиент подписывают акт приемки-сдачи работ. В заключение оформляется гарантия.

Вариант 10. Страхование квартиры и домашнего имущества

Процесс начинается с поиска клиента страховым агентом. Способы могут быть разные – обход квартир, опрос постоянных клиентов, звонки по телефону и т.д. Потенциального клиента необходимо заинтересовать. Страховой агент рассказывает, какие в представляемой им компании существуют варианты страхования (какие риски страхуются, каковы условия страхования). Если клиент заинтересовался, то страховой агент знакомит его с порядком страхования, в частности, какие документы необходимо подготовить, и назначается дата обследования страхуемой квартиры и имущества.

Страховой агент приходит в квартиру. Уточняет у клиента, каков объект страхования (квартира, имущество), какой вариант выбирает клиент (базовый, полный или расширенный) и какова желаемая предельная сумма страхования.

Если страхуется имущество, то агент составляет перечень страхуемого имущества и ведет фотосъемку. Если страхуется квартира, то агент составляет описание квартиры (отделки), в котором приводится: общая характеристика здания (тип дома, год постройки, количество этажей и др.); общая характеристика квартиры (этаж, площадь, принадлежность и др.); детальная характеристика квартиры (отделка стен, потолка, половое покрытие). Составляется также описание инженерного оборудования (санитарно-технических приборов, отопительного оборудования, и др.). Ведется фотосъемка.

Страховой агент передает составленные описания имущества и/или квартиры, а также фотографии эксперту страховой компании. Эксперт осуществляет калькуляцию, исходя из желаемой суммы страхования. При этом эксперт использует базы данных для определения стоимости объекта страхования, страховой суммы (суммы страхового возмещения) и страховой премии (платы за страхование). Если вычисленная сумма объекта страхования превышает предельную сумму, назначенную клиентом, то страховой агент связывается с клиентом и согласует с ним требуемые изменения.