

Министерство образования и науки РФ

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

Кафедра автоматизации обработки информации (АОИ)

УТВЕРЖДАЮ
Зав. каф. АОИ, профессор

_____ Ю.П. Ехлаков
" ____ " _____ 2011 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

**для выполнения практических и самостоятельных работ
по дисциплине**

«МОДЕЛИРОВАНИЕ И АНАЛИЗ БИЗНЕС-ПРОЦЕССОВ»

для студентов, обучающихся по направлению 080700 –
"Бизнес-информатика" (бакалавр)

Разработчик:
профессор каф. АОИ, д.т.н.
_____ М.П. Силич

Томск 2011

СОДЕРЖАНИЕ

Введение	3
Практическая работа №1 Знакомство с основами методологии IDEF0... Создание диаграмм корневого и первого уровня	4
1.1. Выполнение работы на примере «Создание продукта»	4
1.2. Самостоятельная работа	15
1.3. Выполнение индивидуального задания	16
Практическая работа №2 Создание диаграмм декомпозиции	17
2.1. Выполнение работы на примере «Создание продукта»	17
2.2. Самостоятельная работа	21
2.3. Выполнение индивидуального задания	21
Практическая работа №3 Создание словаря данных для IDEF-модели ..	22
3.1. Выполнение работы на примере «Создание продукта»	22
3.2. Самостоятельная работа	30
3.3. Выполнение индивидуального задания	30
Практическая работа №4 Дополнительные возможности пакета Design/IDEF ...	31
4.1. Выполнение работы на примере «Создание продукта»	31
4.2. Самостоятельная работа	38
4.3. Выполнение индивидуального задания	38
Практическая работа №5 Знакомство с UML и Rational Rose. Создание диаграммы вариантов использования	39
5.1. Выполнение работы на примере «Продажа продукта»	39
5.2. Самостоятельная работа	49
5.3. Выполнение индивидуального задания	49
Практическая работа №6 Создание диаграммы деятельности	50
6.1. Выполнение работы на примере «Продажа продукта»	50
6.2. Самостоятельная работа	54
6.3. Выполнение индивидуального задания	54
Практическая работа №7 Создание диаграмм взаимодействия объектов	55
7.1. Выполнение работы на примере «Продажа продукта»	55
7.2. Самостоятельная работа	62
7.3. Выполнение индивидуального задания	62
Практическая работа №8 Создание диаграммы классов	63
8.1. Выполнение работы на примере «Продажа продукта»	63
8.2. Самостоятельная работа	68
8.3. Выполнение индивидуального задания	68
Индивидуальное творческое задание	69
Список рекомендуемой литературы	70
Приложение 1. Варианты индивидуальных заданий	71

ВВЕДЕНИЕ

Практические и самостоятельные работы по дисциплине имеют целью получение навыков самостоятельного моделирования и анализа бизнес-процессов с помощью современных инструментальных средств.

Практические работы выполняются в соответствии с методическими рекомендациями. Выполнение каждой работы включает в себя три этапа:

1. Изучение методического материала по теме практического занятия и выполнение заданий, описанных в методических рекомендациях.
2. Самостоятельная работа по подготовке к выполнению индивидуального задания.
3. Выполнение индивидуального задания и защита работы.

Первый и третий этап выполняются в компьютерном классе во время, отведенное по расписанию для проведения практических занятий, второй этап (самостоятельная работа) выполняется в свободное время между аудиторными практическими занятиями.

Самостоятельная работа состоит в сборе информации о конкретном бизнес-процессе, выбранном студентом, и составлении текстового описания. Список бизнес-процессов, из которого студент выбирает процесс для выполнения индивидуального задания, приведен в приложении 1. Сбор информации о бизнес-процессе осуществляется на основе изучения литературных источников (книг, статей в журналах, в сборниках), публикаций в Интернете, законодательных, нормативно-правовых актов, знаний и опыта коллег и знакомых. Источники информации выбираются студентом самостоятельно.

Защита практической работы выполняется путем демонстрации созданной модели и собеседования.

Самостоятельная работа включает в себя помимо подготовки к практическим занятиям и выполнение творческого индивидуального задания. Творческое задание состоит в применении некоторой методологии для моделирования конкретного бизнес-процесса.

ПРАКТИЧЕСКАЯ РАБОТА №1

ЗНАКОМСТВО С ОСНОВАМИ МЕТОДОЛОГИИ IDEF0. СОЗДАНИЕ ДИАГРАММ КОРНЕВОГО И ПЕРВОГО УРОВНЯ

1.1. Выполнение работы на примере «Создание продукта»

Цель работы: Ознакомиться с основами методологии IDEF0 и основами работы с пакетом Design/IDEF. Создать диаграммы корневого и первого уровня на примере модели «Создание продукта», используя специальные средства и команды Design/IDEF (создание IDEF-блоков и дуг, создание текста и т.д.).

Порядок выполнения работы

1. Знакомство с основами методологии IDEF0 и с пакетом Design/IDEF

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 проектное 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. В соответствии с проектом ICAM было разработано семейство методологий IDEF, в которое кроме методологии создания функциональной модели сложной системы IDEF0 вошли еще две самостоятельные методологии моделирования:

- IDEF1 – методология создания информационной модели производственной среды или системы (основана на реляционной теории Кодда и использование ER – диаграмм Чена);
- IDEF2 – методология создания динамической модели производственной среды или системы.

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 блока. Детали (компоненты) каждого из основных блоков показаны на других диаграммах в виде блоков. Далее они могут быть превращены в более подробные диаграммы, до тех пор, пока не будет достигнута требуемая степень детализации.

Блоки представляют собой функции (действия, процессы или операции), а входящие и исходящие из них стрелки представляют объекты (информацию, предметы). Блок и его стрелки на диаграмме рассматриваемого уровня описывается более подробно блокам и стрелкам диаграмм нижнего уровня.

Блоки-функции показывают, что должно выполняться, причем без идентификации каких-либо других аспектов, например, таких как потребности в них или средства их осуществления. Наименование функций записывается внутри блока. Оно должно содержать существительное, обозначающее действие. Блоки нумеруются в нижнем углу.

Стрелки на диаграмме играют роль интерфейсов (связей) блоков с внешней для них средой. Каждая из стрелок имеет метку, характеризующую ее. Назначение стрелок зависит от стороны блока, в которую стрелка входит или выходит (рис. 1.1):

- Входящие стрелки слева от блока представляет собой предметы (материальные объекты) или информацию (информационные объекты), необходимые для выполнения функции. Это сырье, материалы, исходные данные или "вход" функции (стрелка типа I - input).

- Выходящие стрелки справа из блока показывают предметы или данные, полученные в результате выполнения функции блока. Это результат (стрелки типа O - output). Функции преобразуют объекты слева направо (от входа к выходу). Таким образом, блок представляет собой переход от состояния "до" к состоянию "после".

- Входящие стрелки сверху блока является управлением, описывающим условия или данные, которые управляют выполнением функции (стрелки типа C - control). Как и входы, они тоже могут являться

информацией, но назначение входа и управления различны. Их разделение является важным для понимания работы системы.

- Входящая снизу стрелка представляет собой «механизм», обозначающая собой либо человека, либо некоторое средство, выполняющее функцию (стрелка М – mechanism).

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.

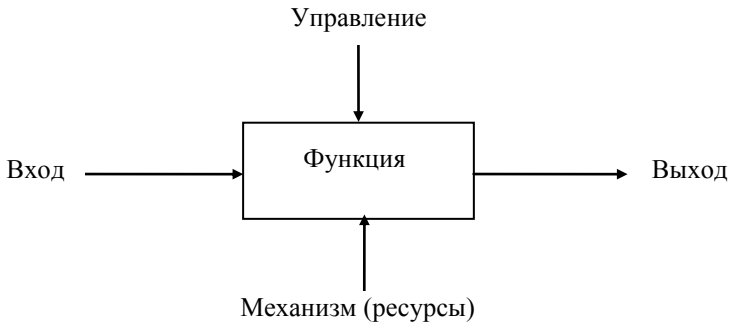


Рис. 1.1. Функциональный блок и входящие/выходящие стрелки

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. Обратная связь, итерация, непрерывные процессы, перекрытие (во времени) функций легко отображаются стрелками.

Стрелки могут разветвляться или соединяться. Каждая из ветвей может представлять один и тот же объект или различные объекты одного и того же типа.

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.

Примеры типичных наименований блоков и стрелок:

Функции

планирование	разработка	классификация
управление	проектирование	измерение
контроль	изготовление	редактирование
расчет	производство	учет
продажа	реализация	сбор

Управление

инструкции	требования	чертеж
стандарты	запросы	правило
указания	заявки	
задания	план	

Механизм

персонал	компьютер	информационная система
фирма	станок	программное обеспечение
заказчик	аппаратура	подрядчик
инструмент	оператор	оснастка

Методологии IDEF послужили основой для создания нескольких CASE-продуктов. Один из них Design/IDEF, разработанный фирмой Meta Software (США). Пакет Design/IDEF – графическая среда для проектирования и моделирования сложных систем широкого назначения. Основные возможности данного пакета:

- организация графической среды; позволяющей создавать стандартные и пользовательские графические объекты, осуществлять выравнивание и манипулирование объектами, и т.д.;
- обеспечение непротиворечивости модели;
- поддержка встроенного “Словаря данных”, позволяющего хранить информацию и о функциях и потоках данных в IDEF-модели;
- генерация отчетов;
- организация коллективной работы;
- моделирование данных (IDEF1-, IDEF1X- и ER-методологии).

2. Начало работы с пакетом Design/IDEF

Запустите Design/IDEF. После запуска программы открывается главное окно, содержащее: меню в верхней части окна, стандартную панель инструментов (под строкой меню) и специальную панель инструментов в левой части окна (рис. 1.2).

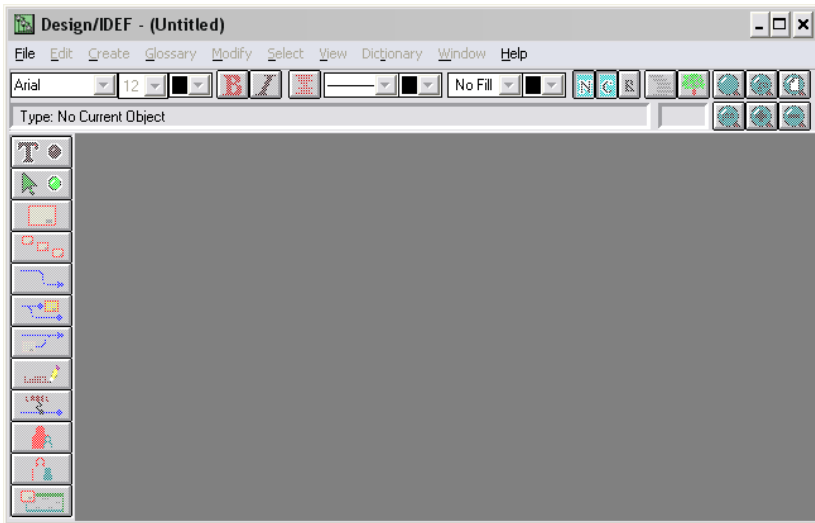


Рис. 1.2. Вид главного окна

Выберите в меню команду File/New, в открывшемся диалоговом окне щелкните на кнопке ОК для подтверждения того, что Вы будете использовать методологию IDEF0 и стандартную форму (мастерскую страницу), содержащуюся в файле startup\startup.msp. Информация с этой страницы будет воспроизводиться на каждом уровне с соответствующим изменением атрибутов, таких как название, узел, дата и др.


Появится окно диаграммы с именем A-0: P.1. Переключать окна можно через меню Window.

Сохраните файл под именем, которое Вы присвоите своей модели. Для этого выберите в меню команду File/ Save и в открывшемся диалоговом окне введите имя файла (расширение idd автоматически добавится к имени файла).

Щелкните кнопкой мыши на кнопке ОК.


3. Создание контекстной диаграммы

В Design/IDEF термин страница - синоним термина диаграмма. При создании новой модели автоматически создаётся диаграмма верхнего уровня (контекстная диаграмма) A-0, содержащая единственный блок A0.

Для ввода текста в блок A0 необходимо включить текстовый режим: выберите в меню Modify/Turn On Text или щёлкните по кнопке  на специальной панели инструментов. Щёлкните внутри блока A0 и напечатайте «Создание продукта».


Изменить шрифт, размер, начертание, цвет текста можно с помощью соответствующих инструментов стандартной панели инструментов, расположенной под меню или в диалоговом окне атрибутов, вызываемом командой меню Edit/Set Attributes ...

Отмените текстовый режим: выберите в меню Modify/Turn Off Text либо повторно щёлкните по кнопке с буквой “Т” на инструментальной панели, либо нажмите клавишу Esc.

Можно вместо этого включить режим перемещения, щёлкнув по кнопке .

Блок A0 будет выделен черными маркерами. Его можно перемещать с помощью мыши, изменять размеры (потянув за маркеры), удалить (с помощью клавиши Delete).

Стандартная контекстная IDEF0-диаграмма включает формулировки цели и точки зрения модели.

Введите формулировки как метки. Для этого выберите в меню Create/Label или щёлкните по кнопке  на специальной панели инструментов. Указатель примет форму указателя метки.

Щёлкните левой кнопкой мыши ниже блока A0.

Напечатайте: «Цель: Организовать процесс создания продукта на заказ». Нажмите Enter и напишите: «Точка зрения: Группа разработки».

Нажмите клавишу Esc, чтобы закончить создание метки или включите режим перемещения. Метка будет выделена черными маркерами. Передвиньте метку с помощью мыши в нижнюю часть страницы.

Вы в любой момент можете изменить текст метки. Для этого нужно выделить метку, щёлкнув на ней мышью, включить текстовый режим и отредактировать текст.

Сохраните Вашу модель с помощью команды меню File/ Save.

4. Создание внешних IDEF-дуг

Дуги могут быть созданы только между блоком и другим блоком или между блоком и меткой. Создайте и разместите шесть меток как показано на рисунке 1.3. Каждая метка создается так же, как создавалась метка с формулировкой цели и точки зрения.

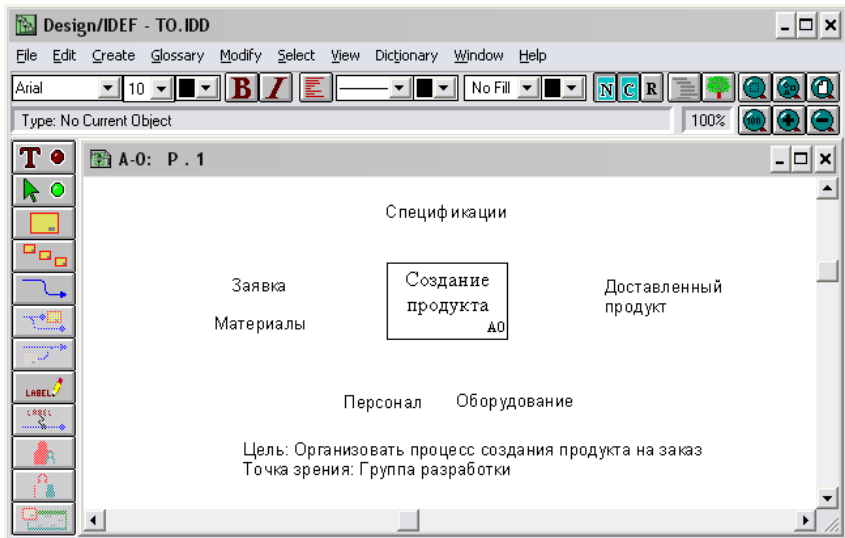



Рис. 1.3. Размещение меток

Чтобы нарисовать дугу выберите команду меню Create/Arrow или щелкните по кнопке  на панели инструментов. Указатель примет форму стрелки. Поместите указатель внутрь метки «Заявка» около ее правой границы.

Нажмите кнопку мыши и, не отпуская ее, переместите указатель до левой стороны блока A0. Отпустите кнопку мыши, чтобы закончить создание дуги.

Указатель активен, пока не отмените его нажатием клавиши Esc или не включите режим перемещения.

Аналогичным образом создайте дуги от других меток. При создании дуг механизма (от меток «Персонал» и «Оборудование») перемещайте указатель от метки к нижней стороне блока. При создании дуги управления (от метки «Спецификации») перемещайте указатель

от метки к верхней стороне блока. При создании дуги выхода (к метке «Доставленный продукт») перемещайте указатель от правой стороны блока к метке.

Отмените режим рисования дуг. Нечто, похожее на рисунок 1.4, должно получиться и у Вас.

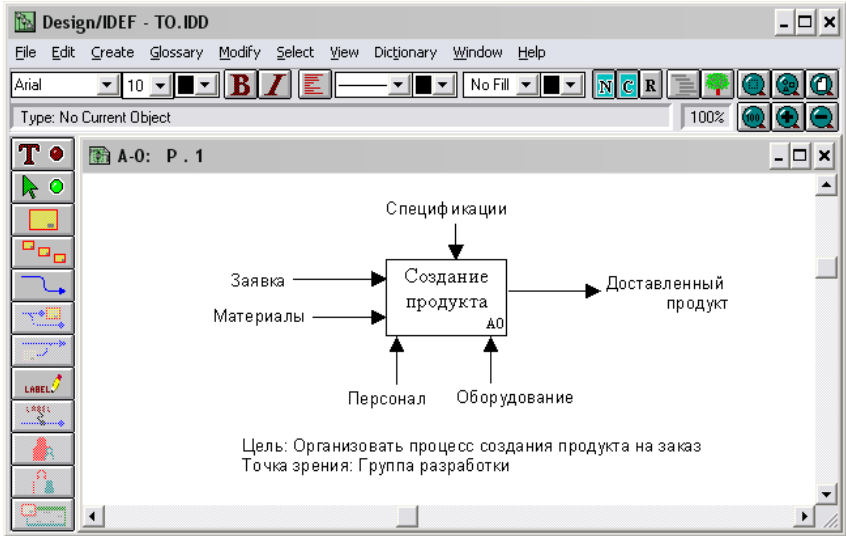




Рис. 1.4. Создание внешних дуг


Выберите File/Save, чтобы сохранить последние изменения.

5. Создание диаграммы первого уровня

Чтобы создать новую дочернюю диаграмму, содержащую функциональные блоки, детализирующие содержание родительского блока A0, необходимо этот блок декомпозировать.

Выберите блок A0, щелкнув на нем мышью. Выберите команду меню Create/Decompose или щелкните по кнопке  на специальной панели инструментов. Будет создана новая страница P2 для изображения диаграммы следующего уровня. При этом возле правого нижнего угла блока A0 появится метка «P2».

Для перехода на созданную дочернюю страницу выберите команду меню Select/Child или щелкните по кнопке  на специальной панели инструментов. То же самое можно сделать, выделив блок A0 и дважды щелкнув на нем мышью.

Вернуться на родительскую страницу P1 можно, выбрав команду меню Select/ Parent или щелкнув по кнопке . Кроме того, перейти на нужную Вам страницу можно, выбрав ее имя в меню Window.

Перейдите на страницу P2. Атрибуты страницы P2 будут автоматически заполнены.

Текст меток ("портовых узлов") блока A0 переносится на созданную страницу по ее краям соответственно расположению на родительской диаграмме.

Портовые узлы кроме текста меток будут содержать и так называемый ICOM-код. ICOM-код состоит из букв, показывающих роль соответствующей дуги: I – Input, C – Control, O – Output, M – Mechanism. Номер после буквы указывает позицию дуги в группе дуг, выполняющих ту же роль. Например, если имеется две дуги входа, то первая из них (сверху) будет иметь код I1, а вторая (снизу) – I2.

Таким образом, на странице P2 должны быть метки: I1 «Заявка», I2 «Материалы», C1 «Спецификации», M1 «Персонал», M2 «Оборудование» и O1 «Доставленный продукт».


Чтобы увидеть созданную IDEF-страницу всю целиком, выберите команду View/Reduce. Чтобы вернуться к стопроцентному масштабу выберите View/Enlarge. Масштаб можно задавать в диалоговом окне, вызываемом командой меню View/Zoom.... Изменять масштаб можно также с помощью группы кнопок стандартной панели инструментов, расположенных в правом верхнем углу окна, имеющих изображения в виде лупы.

6. Размещение IDEF-блоков

Диаграмма первого уровня будет содержать три функциональных блока, детализирующих блок A0 «Создание продукта»:

- A1 «Прием заявки»;
- A2 «Изготовление продукта»;
- A3 «Доставка продукта».


Для того, чтобы создать блоки, можно воспользоваться командой меню Create/Place Boxes. Появится диалоговое окно, в котором можно задать нужное количество блоков (в данном случае 3).

То же самое можно сделать, щелкнув по кнопке  специальной панели инструментов и выбрав в списке выбора строку с цифрой 3.

На поле диаграммы появится три блока, размещенных на равном расстоянии по диагонали страницы: от левого верхнего угла к правому нижнему. Блоки нумеруются автоматически.

Для того, чтобы вписать текст в блок А1, выделите его, включите текстовый режим (см. п.3) и напечатайте: «Прием заявки».

Аналогичным образом впишите текст «Изготовление продукта» в блок А2 и «Доставка продукта» в блок А3. Если текст не помещается в блок, можно увеличить ширину блока.

Вы в любой момент можете добавить еще один блок, выбрав команду меню Create/IDEF Vox или щелкнув по кнопке  специальной панели инструментов. Удалить блок можно, выделив его и нажав клавишу Delete.

7. Рисование дуг от портовых ICOM - узлов

Теперь необходимо соединить портовые узлы (метки I1, I2, C1, M1, M2, O1), которые были перенесены с родительской диаграммы, с блоками.

Выделите мышью входной портовый узел I1 и передвиньте его, чтобы он был расположен слева от блока А1. При этом метка «Заявка», связанная с этим узлом, тоже переместится.

Нарисуйте дугу от узла I1 к левой стороне блока А1 (см. п. 4).

Таким же образом создайте дуги:

- от узла I2 «Материалы» к левой стороне блока А2;
- от узла управления C1 «Спецификации» – к верхней стороне блока А2;
- от узла механизма M1 «Персонал» – к нижней стороне блока А1;
- от узла механизма M2 «Оборудование» – к нижней стороне блока А2;
- от правой стороны блока А3 – к выходному узлу O1 «Доставленный продукт».

Добавьте новый портовый узел, которого не было на родительской диаграмме: «Инструкции». Для этого создайте метку «Инструкции» повыше блока А1. Проведите от нее дугу к верхней стороне блока А1. Верхняя часть дуги будет помещена в туннель в виде круглых скобок: (). Это означает, что дуга идет от портового узла, который не был перенесен с родительской диаграммы.

У Вас должно получиться что-то похожее на рисунок 1.5.

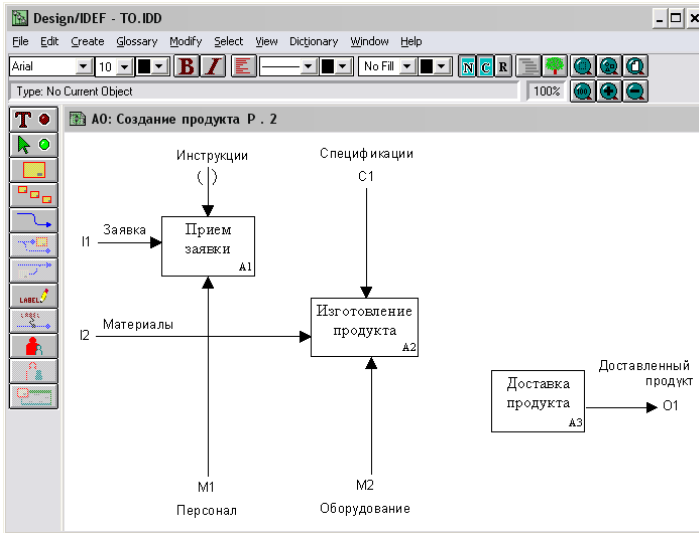


Рис. 1.5. Создание внешних дуг на диаграмме первого уровня

8. Рисование дуг, соединяющих блоки

Соединим теперь блоки друг с другом.

Выход блока A1 «Заказ» должен являться для блока A2 управлением, т.к. описание продукта, содержащееся в заказе, показывает, как должно осуществляться изготовление продукта.

Чтобы создать дугу, являющуюся выходом для блока A1 и управлением для блока A2, выберите команду меню Create/Arrow или соответствующую кнопку на специальной панели инструментов. Не отпуская кнопку мыши, соедините правую сторону блока A1 с верхней стороной блока A2. Отпустите кнопку мыши.

Если дуга получилась неправильно (например, вошла не с той стороны блока, с которой нужно), выделите ее, нажмите клавишу Delete и попробуйте нарисовать дугу заново.


Выход блока A2 «Продукт» должен являться входом для блока A3. Создайте дугу от правой стороны A2 к левой стороне A3 аналогично тому, как Вы создавали дугу, соединяющую блоки A1 и A2. Нажмите Esc.

Выберите команду меню File/Save, чтобы сохранить изменения.


9. Создание присоединенных меток

Чтобы пометить дуги, соединяющие блоки A1, A2 и A3 друг с другом, создадим для каждой дуги присоединенную метку.

Выберите команду меню Create/Label или соответствующую кнопку на специальной панели инструментов. Поместите указатель мыши чуть выше горизонтального сегмента дуги, идущей от блока A1 к блоку A2. Напечатайте: Заказ. Нажмите Esc.

Выберите команду меню Create/Attach Label или щелкните по кнопке  на специальной панели инструментов. Поместите указатель мыши на горизонтальный сегмент дуги напротив метки. Когда изображение дуги начнет мигать, щелкните левой кнопкой мыши. Метка будет соединена с дугой линией. Если Вы будете перемещать метку, она все равно будет соединена с дугой.

Аналогично создайте метку «Продукт» правее вертикального сегмента дуги, соединяющей блоки A2 и A3. Присоедините созданную метку к этому сегменту дуги.

Если Вы присоединили метку не в том месте дуги, Вы можете отсоединить метку. Для этого нужно выделить метку и выбрать команду меню Create/Detach Label или кнопку  на специальной панели инструментов.

Попробуйте отсоединить и заново присоединить любую из меток.

Выберите File/Save, чтобы сохранить изменения.

1.2. Самостоятельная работа

1. Выберите бизнес-процесс, для которого будет разрабатываться IDEF0-модель. Список рекомендуемых бизнес-процессов приведен в приложении 1. По согласованию с преподавателем Вы можете выбрать процесс не из списка.

2. Соберите информацию о выбранном бизнес-процессе, в том числе:

- об основных этапах выполнения процесса;
- о результатах выполнения каждого этапа и о том, как используются эти результаты;
- о материалах, необходимых для выполнения каждого этапа, и источниках их получения;
- об исполнителях каждого этапа и об оборудовании, используемом при его выполнении;

- об управляющей информации, необходимой для выполнения каждого этапа.

3. На основе собранной информации создайте текстовое описание процесса и составьте рукописные IDEF0-диаграммы.

1.3. Выполнение индивидуального задания

1. Создайте новый проект, выбрав в меню команду File/New и задав имя файла, в котором будет храниться новая модель, формируемая по индивидуальному заданию.

2. Создайте контекстную диаграмму, содержащую в качестве блока A0 выбранный бизнес-процесс. Создайте внешние дуги. Задайте цель создания модели и точку зрения.

3. Декомпозируйте бизнес-процесс. Создайте диаграмму первого уровня. Сохраните модель.

ПРАКТИЧЕСКАЯ РАБОТА №2

СОЗДАНИЕ ДИАГРАММ ДЕКОМПОЗИЦИИ

2.1. Выполнение работы на примере «Создание продукта»

Цель работы: закончить создание диаграммы первого уровня модели «Создание продукта» и создать диаграмму декомпозиции на базе диаграммы первого уровня.


Порядок выполнения работы

1. Открытие модели

Откройте модель «Создание продукта», созданную на предыдущей практической работе, выбрав команду меню File/Open и выбрав имя файла. Созданная Вами на предыдущей работе модель будет загружена.

2. Создание разветвлений

Выход блока A1 «Заказ» должен являться управлением не только для блока A2, но и для блока A3, т.к. адрес клиента, содержащийся в заказе, в некотором роде управляет выполнением блока A3 (показывает, куда осуществляется доставка продукта). Для того, чтобы создать разветвление:

- выделите дугу, соединяющую блоки A1 и A2;
- выберите команду меню Create/Branch или кнопку  на специальной панели инструментов;
- поместите указатель мыши на верхнюю сторону блока A3 и, когда изображение блока мигает, щелкните левой кнопкой мыши.

Ветвь дуги будет проведена.

Теперь необходимо разветвить дугу от портового узла M1, т.к. «Персонал» является механизмом не только для блока A1, но и для блоков A2 и A3. Создайте ветви дуги, соединяющей M1 с блоком A1, которые соединяли бы M1 с блоками A2 и A3, аналогично тому, как Вы создавали разветвление дуги, выходящей из блока A1. При этом не забывайте, что местом присоединения ветвей должны быть нижние стороны блоков (указатель мыши нужно помещать на нижние стороны блоков).

Присоедините метки: к ветви, являющейся механизмом блока A1 – метку «Отдел приема заявок»; к ветви, являющейся механизмом блока A2 – метку «Цех»; к ветви, являющейся механизмом блока A3 – метку «Отдел доставки».

У Вас должно получиться что-то похожее на рисунок 2.1.

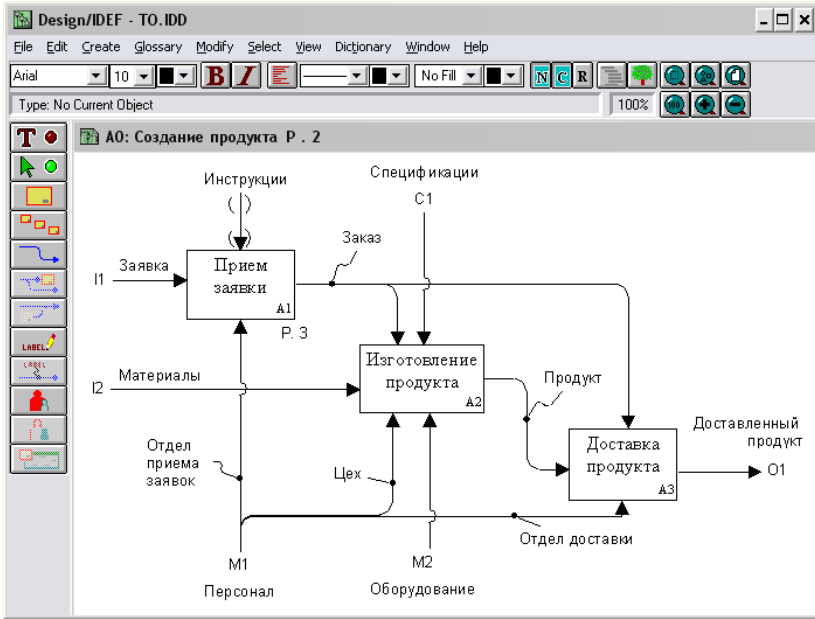


Рис. 2.1. Разветвление дуг

Выберите File/Save, чтобы сохранить изменения.

3. Создание мостов

Для того, чтобы места пересечения дуг были изображены в виде арки (мостов) выполните следующие действия:

- выберите команду меню Edit/Set Attributes;
- в диалоговом окне “IDEF Attributes” в группе кнопок “IDEF0”, расположенной в левой верхней части окна, выделите пункт Arrow;
- в списке выбора “Bridge Style” в средней части окна выберите пункт “Arcs”;

Щелкните на кнопке ОК.

4. Помещение дуги в туннель

Для того, чтобы дуга управления «Инструкции» в дальнейшем не переносилась на диаграммы более низких уровней модели, ее необходимо поместить в "туннель".

Выделите дугу «Инструкции». Выберите команду меню Create/Tunnel. В появившемся диалоговом окне пометьте поле, соответствующее Arrow Head. Щелкните мышью на кнопке ОК диалогового окна. Нижняя часть дуги будет помещена в туннель в виде круглых скобок: (). Это означает, что дуга не будет переноситься на диаграммы нижнего уровня.

5. Декомпозиция IDEF-блока

Для дальнейшей детализации блока A1 «Прием заявки» его нужно декомпонировать.

Создайте дочернюю диаграмму блока A1 аналогично тому, как вы создавали диаграмму первого уровня (см. п.5 лабораторной работы №1).

На дочерней диаграмме блока A1 разместите три блока: A11 «Выбор продукта», A12 «Оформление заказа» и A13 «Оплата». Размещение блоков описывается в п. 6 лабораторной работы №1.

Вы можете размещать блоки по одному. В этом случае блоки могут быть расположены на странице неравномерно. Чтобы равномерно расположить все блоки на странице, надо сгруппировать их, обведя все блоки выделяющим прямоугольником, и использовать команды раздвижения из меню Modify/Spread.

Для выравнивания блоков используются также команды меню Modify/Align. Выделите по очереди блоки, удерживая нажатой клавишу Shift (последним выделяется блок, по которому будет выравниваться остальные блоки) и попробуйте различные команды выравнивания.

Если Вы меняли размеры отдельных блоков и хотите сделать блоки одинакового размера, выделите по очереди блоки, удерживая нажатой клавишу Shift (последним выделяется блок, по которому будет равняться размер всех блоков) и используйте команды меню Modify/Same Size.

Вы можете также перенумеровать блоки, воспользовавшись командой меню Edit/Renumber Box.

6. Рисование дуг

Разместите на диаграмме метки и нарисуйте дуги так, чтобы получилось что-то похожее на рисунок 2.2.

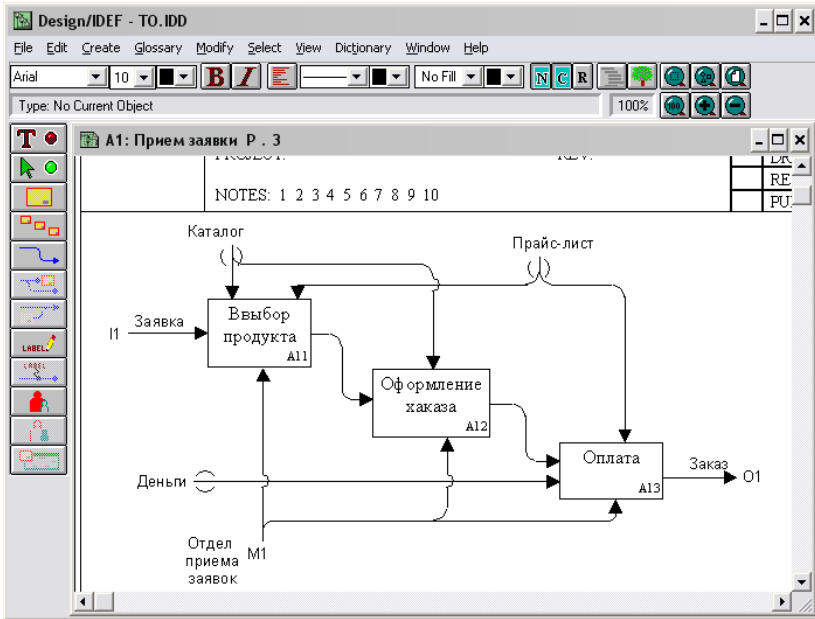


Рис. 2.2. Диаграмма декомпозиции блока A1

В процессе рисования дуг Вы перемещаете указатель мыши напрямую от начала дуги к ее окончанию. Иногда, при рисовании ломаной дуги гораздо удобнее рисовать ее сегментами. При построении сегментированной дуги:


- нарисуйте сначала первый прямой сегмент дуги, перемещая с нажатой левой кнопкой мыши указатель мыши от блока (метки) до точки изгиба;
- отпустите кнопку мыши;
- не меняя положения мыши, опять нажмите левую кнопку мыши и рисуйте следующий прямой сегмент дуги и т.д. пока не будет нарисован последний сегмент, заканчивающийся на стороне блока либо на метке.

Для перемещения дуги:

- выделите любую дугу;
- подведите указатель к метке у конца дуги и, нажав кнопку мыши, переместите указатель на новое место на той же стороне блока либо даже на сторону другого блока.

7. Создание соединений дуг

Дуга выхода O1 должна являться не только выходом блока A13, но и выходом блока A12, т.е. должна сливаться из выходов двух блоков. Для соединения (слияния) дуг:

- выделите дугу, соединяющую блок A13 с меткой O1;
- выберите команду меню Create/Join или кнопку  на специальной панели инструментов;
- поместите указатель мыши на середину правой стороны блока A12;
- когда контуры блока начнут мерцать, щелкните левой кнопкой мыши.

Выберите File/Save, чтобы сохранить изменения.

2.2. Самостоятельная работа

1. Соберите информацию о каждом из основных этапов выбранного бизнес-процесса, в том числе:

- о последовательности работ в ходе выполнения этапа;
- о результатах выполнения каждой работы и о том, как используются эти результаты;
- о материалах, необходимых для выполнения каждой работы, и источниках их получения;
- об исполнителях каждой работы и об оборудовании, используемом при ее выполнении;
- об управляющей информации, необходимой для выполнения каждой работы.

2. Представьте собранную информацию в виде текстового описания. Составьте рукописные IDEF0-диаграммы.

2.3. Выполнение индивидуального задания

1. Откройте файл, в котором хранится модель, созданная на предыдущей практической работе по индивидуальному заданию.

2. Создайте диаграммы декомпозиции для каждого из основных блоков диаграммы первого уровня. Сохраните модель.

ПРАКТИЧЕСКАЯ РАБОТА №3

СОЗДАНИЕ СЛОВАРЯ ДАННЫХ ДЛЯ IDEF-МОДЕЛИ

3.1. Выполнение работы на примере «Создание продукта»

Цель работы: Создать словарь данных для IDEF-модели «Создание продукта», созданной на предыдущих практических работах, в котором будет храниться информация о графических объектах модели.

Порядок выполнения работы

1. Открытие модели

Откройте модель «Создание продукта», созданную на предыдущей практической работе, выбрав команду меню File/Open и выбрав имя файла.

2. Создание словаря

Выберите команду меню Dictionary/Create. Появится диалоговое окно для наименования словаря. Введите имя словаря и щелкните на кнопке ОК. Появится окно ввода имени документа, посредством которого связывается IDEF-модель со словарем. Имя создается по умолчанию. В дальнейшем его будет невозможно изменить. Щелкните на кнопке ОК для ввода имени, указанного по умолчанию.

3. Определение типов записей

Прежде чем вводить в словарь конкретную информацию об объектах модели, надо определить типы записей, используемых для хранения этой информации.

Тип записи выступает в роли шаблона: он устанавливает формат или образец для основной информационной единицы словаря - записи. Записи состоят из отдельных единиц данных, называемых полями. Запись может иметь до семи полей. Записи одного типа имеют одинаковые поля, но значения полей у каждой записи будут уникальными.

В нашем словаре мы создадим два типа записей: *Функция* и *Кадры*.

Тип записи *Кадры* будет использоваться для описания подразделений персонала, таких как «Отдел приема заявок», «Цех» и «Отдел доставки». Для каждой из соответствующих меток на диаграмме пер-

вого уровня будет создана запись типа Кадры, в которой будет храниться информация о подразделении (руководитель, количество сотрудников, списочный состав, место расположения, телефон).

Создание типа записи Кадры:

- выберите команду меню Dictionary/Define Schema;
- появится окно выбора типа записей (для нового словаря оно пустое);
- щелкните на кнопке New для определения нового типа записи;
- в появившемся окне напечатайте в поле "Record Type" имя нового типа: Кадры (см. рис. 3.1);
- в поле "Number of fields" напечатайте значение 5 (см. рис. 3.1), т.к. для каждой записи о кадрах информация будет состоять из пяти полей;
- установку ассоциированных по умолчанию объектов и форм в группе "Default assignment" не меняйте (см. рис. 3.1);
- установку по умолчанию способа задания имен записей в группе "Default Record Naming Style" также не меняйте, т.к. имена записей Вы будете задавать сами (user suppliers name);
- щелкните на кнопке ОК.

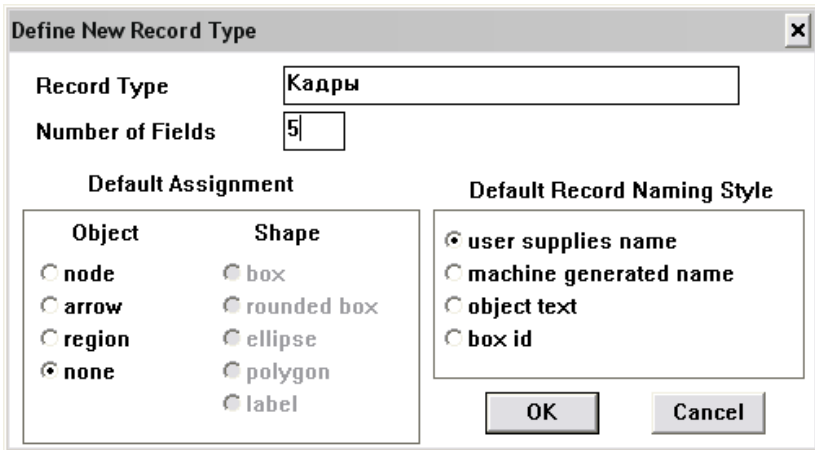


Рис. 3.1. Диалоговое окно определения нового типа записей

В появившемся диалоговом окне определения полей (см. рис. 3.2) напечатайте для каждого из пяти полей их имена: Руководитель, Количество сотрудников, Списочный состав, Место расположения, Телефон.

Для каждого поля установите тип данных:

- щелкните на кнопке "undeclared" рядом с именем поля;
- в появившемся окне выберите тип данных: для полей «Руководитель», «Место расположения» и «Телефон» - тип string (строка), для поля «Количество сотрудников» - тип integer (целое число), для поля «Списочный состав» - тип Description (описание, которое может иметь максимальный размер в 256 символов).



Рис. 3.2. Диалоговое окно определения полей

Щелкните на кнопке ОК. И в окне определения нового типа записей тоже щелкните на кнопке ОК.

Тип записей Кадры создан.

Второй тип записи - Функция - будет использоваться для описания функциональных блоков модели. Для каждого блока будет создана запись данного типа, имеющая то же название, что и блок, и содержащая информацию о соответствующей функции (описание функции, средняя продолжительность ее выполнения, сотрудники, выполняющие функцию).

Создание типа записи Функция аналогично созданию типа Кадры. В диалоговом окне определения нового типа записей (аналогичному окну на рис. 3.1):

- в поле "Record Type" напечатайте имя нового типа: Функция;
- в поле "Number of fields" напечатайте значение 3 (три поля);
- в группе "Default Record Naming Style" вместо "user suppliers name" выберите "object text" (текст объекта), т.к. наименования запи-

сей будет совпадать с наименованиями блоков, для которых эти записи создаются.

В диалоговом окне определения полей (аналогичному окну на рис. 3.2):

- напечатайте для каждого из трех полей их имена: Описание, Продолжительность (час), Сотрудники;

- установите для поля «Описание» тип Description, для поля «Продолжительность (час)» – тип real (действительное число), для поля «Сотрудники» – тип structure (структура), т.к. в нем будет храниться ссылка на запись типа Кадры (заданного Вами ранее);

- когда Вы выберете тип structure, появится окно выбора типа записи, в котором будет содержаться два введенных ранее типа – Кадры и Функция;

- выберите тип Кадры и щелкните на кнопке ОК;

- щелкните на кнопке ОК в диалоговом окне определения полей и в окне определения нового типа записей.

Итак, словарь создан. Словарь хранится на диске в виде трех файлов с расширениями .IDX, .DAT и .INF. Система снабжает словарные файлы внутренним номером версии для указания времени их создания или использования в модели. Этот номер сверяется и используется для согласованности с внутренним номером версии модели.

Выберите команду меню File/Save для обновления номера версии вашей модели.

Каждый открытый вами словарь автоматически сохраняется в словарных файлах на диске, с которым вы работаете. Удаление этих файлов с диска удаляет словарь, но не корректирует ссылающуюся на него информацию из соответствующей модели. Чтобы файлы были согласованы с моделью, необходимо сохранять свою модель после каждого ее открытия. Рекомендуется создавать резервные версии трех словарных файлов на другом диске. Нельзя переименовывать созданный словарь.

4. Создание записей

После определения типов записей можно создавать записи.

Перейдите на диаграмму первого уровня созданной Вами IDEF-модели. Создадим запись типа Кадры для метки «Отдел приема заявок»:

- выделите метку «Отдел приема заявок»;

- выберите команду меню Dictionary/Create Record (создать запись);

- появится окно выбора типа записи, содержащее список введенных ранее типов (в данном случае в списке два типа – Кадры и Функция);
 - выделите тип Кадры в списке типов записей и щелкните на кнопке ОК;
 - появится окно для ввода имени записи, содержащее список ранее созданных записей (в данном случае список пустой);
 - напечатайте в поле имени записи: «Отдел приема заявок» и щелкните на кнопке ОК;
 - появится окно для ввода значений полей (рис. 3.3);
 - введите значение для поля «Руководитель» – Иванов А.В.;
 - введите значение для поля «Количество сотрудников» - 5;
 - щелкните на строке ввода для поля «Списочный состав» и в открывшемся окне Description введите список сотрудников отдела (фамилия и должность), щелкните на кнопке ОК, чтобы закрыть окно Description;
 - введите значение для поля «Место расположения» – адрес отдела и для поля «Телефон» – номер телефона.
- Щелкните на кнопке ОК. Запись «Отдел приема заявок» готова.

Field Name	Value	Data Type
Руководитель	Иванов А.В.	string
Количество сотрудников	5	integer
Списочный состав	Иванов А.В. - зав. отдело...	Description
Место расположения	ул Ленина, 5б, оф. 214	string
Телефон	44-22-33	string

Рис. 3.3. Окно задания значений полей записи

Создайте аналогичным образом записи для меток «Цех» и «Отдел доставки».

Создадим запись типа «Функция» для блока А1, расположенного на диаграмме первого уровня. Для этого:

- выделите блок А1 «Прием заявки»;
- выберите команду меню Dictionary/Create Record;
- в появившемся окне выбора типа записи выберите тип записи «Функция» и щелкните на кнопке ОК.

Появляется окно для ввода значений полей. При этом имя записи «Прием заявок» назначается автоматически по имени объекта, для которого создается запись (по имени блока), т.к. при определении типа «Функция» был установлен автоматический режим именования записей (object text).

В окне задания значений полей (аналогичному окну на рис. 3.3) введите значения полей записи «Прием заявок»:

- щелкните на строке ввода для поля «Описание» и в открывшемся окне Description напечатайте, например, такой текст: «Консультирование клиента. Оформление заказа. Прием оплаты.», щелкните на кнопке ОК, чтобы закрыть окно Description;

- введите значение для поля «Продолжительность (час)» - 0.5;
- щелкните на строке ввода для поля «Сотрудники/Кадры», при этом откроется окно для ввода имени записи (т.к. данное поле должно содержать ссылку на запись типа Кадры), содержащее список имен записей типа Кадры, введенных ранее;

- поскольку функцию «Прием заявок» выполняет Отдел приема заявок, для которого уже была создана запись «Отдел приема заявок», то выберите в списке эту запись и щелкните на кнопке ОК;

- появится окно задания значений полей для записи «Отдел приема заявок», содержащее ранее введенные значения полей;

- Вы можете отредактировать значения или оставить существующие значения;

- щелкните на кнопке ОК и Вы вернетесь в окно задания значений полей для записи «Прием заявок»;

- щелкните на кнопке ОК.

Запись «Прием заявок» создана. Вы вернулись на диаграмму первого уровня.

Создайте аналогичным образом записи для блоков А2 «Изготовление продукта» и А3 «Доставка продукта».

Вы также можете создать записи для блоков А11, А12, А13 диаграммы декомпозиции А1.

Сохраните все изменения.

5. Редактирование записей

Вы можете изменять созданные словарные записи. Выделите в модели объект (блок или метку), для которого уже была создана запись. В меню Dictionary появляется команда Edit Record (редактировать запись). Эта команда открывает окно задания значений полей записи, в котором можно изменить имя записи (если имя задается пользователем), изменить тип записи, изменить значения полей или удалить ссылку между объектом и записью.

Чтобы проверить все команды редактирования, создайте для блока A13 «Оплата», расположенного на диаграмме декомпозиции A1, запись «Оплата продукта» типа «Кадрь».

Для редактирования этой записи:

- выделите блок A13 «Оплата»;
- выберите команду меню Dictionary/Edit Record;
- в появившемся окне задания значений полей щелкните на кнопке Change Name (изменить имя);
- появится окно задания имени записи. Замените имя «Оплата продукта» на «Касса»;
- щелкните на кнопке ОК и Вы вернетесь в окно задания значений полей;
- щелкните на кнопке Change Type (изменить тип записи) и в появившемся окне выбора типа записи выберите тип Функция;
- щелкните на кнопке ОК и Вы вернетесь в окно задания значений полей (обратите внимание, что теперь записи автоматически присваивается имя «Оплата», совпадающее с именем блока, т.к. при определении типа Функция был выбран способ присвоения имени по имени объекта);
- измените значения полей записи;
- щелкните кнопку ОК, чтобы вернуться в модель.

Можно удалить ссылку на словарную запись. Запустите команду редактирования записи (Dictionary/Edit Record) для блока «Оплата». В окне задания значений полей щелкните на кнопке Delete Reference (удалить ссылку). Вы вернетесь к диаграмме. Сохраняя выделение блока «Оплата», откройте меню Dictionary. Команда Edit Record превратилась в Create Record, т.к. для данного блока больше нет ссылки на запись в словаре.

Если удаляемая ссылка последняя для данной записи, запись удаляется из словаря.

6. Просмотр словаря данных

Для того, чтобы просмотреть все записи, имеющиеся в текущем словаре, нужно открыть окно браузера «Data Dictionary Browse» (Просмотр словаря данных) с помощью команды меню Dictionary/Browse.

Откройте окно браузера. Имя словаря, связанного с моделью, приводится после слова Dictionary.

Окно состоит из двух панелей:

- левая панель содержит список типов записей (один из типов является выделенным),

- правая панель содержит список записей выделенного типа.

Просмотрите базу данных словаря, выделяя имя каждого типа записей.

Выделите любую из записей на правой панели окна браузера и выполните ряд команд с помощью командных кнопок, расположенных внизу окна:

- щелкните на кнопке "Edit". Появится окно задания значений полей для текущей записи. Вы можете изменить значения полей или имя записи (если имя задается пользователем);

- щелкните на кнопке "Ref Info". Появится окно, содержащее следующую информацию о текущей записи: тип записи, имя записи, количество объектов в текущем документе, ссылающихся на данную запись, количество объектов в других документах, ссылающихся на данную запись, количество структурных ссылок из других записей словаря на данную запись;

- щелкните на кнопке "Ref List", при этом Будет создан список объектов, ссылающихся на текущую запись словаря. Щелкните на кнопке ОК, чтобы вернуться к модели. Выберите в меню Select команду Next Reference. В модели будет выделен объект, который имеет ссылку на текущую запись. Каждый раз этой командой вы будете выделять следующий объект, ссылающийся на текущую запись, пока не будет достигнут конец списка ссылок. Командой Select/Previous Reference вы можете выделить предыдущий объект из списка ссылок;

- щелкните на кнопке "Delete Rec". Появится окно, содержащее информацию о ссылках на удаляемую запись и кнопки ОК, Cancel. Если Вы щелкните на кнопке ОК, текущая запись будет удалена из словаря.

С помощью команды "Delete Rec" Вы можете также удалить тип записи, если словарь не содержит записей данного типа.

3.2. Самостоятельная работа

1. Для выбранного в качестве индивидуального задания бизнес-процесса выделите основные типы объектов, информация о которых будет содержаться в словаре. Это могут быть:

- подразделения организации, участвующие в выполнении бизнес-процесса;
- документы (планы, заказы, проекты), создаваемые в ходе выполнения процесса;
- продукция (услуги), создаваемые в ходе выполнения процесса;
- этапы (подэтапы, работы) процесса.

2. Для каждого выделенного типа объектов составьте список атрибутов (полей данных), достаточных для описания данного типа. Определите, какого типа должны быть атрибуты.

3. Для каждого типа объектов составьте перечень конкретных объектов данного типа (содержащихся в модели в качестве меток или блоков). Определите, какими должны быть конкретные значения атрибутов для каждого из выделенных объектов.

3.3. Выполнение индивидуального задания

1. Откройте файл, в котором хранится модель, созданная на предыдущей практической работе по индивидуальному заданию.

2. Определите типы записей. Каждому типу объектов должен соответствовать свой тип записей. При определении типа записи каждому атрибуту сопоставляется поле данных.

3. Создайте записи для конкретных объектов, содержащихся в модели в качестве меток или блоков. При создании записи введите значения атрибутов в соответствующие поля данных. Сохраните словарь.

ПРАКТИЧЕСКАЯ РАБОТА №4

ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ПАКЕТА DESIGN/IDEF

4.1. Выполнение работы на примере «Создание продукта»

Цель работы: Ознакомиться с дополнительными возможностями пакета Design/IDEF:

- создание текстовых и FEO-страниц;
- вычисление стоимости на основе функций IDEF-диаграмм;
- создание и работа с деревом узлов.

Применить дополнительные возможности к IDEF-модели "Создание продукта", созданной на предыдущих практических работах.

Порядок выполнения работы

1. Открытие модели

Откройте модель "Создание продукта", созданную на предыдущих лабораторных работах, выбрав команду меню File/Open и выбрав имя файла.

2. Создание текстовой страницы

На конечном этапе разработки модели к каждой диаграмме должны быть приложены: страница с сопроводительным текстом; глоссарий, содержащий описания объектов модели; и, возможно, диаграммы FEO.

Сопроводительный текст дополняет содержание диаграммы. В нем коротко и четко описывается процесс, представленный на диаграмме. С одной и той же диаграммой может быть связано не более одной страницы текста.

Создадим текстовую страницу, содержащую описание диаграммы конечного уровня А-0. Для этого:

- выберите команду меню Create/New Page...;
- в открывшемся диалоговом окне в поле "Methodology" щелкните на кнопке в конце строки ввода, чтобы показать список выбора, и в списке выберите Text;
- щелкните на кнопке ОК, чтобы закрыть окно.

Текстовая страница P4 будет создана, и Вы переместитесь на нее. Специальная панель инструментов в левой части окна изменится. На ней будут представлены инструменты для рисования различных

фигур (прямоугольников, эллипсов, многоугольников и др.), линий, стрелок и т.д. (см. рис. 4.1), т.к. текст может содержать графические объекты и может помещаться внутри графических объектов.

Выберите на панели инструментов инструмент «Прямоугольник». Поместите прямоугольник на странице и растяните его, чтобы в него мог поместиться сопроводительный текст. Включите текстовый режим (выберите кнопку с буквой T на панели инструментов) и напечатйте текст внутри прямоугольника (см. рис. 4.1).

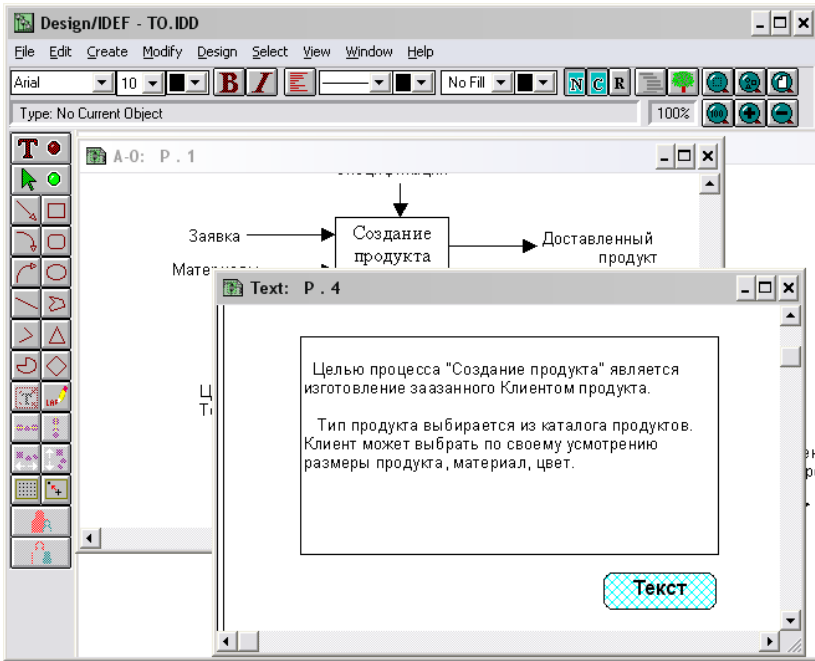



Рис. 4.1. Создание текстовой страницы

Чтобы связать диаграмму А-0 (на странице Р1) с текстовой страницей Р4, нужно поместить на нее объект, который будет работать как кнопка для перехода с одной страницы на другую.

Создайте такой объект на текстовой странице, например, в виде прямоугольника с закругленными углами и надписью «Текст» внутри (см. рис. 4.1). Можно его заштриховать и выделить цветом (используйте для этого стандартную панель инструментов, расположенную под строкой меню).

Чтобы переместить созданный объект на диаграмму А-0, выделите его и выберите команду меню Modify/Move to Page. В открывшемся окне выберите строку “А-0: Р1”. Объект будет перемещен на страницу Р1.

Перейдите на страницу Р1 с диаграммой А-0 (выбрав в меню Window соответствующее имя страницы). Выделите перемещенный объект. Выберите команду меню Modify/Attach и в открывшемся окне выберите строку “Text: Р4”. К выделенному объекту будет присоединена текстовая страница Р4. Теперь чтобы перейти на присоединенную страницу, достаточно дважды щелкнуть на данном объекте или выделить этот объект и нажать кнопку  на специальной панели инструментов.

Отсоединить страницу от объекта можно с помощью команды меню Modify/Detach.

3. Создание FEO-страниц

FEO-диаграммы (FEO, For Exposition Only - для экспозиции только) являются дополнительными IDEF0-диаграммами, которые выпадают из строгой иерархии моделей. На них могут отображаться альтернативные или дополнительные процессы. При их построении не обязательно придерживаться правил построения IDEF-диаграмм: они могут содержать более шести блоков, неполную структуру дуг, снабжаться пометками и т.д.

Создадим FEO-диаграмму, аналогичную диаграмме первого уровня А0 (альтернативную). Для этого:

- выберите команду меню Create/New Page...;
- в открывшемся диалоговом окне в поле “Methodology” введите FEO (или выберите в выпадающем списке);
- щелкните на кнопке ОК, чтобы закрыть окно.
- FEO-страница будет создана, и Вы переместитесь на нее;
- создайте диаграмму аналогичную диаграмме А0, но для случая, когда клиент заказывает уже готовый продукт (блок «Изготовление продукта» отсутствует).

Чтобы связать созданную FEO-страницу со страницей Р2, на которой помещена А0, нужно создать графический объект, который будет работать как кнопка перехода с одной страницы на другую. Создайте такой объект с надписью “ FEO” аналогично тому, как Вы создавали объект для перехода на текстовую страницу (создавать такой объект лучше на текстовой странице). Переместите его на страницу Р2. Перейдите на страницу Р2, выделите созданный объект и присоедини-

те к нему FEO-страницу с помощью команды меню Modify/Attach. Теперь данный объект будет служить кнопкой перехода.

Вы можете создать и кнопку возврата для перехода с присоединенной FEO-страницы на исходную страницу с диаграммой A0. Для этого создайте графический объект с надписью «Возврат» (так же, как Вы создавали объекты с надписями «Текст» и «FEO»), переместите его на FEO-страницу, выделите данный объект и присоедините к нему исходную страницу P2 с помощью команды меню Modify/Attach.

4. Вычисление стоимости на основе функций IDEF-диаграмм

Одной из возможностей глоссария (помимо описания объектов созданной IDEF-модели) является определение стоимости каждой функции (функционального блока) модели. Эта возможность позволяет Вам определять категории расходов для функций, задавать стоимости по каждой категории расходов, задавать длительность и частоту функций, а также рассчитывать общую стоимость для каждой функции. Информация о стоимости функции может экспортироваться и импортироваться в текстовые и табличные форматы.

Стоимость выполнения функции определяется через стоимость используемых ресурсов, представленных как входные дуги, дуги управления и механизмов. Можно выделить стандартные категории расходов (платы за используемые ресурсы), общие для всех функциональных блоков. Эти категории называются центрами стоимости (cost centers). Примеры центров стоимости:

Рабочая сила	зарплата исполнителей функции
Оборудование	амортизационные отчисления за используемое оборудование
Помещение	оплата за используемое помещение
Материалы	оплата расходных материалов
Управление	затраты на управление (составление графика работ, планирование и т.д.)

Стоимость выполнения любого функционального блока определяется как сумма стоимостей по всем центрам затрат. Таким образом, центры затрат трактуются, как статьи расходов на выполнение функции.

Стоимость декомпозированных функциональных блоков можно определять через стоимости дочерних блоков. Для этого сначала не-

обходимо задать частоту выполнения каждого из дочерних блоков (число раз, которое соответствующая функция выполняется в рамках выполнения родительской функции). Затем стоимость каждого дочернего блока умножается на его частоту и результаты складываются. При этом происходит суммирование по всем центрам затрат.

Для того, чтобы задать центры стоимости для всех функций:

- выберите команду меню Glossary/Activity Centers;
- в появившемся диалоговом окне введите имена для центров стоимости функций: Рабочая сила, Оборудование, Помещение, Материалы, Управление (см. рис. 4.2);
- щелкните на кнопке ОК.

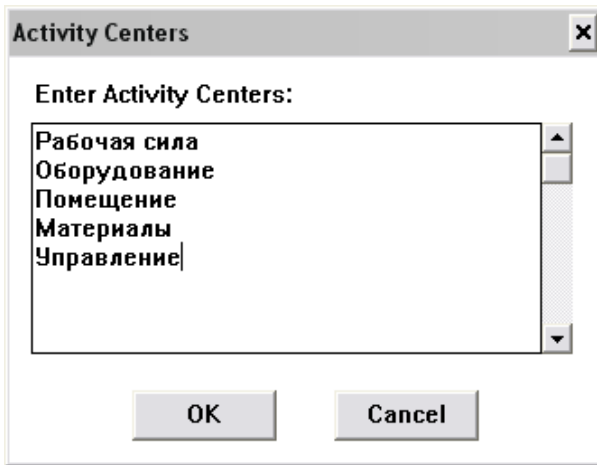


Рис. 4.2. Окно ввода центров стоимости

Задание стоимости функциональных блоков следует начинать с диаграмм нижнего уровня, т.к. по умолчанию стоимость декомпозируемых блоков складывается из стоимостей дочерних блоков. Однако, можно игнорировать это правило, если сделать установки, позволяющие непосредственно задавать стоимости родительских блоков.

Определим стоимость блока A1. Перейдите на диаграмму первого уровня A0. Выделите блок A1. Выберите команду меню Glossary/Cost Information.

Появится диалоговое окно (рис. 4.3). В левой части окна расположена панель для ввода стоимостей по каждому центру стоимости (строки ввода этой панели сначала будут недоступными для ввода и выделены серым цветом).

В правой части окна находится строка для ввода частоты выполнения функции и панель для ввода характеристик времени.

Снизу находятся установки режима вычисления стоимости:

- Value Added - если отмечена эта установка, стоимость родительской функции замещается автоматически, когда Вы вводите или изменяете стоимость для функций нижнего уровня (на диаграмме декомпозиции);

- Compute From Decompositions - если отмечена эта установка, стоимость родительской функции вычисляется через стоимости функций нижнего уровня.

Уберите галочку рядом с установкой “Compute From Decompositions” для того, чтобы можно было непосредственно задать стоимость блока A1 вместо того, чтобы она автоматически вычислялась через стоимости дочерних блоков A11, A12, A13.

Cost Information

A1: Прием заявки

Cost Information

Рабочая сила	300
Оборудование	50
Помещение	20
Материалы	20
Управление	10
Total:	\$400.00
Freq. x Total:	

Frequency Multiplier 1.0

Time Information

Units Days

Duration 0.00000

Freq. x Duration

Value Added

Compute From Decompositions

OK

Cancel

Рис. 4.3. Окно задания стоимости функции

Введите значения стоимости для различных центров стоимости (категорий расходов функций).

Общая стоимость функции (Total) будет вычислена автоматически, как сумма стоимостей по всем центрам стоимости.

Вы можете также задать:

- частоту выполнения функции, т.е. количество повторений функции в рамках выполнения родительской функции – в строке Frequency Multiplier (по умолчанию частота равна 1);
- единицы измерения времени (день, неделя, месяц или год) – в строке Units панели Time Information;
- длительность функции (в заданных единицах времени) – в строке Duration панели Time Information.

Щелкните на кнопке ОК, чтобы закрыть окно.

Задайте аналогичным образом стоимости функциональных блоков A2 и A3.

Перейдите на диаграмму корневого уровня A-0. Выделите блок A0 и откройте окно задания стоимости через команду меню Glossary/Cost Information. Если Вы не будете убирать галочку рядом с установкой “Compute From Decompositions”, то стоимость блока A0 будет автоматически вычислена как сумма стоимостей дочерних блоков A1, A2, A3, умноженных на их частоту. Убедитесь в этом, подсчитав самостоятельно стоимость блока A0 через стоимости дочерних блоков.

Дополнительно глоссарий позволяет ввести детальное описание для функциональных блоков с помощью команды Glossary/Glossary Entry.

5. Создание и работа с деревом узлов

Дерево узлов графически представляет иерархию функциональных блоков IDEF-модели. Информация размещается на отдельной странице.

Для просмотра страницы, содержащей дерево узлов, выберите команду меню View/Node Tree или щелкните на кнопке с изображением дерева в панели инструментов, расположенной ниже меню.

Можно задавать различные формы представления дерева узлов. Чтобы поменять форму дерева:

- выберите команду меню Edit/ Set Options;
- в появившемся окне задания опций выберите режим Node Tree (в левой части окна внизу);
- в панели Tree Form (в средней части окна) выберите форму представления дерева;
- щелкните на кнопке ОК, чтобы закрыть окно.

Просмотрите дерево узлов, чтобы убедиться, что форма изменилась. Попробуйте последовательно все формы дерева узлов.

4.2. Самостоятельная работа

1. Выберите диаграммы модели бизнес-процесса, создаваемой по индивидуальному заданию, для которых можно было бы создать текстовые страницы с пояснительным текстом и FEO-страницы с альтернативными диаграммами. Определите содержание текстовых и FEO-страниц.

2. Выделите категории расходов, по которым будут задаваться стоимости выполнения отдельных этапов (работ) бизнес-процесса.

3. Выберите работы, содержащиеся на диаграммах нижнего уровня, созданных на предыдущих практических работах. Для каждой работы подсчитайте ее усредненную стоимость по каждой категории расходов. Расчет может выполняться для единицы продукции (услуги), т.е. для одного экземпляра процесса, для партии продукции, для вида продукции вне зависимости от количества. Варианты расчета: стоимость использования ресурса в единицу времени умножить на время выполнения работы; стоимость использования ресурса в месяц на данный вид работ поделить на количество единиц продукции в месяц. Например, для вычисления стоимости работы по категории «рабочая сила» можно умножить почасовую оплату исполнителя работы на количество часов, которое он тратит на выполнение этой работы; для вычисления стоимости оборудования для производства единицы продукции можно поделить месячную стоимость работы оборудования на количество изделий в месяц, изготавливаемых на нем.

4. Определите среднее время выполнения каждой работы.

5. Подсчитайте на калькуляторе стоимость всего бизнес-процесса и, суммируя стоимость отдельных работ. Аналогично подсчитайте время выполнения бизнес-процесса.

4.3. Выполнение индивидуального задания

1. Откройте файл, в котором хранится модель, созданная на предыдущих практических работах по индивидуальному заданию.

2. Создайте текстовые и FEO-страницы, свяжите их с диаграммами.

3. Задайте центры стоимости, соответствующие категориям расходов на выполнение этапов (подэтапов, работ) бизнес-процесса.

4. Введите значения центров стоимости, а также время выполнения для каждого из блоков на диаграммах нижнего уровня.

5. Сравните стоимость и время блока A0 со стоимостью и временем бизнес-процесса, вычисленными вами на калькуляторе.

ПРАКТИЧЕСКАЯ РАБОТА №5

ЗНАКОМСТВО С UML И RATIONAL ROSE. СОЗДАНИЕ ДИАГРАММЫ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

5.1. Выполнение работы на примере «Продажа продукта»

Цель работы: Ознакомиться с основными возможностями языка моделирования UML и основами работы с CASE-средством Rational Rose. Создать диаграмму вариантов использования для бизнес-процесса «Продажа продукта» с помощью инструментального средства Rational Rose.

Порядок выполнения работы

1. 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 и стоимости их разработки, 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, на-

чали 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.

2. 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. Модель, формируемая с помощью Rose, детально описывает, что система содержит и как функционирует.

3. Начало работы с Rational Rose

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. 5.1.

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.

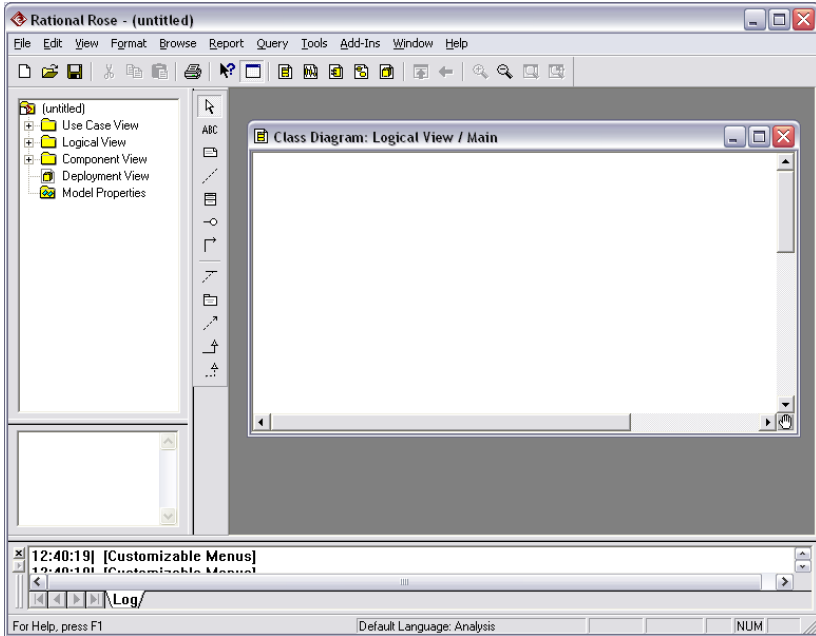


Рис. 5.1. Главное окно Rational Rose

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. Знакомство с основными понятиями диаграммы вариантов использования (Use case diagram)

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.

Пример диаграммы вариантов использования приведен на рис. 5.2. Основными элементами диаграммы являются варианты использования (use case), которые отображаются в виде эллипса, и акторы (actor), которые отображаются в виде фигуры «человечка».

Вариантом использования 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.

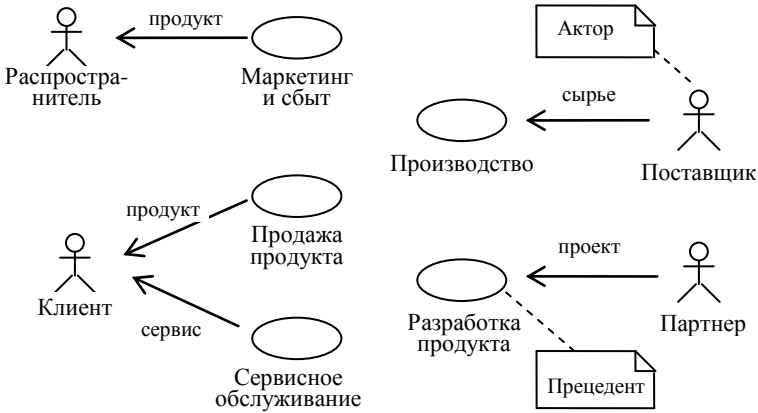


Рис. 5.2. Диаграмма вариантов использования

Актеры или действующие лица моделируют окружение системы. Это любые сущности, взаимодействующие с системой. Для информационных систем актерами являются пользователи, для бизнес-систем – клиенты, заказчики, поставщики, партнеры и т.д. Актеры взаимодействуют с системой через варианты использования.

Взаимосвязи между вариантами использования и актерами 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.

Между вариантами использования отношения ассоциации недопустимы, т.к. каждый из них описывает самостоятельный законченный процесс. Однако между ними могут быть отношения обобщения, означающие, что один из процессов является частным случаем другого, более общего процесса. Отношения обобщения могут быть установлены и между актерами.

Кроме того, между вариантами использования могут устанавливаться 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.

На диаграмму могут быть помещены также примечания, содержащие поясняющий текст для некоторого элемента диаграммы.

Таким образом, диаграмма вариантов использования может содержать следующие виды элементов:

- актер (Actor), который графически отображается в виде фигуры «человечка» под которой записывается имя актора;
- вариант использования (Use Case), который графически отображается в виде эллипса, под которым записывается имя.
- примечание (Note), которое может быть привязано к любому объекту диаграммы. Графически оно отображается в виде прямоугольника с «загнутым» правым уголком, внутри которого записывается текст примечания;
- отношение ассоциации (Association), которое графически отображается в виде сплошной линии со стрелкой на конце;
- отношение наследования или обобщения (Generalization), которое графически отображается в виде сплошной линии со стрелкой в виде незакрашенного треугольника;
- отношение зависимости или реализации (Dependency or instantiates), которое графически отображается в виде пунктирной линии со стрелкой на конце;
- якорь для замечания (Note Anchor), который позволяет соединить элемент Note с любым элементом на диаграмме. Графически отображается в виде пунктирной линии без стрелки.

Элементы на диаграмме могут располагаться на любом месте.

5. Построение Use case diagram











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 5.1.

Прежде всего, Вам необходимо поместить на диаграмму вариант использования (прецедент), соответствующий моделируемому бизнес-процессу «Продажа продукта». Для этого необходимо выполнить следующие действия:

- на панели инструментов выберите инструмент Use Case;
- щелкните мышью в окне диаграммы на любом свободном месте.

Таблица 5.1

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

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. Переименовать можно и другим способом: щелкните на выделенном элементе двойным щелчком и в открывшемся диалоговом окне можно ввести новое имя в поле Name.

После того, как Вы создадите элемент, его можно перемещать, удалять, изменять размеры. В любой момент Вы можете активизировать любой из созданных элементов диаграммы (для этого на панели инструментов должен быть активен инструмент выбора Selection Tool) 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).

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 (рис. 5.3). Для этого:

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

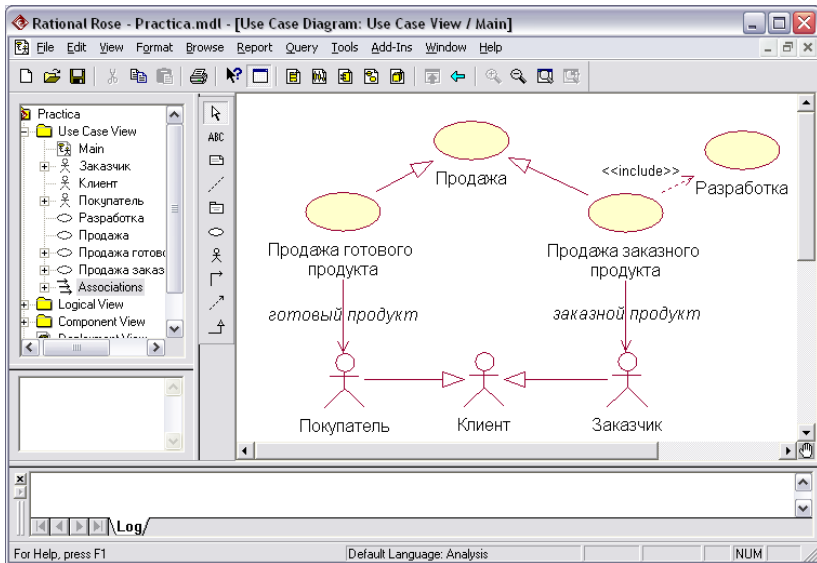


Рис. 5.3. Диаграмма вариантов использования в среде Rational Rose

Если некоторый бизнес-процесс содержит относительно самостоятельный фрагмент, то его можно выделить в самостоятельный прецедент, связанный с исходным прецедентом отношением включения. Например, из прецедента «Продажа заказного продукта» можно выделить фрагмент «Разработка» и создать соответствующий прецедент (см. рис. 5.3). Чтобы установить между прецедентами отношение включения нужно выполнить следующее:

- выберите инструмент Dependency or Instantiates (отношение зависимости),

- не отпуская кнопки мыши, перемещайте указатель от исходного прецедента к прецеденту-фрагменту;
- 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.

Обязательными элементами диаграммы Use case являются также акторы. 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.

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.

На диаграмме может быть размещено несколько акторов. Например, с прецедентом «Продажа готового продукта» взаимодействует актор «Покупатель», а с прецедентом «Продажа заказного продукта» – актор «Заказчик» (см. рис. 5.3). Можно добавить обобщенный актор (в данном случае «Клиент») и установить отношения обобщения между акторами.

6. Документирование

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, актор, отношение) и в окне Documentation введите поясняющий текст. 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.

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

Сохраните проект с помощью меню File→Save.

5.2. Самостоятельная работа

1. Выберите бизнес-процесс, для которого будет разрабатываться модель на языке UML из списка, приведенного в приложении 1. Это может быть тот же процесс, для которого Вы построили IDEF-модель по индивидуальному заданию на предыдущих практических работах, или другой процесс.

2. Соберите информацию о выбранном бизнес-процессе, в том числе:

- о входах и выходах процесса, т.е. о необходимых для выполнения процесса материалах, сырье, информации и о результатах выполнения процесса – продукции, услугах, документах;

- об окружении процесса, т.е. о субъектах, взаимодействующих с процессом – получающих его выходы или передающих что-либо на его вход.

3. Определите, каким образом можно структурировать процесс. Например, можно ли выделить различные версии бизнес-процесса, выполняемые в различных условиях; можно ли выделить крупные фрагменты процесса, выполняемые лишь при определенных условиях; можно ли выделить подпроцессы, расширяющие процесс. Можете также выделить обобщенный бизнес-процесс и обобщенные субъекты окружения.

4. На основе собранной информации составьте текстовое описание процесса (различных версий процесса, крупных фрагментов), его окружения, входов и выходов. Составьте рукописную Use case - диаграмму.

5.3. Выполнение индивидуального задания

1. Создайте новый проект, задав имя файла, в котором будет храниться новая модель, формируемая по индивидуальному заданию.

2. Создайте диаграмму Use case. На диаграмме должны быть: прецеденты, соответствующие выбранному процессу, его подпроцессам, его версиям; акторы, взаимодействующие с процессом; связи (отношения коммуникации) акторов с прецедентами; структурирующие отношения (обобщения, включения, расширения) между прецедентами или между акторами.

3. Документируйте основные объекты созданной диаграммы.

ПРАКТИЧЕСКАЯ РАБОТА №6

СОЗДАНИЕ ДИАГРАММЫ ДЕЯТЕЛЬНОСТИ

6.1. Выполнение работы на примере «Продажа продукта»

Цель работы: Создать диаграмму деятельности (Activity diagram) для бизнес-процесса «Продажа продукта» с помощью инструментального средства Rational Rose.

Порядок выполнения работы

1. Знакомство с основными понятиями 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.

Пример диаграммы деятельности показан на рис. 6.1.

Графически диаграмма деятельности представляется в форме графа, вершинами которого являются действия (операции, шаги процесса), а дугами – переходы от одного действия к другому. На диаграмме можно также отразить ветвление, т.е. возможность перехода к различным действиям в зависимости от некоторых условий.

Начальное состояние (Start State), соответствующее началу процесса, обозначается в виде закрашенного кружка; конечное состояние (End State), соответствующее завершению процесса – в виде закрашенного кружка, помещенного в окружность.

Действие (activity) изображается фигурой, напоминающей прямоугольник с закругленными сторонами, внутри которого записывается выражение действия.

Переход (Tranzition) 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, внутри которого нет никакого текста. В него может входить только одна стрелка. Выходящих стрелок может быть две или более. Для каждой из них указывается соответствующее условие, при котором выполняется данный переход.



Рис. 6.1. Диаграмма деятельности прецедента «Продажа»

2. Построение Activity diagram















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

Откройте проект «Продажа продукта», созданный во время выполнения практической работы №5, через меню File→Open. В окне браузера раскройте пакет представления вариантов использования (Use case view) и активизируйте диаграмму Main.

В окне диаграммы выберите вариант использования, для которого будете создавать диаграмму деятельности. Установите курсор на этот вариант и активизируйте всплывающее меню щелчком правой кнопки мыши. Выберите в меню Sub Diagrams → New Activity Diagram.

На рабочем столе появится пустое окно диаграммы деятельности. Специальная панель инструментов примет вид, соответствующий данному виду диаграмм. Назначение инструментов панели поясняется в таблице 6.1.

Таблица 6.1

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
	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 Synchronization	помещает на диаграмму новый символ горизонтальной синхронизации
	Vertical Synchronization	помещает на диаграмму новый символ вертикальной синхронизации
	Desizion	помещает на диаграмму новое ветвление
	SwimLane	помещает на диаграмму новую дорожку

Для того, чтобы поместить на диаграмму некоторый элемент (начальное или конечное состояние, действие, ветвление), необходимо выполнить следующие действия:

- на панели инструментов выберите нужный инструмент (Start State, End State, Activity, Desizion);
- 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 нужно ввести описание действия.

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.

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.

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. 6.2).

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

Еще одна полезная возможность – синхронизация действий, т.е. возможность показать параллельные потоки действий. Для этого используется специальный символ в виде отрезка горизонтальной или вертикальной линии, который 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).

На диаграмме, представленной на рис. 6.2, после выполнения действия «Принять заказ» происходит разделение на два параллельных потока, содержащие соответственно действия «Принять оплату» и «Изготовить продукт». А потом эти потоки сливаются.

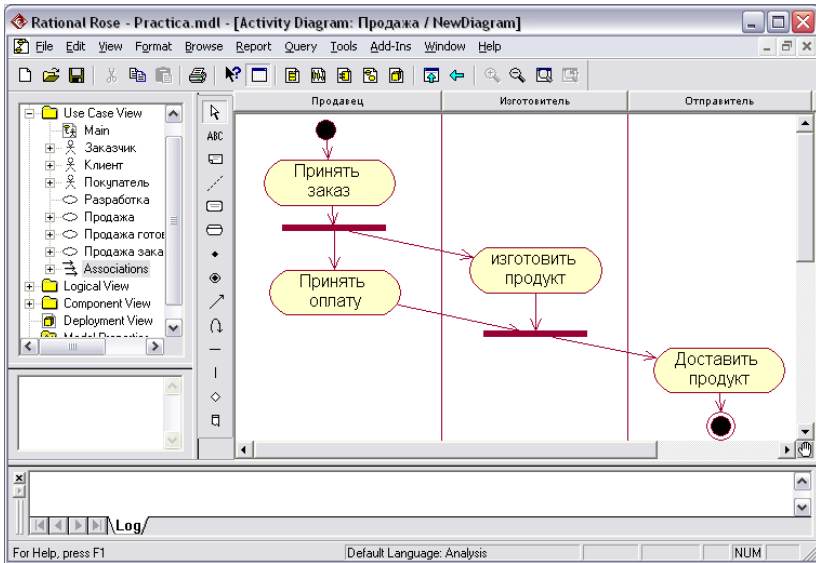


Рис. 6.2. Диаграмма деятельности в среде Rational Rose

Чтобы поместить на диаграмму символ синхронизации выберите инструмент Horizontal Synchronization или Vertical Synchronization и щелкните мышью в окне диаграммы.

В заключение можете документировать те действия, которые нуждаются в подробном объяснении. Сохраните проект.

6.2. Самостоятельная работа

1. Соберите информацию о ходе выполнения выбранного бизнес-процесса, его отдельных версий и фрагментов.
2. Составьте рукописные диаграммы деятельности.

6.3. Выполнение индивидуального задания

1. Откройте файл, в котором хранится модель, созданная на предыдущей практической работе по индивидуальному заданию.
2. Создайте диаграммы деятельности для каждого из прецедентов, представленных на диаграмме Use case.

ПРАКТИЧЕСКАЯ РАБОТА №7

СОЗДАНИЕ ДИАГРАММ ВЗАИМОДЕЙСТВИЯ ОБЪЕКТОВ

7.1. Выполнение работы на примере «Продажа продукта»

Цель работы: Создать диаграммы последовательности взаимодействия объектов (Sequence diagram) и диаграммы кооперации (Collaboration diagram) для бизнес-процесса «Продажа продукта» с помощью инструментального средства Rational Rose.

Порядок выполнения работы

1. Знакомство с основными понятиями диаграммы последовательности (Sequence diagram)

Диаграмма вариантов использования показывает функции системы, диаграммы деятельности показывают в деталях ход выполнения процессов. 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. Диаграммы взаимодействия, к которым относятся диаграмма последовательности и диаграмма кооперации, как раз и отражают взаимодействие объектов. Диаграмма последовательности позволяет отразить динамику взаимодействий между участниками процесса, а диаграмма кооперации отражает статический взгляд на взаимодействие, абстрагируясь от последовательности действий.

Взаимодействие объектов в UML рассматривается, прежде всего, как коммуникация, т.е. передача сообщений (информации). Однако взаимодействие можно рассматривать и как передачу вещества (продукции, деталей, материалов), понимая под сообщением не только информацию, но и передачу некоторого 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. Объектом, участвующим во взаимодействии может быть и актер.

Пример диаграммы последовательности приведен на рис. 7.1.

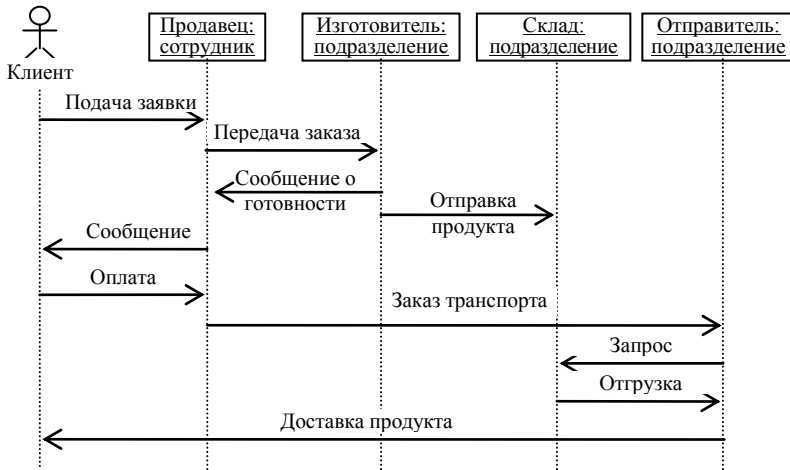


Рис. 7.1. Диаграмма последовательности прецедента «Продажа заказного продукта»

Графически каждый объект (Object), передающий или получающий сообщение, располагается в верхней части диаграммы и изображается в виде прямоугольника, от которого вниз идет пунктирная линия, называемая линией жизни объекта (см. рис. 7.1). Внутри прямоугольника записывается имя объекта и через двоеточие имя класса, к которому принадлежит объект (имя подчеркивается, что является признаком объекта).

Передача сообщения (Object Message) изображается в виде горизонтальной стрелки с именем сообщения от линии жизни объекта-инициатора к линии жизни объекта-получателя. Последовательность взаимодействий отражается через последовательное их расположение на диаграмме сверху - вниз. Другими словами, сообщения, расположенные на диаграмме выше, инициируются раньше тех, которые расположены ниже. Объекты тоже следует располагать в порядке их активности: крайним слева изображается объект, который является инициатором первого сообщения, правее – получатель сообщения и т.д.

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.

На диаграмме могут также располагаться комментарии или примечания (Notes), которые могут ассоциироваться с отдельными объектами или сообщениями.

2. Построение *Sequence diagram*

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 и откройте всплывающее меню щелчком правой кнопки мыши. Выберите пункт New → Sequence Diagram (см. рис. 7.2). Дайте новой диаграмме последовательности имя. Дважды щелкнув на диаграмме в браузере, откройте ее.

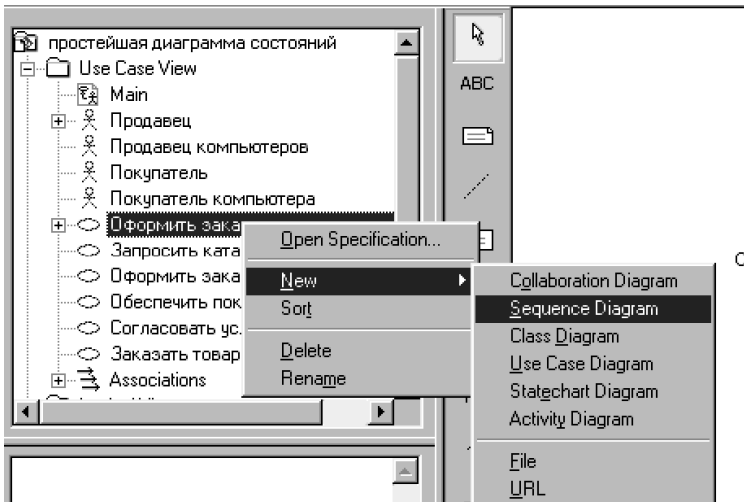









Рис. 7.2. Создание диаграммы последовательности

На рабочем столе появится окно диаграммы последовательности. Специальная панель инструментов примет вид, соответствующий данному виду диаграмм. Назначение инструментов панели поясняется в таблице 7.1.

Таблица 7.1.

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)

Пример диаграммы последовательности, построенной в среде Rational Rose, приведен на рис. 7.3.

Первым этапом создания диаграммы последовательности является выделение объектов-участников процесса.

Если инициатором сообщения является актер, его следует перенести с диаграммы вариантов использования, построенной в ходе выполнения практической работы №5. 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.

Чтобы создать объект (получателя или инициатора сообщений), на панели инструментов выберите инструмент Object и щелкните мышью в верхней части окна диаграммы правее уже помещенных объектов. Задать имя объекта можно двумя способами:

- щелкните на выделенном объекте и введите имя внутри прямоугольника, обозначающего объект;
- щелкните двойным щелчком на выделенном объекте и в появившемся диалоговом окне введите имя в поле Name.

Кроме того, в диалоговом окне 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.

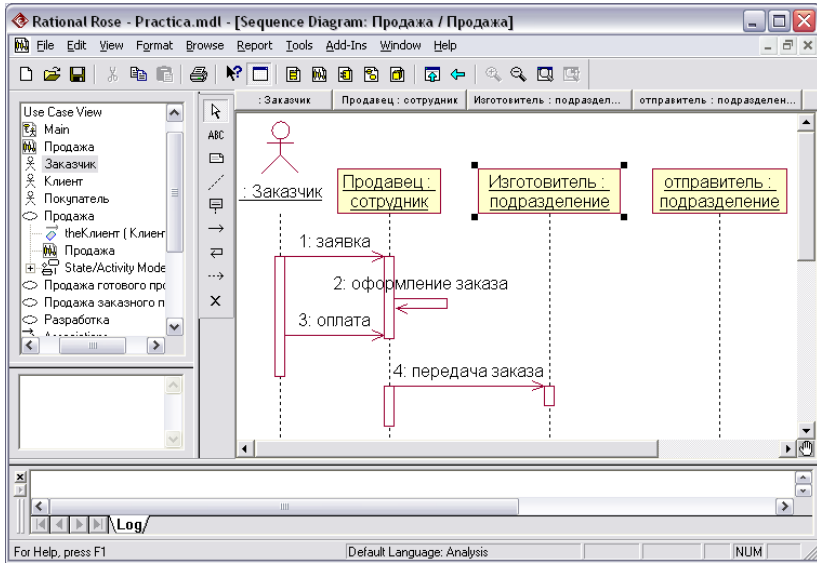


Рис. 7.3. Диаграмма последовательности в среде Rational Rose

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.

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 вве-

дите 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.

В процессе работы с диаграммой последовательности может потребоваться удалить ранее нарисованные сообщения. При этом оставшиеся сообщения будут автоматически перенумерованы.

Дополнительно можно поместить на диаграмму примечания (комментарии) и связать их с сообщениями. Вы можете также разместить на диаграмме произвольный текст (с помощью инструмента Text Box).

Последним этапом создания диаграммы является документирование элементов диаграммы (объектов и сообщений). Документация (поясняющий текст) на активный элемент вносится в окне Documentation.

Сохраните проект.

3. Построение 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 дру-

ую. При этом изменения, вносимые в одну из диаграмм, будут автоматически вноситься и на другую диаграмму.

Пример диаграммы кооперации, построенной в среде Rational Rose, приведен на рис. 7.4.

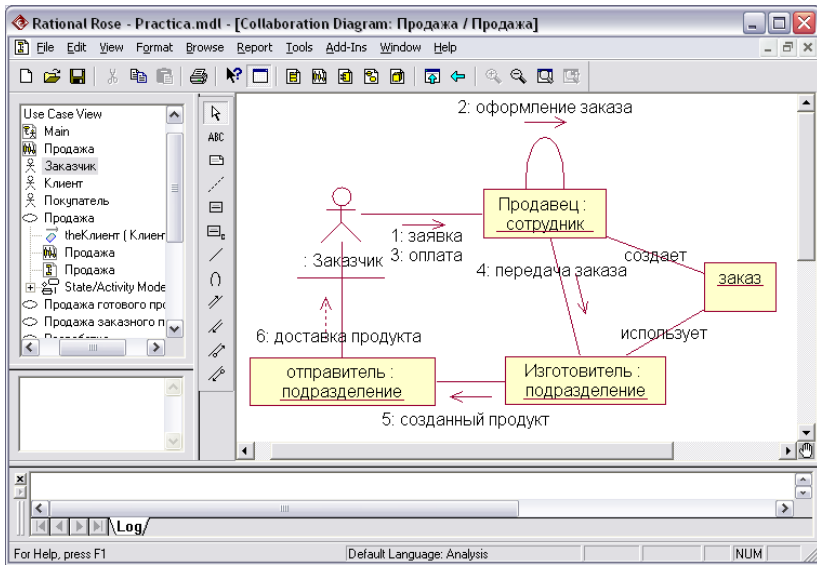


Рис. 7.4. Диаграмма кооперации в среде Rational Rose

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.

Можно поместить на диаграмму новые объекты, в частности объекты-сущности. Для этого необходимо выполнить следующие действия:

- на панели инструментов выберите инструмент Object;
- щелкните мышью в окне диаграммы на любом свободном месте;
- введите имя объекта (класса).

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.

Кроме динамических связей (сообщений) на диаграмме кооперации могут быть показаны и разнообразные статические связи (Link), которые изображаются отрезком прямой линии, соединяющей два прямоугольника объектов (без стрелки). Рядом с линией может записываться имя соответствующей связи.

Для того, чтобы установить статическую связь между элементами диаграммы, необходимо выполнить следующие действия:

- на панели инструментов выберите инструмент Object Link,
- не отпуская кнопки мыши, перемещайте указатель от одного элемента к другому.

Чтобы задать имя связи, щелкните на выделенной связи двойным щелчком и в открывшемся диалоговом окне введите имя.

Вы можете также разместить на диаграмме примечания с помощью инструмента Note и привязать его к некоторому элементу с помощью инструмента Anchor Note to Item или поместить произвольный текст с помощью инструмента Text Box.

Сохраните проект.

7.2. Самостоятельная работа

1. Соберите информацию о том, какие объекты участвуют в выполнении выбранного вами бизнес-процесса, а также о том, каким именно образом они взаимодействуют в ходе выполнения процесса.

2. Создайте рукописные диаграммы последовательности и кооперации для бизнес-процесса (его отдельных версий и фрагментов).

7.3. Выполнение индивидуального задания

1. Откройте файл, в котором хранится модель, созданная на практических работах №5 и №6 по индивидуальному заданию.

2. Создайте диаграммы последовательности для каждого из прецедентов, представленных на диаграмме Use case.

3. Преобразуйте диаграммы последовательности в диаграммы кооперации.

ПРАКТИЧЕСКАЯ РАБОТА №8

СОЗДАНИЕ ДИАГРАММЫ КЛАССОВ

8.1. Выполнение работы на примере «Продажа продукта»

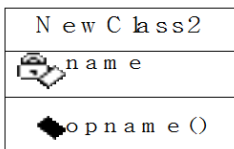
Цель работы: Создать диаграмму классов (Class diagram) для бизнес-процесса «Продажа продукта» с помощью инструментального средства Rational Rose.

Порядок выполнения работы

1. Знакомство с основными понятиями диаграммы классов (Class diagram)

Класс — это 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.

Класс изображается как прямоугольник, разделенный на три части (рис. 8.1, а): в верхней части записывается название класса, в середине — атрибуты, в нижней части — операции. Однако класс может иметь и другое изображение, связанное с категорией класса. В модели бизнеса используются две основных категории классов: business worker (исполнитель) и business entity (сущность). Обозначения этих категорий классов приведены на рис. 8.1, б и рис. 8.1, в.

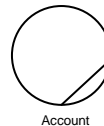


а



Clerk

б



Account

в

Рис.8.1 Обозначения классов

Диаграмма классов позволяет отразить не только классы объектов, участвующих в выполнении бизнес-процесса, но и отношения между классами.

На рис. 8.2 представлен пример диаграммы классов, построенной для прецедента «Продажа продукта».

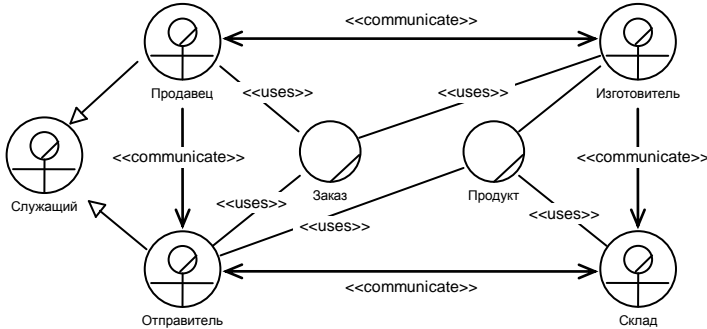


Рис. 8.2. Диаграмма классов для прецедента «Продажа продукта»

На диаграмме отражены классы исполнителей, выполняющих прецедент (Продавец, Исполнитель, Склад, Отправитель), а также классы объектов-сущностей, используемых в ходе выполнения прецедента (Заказ, Продукт). Между классами исполнителей установлены отношения *коммуникации* (ассоциации со стереотипом *communicate*), отражающие их взаимодействие. Между классами объектов-сущностей, как правило, 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 создает Заказ, Изготовитель использует Заказ для получения описания продукта, Отправитель использует Заказ для получения информации о том, куда доставлять продукт. Соответствующие отношения использования представлены на рис. 8.2.

На диаграмме классов могут быть отражены также отношения структурирования – обобщения и включения. Так, на рис. 8.2 показаны отношения обобщения между абстрактным классом Служащий и более конкретными классами Продавец, Отправитель.

2. Построение Class diagram

Создайте новую диаграмму классов для прецедента «Продажа заказного продукта» аналогично тому, как вы создавали диаграмму последовательности (см. п.2 параграфа 7.1, рис. 7.2), 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 8.1.

Таблица 8.1.

Пиктограмма	Наименование	Назначение
	Class (Класс)	Добавляет на диаграмму новый класс
	Interface (Интерфейс)	Добавляет на диаграмму 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 класс, 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, «Продавец».

Вы можете переименовать класс, задать его стереотип, добавить атрибуты и операции в окне спецификации класса. Для вызова окна щелкните двойным щелчком на выделенном классе. Пример окна спецификации класса приведен на рис. 8.3.

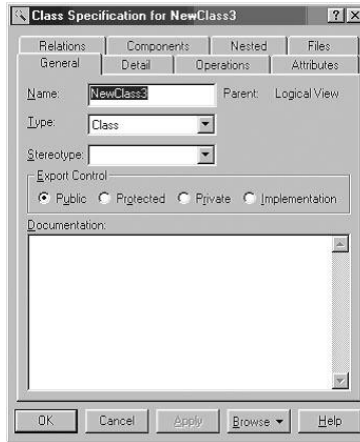


Рис. 8.3. Окно спецификации класса

Задайте стереотип класса, выбрав его из выпадающего списка в поле *Stereotype*. Например, для класса «Продавец» подойдет стереотип *business worker*. Щелкните кнопку **OK**. Вы увидите, что изображение класса поменялось.

Для добавления атрибута к классу щелкните правой кнопкой мыши на классе диаграммы. В открывшемся меню выберите пункт *New Attribute (Новый Атрибут)*. Введите имя атрибута в формате **Имя : Тип данных = Начальное значение**. Начальное значение необязательно.

Можно ввести атрибуты в окне спецификации. Откройте окно спецификации класса. 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, тип данных и значение по умолчанию в соответствующих колонках.

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, пред-

ставленной на рис. 7.1, можно определить, что к обязательствам объекта Продавец относятся: Прием заявки, Оформление заказа, Передача заказа Изготовителю, Прием сообщения о готовности продукта, Сообщение клиенту о готовности продукта, Прием оплаты, Заказ транспорта. Данные операции должны быть внесены в спецификацию соответствующего класса Продавец.

Следующим этапом построения диаграммы классов является установление отношений между классами. Для того, чтобы показать, что объекты одного класса взаимодействуют с объектами другого класса, устанавливаются отношения ассоциации между соответствующими классами. Выберите инструмент Association (Ассоциация) на панели инструментов. Проведите мышью линию ассоциации от одного класса к другому.

Для задания имени отношения выделите его. Введите имя. Или откройте окно спецификации отношения. Перейдите на вкладку General (Общие). Введите имя отношения в поле имени. На этой же вкладке можете ввести стереотип в поле Stereotype. Если отношение связывает классы со стереотипом business worker, то оно, как правило, имеет стереотип «communicate». Если же ассоциация связывает класс business worker с классом business entity, то для нее может быть задан стереотип «uses».

При выборе имени отношения помните, что как правило это глагол или глагольная фраза, описывающая, зачем нужно отношение. Пример имени ассоциации между Продавцом и Заказом – «Формирует», между Изготовителем и Заказом – «Просматривает», между Изготовителем и Продуктом – «Создает».

На диаграмме классов могут быть отражены также отношения структурирования – обобщения и включения. С помощью **обобщений (generalization)** показывают отношения наследования между двумя классами. Введите класс Служащий, являющийся предком для классов Продавец и Отправитель. Добавьте в него атрибуты, общие для классов-потомков, например, «фамилия, имя, отчество», «стаж работы». Соответствующие атрибуты у классов-потомков можно удалить. Выберите кнопку Generalization на панели инструментов. Проведите линию обобщения от подкласса (потомка) к суперклассу (предку).

В заключение можете документировать основные элементы диаграммы. Для добавления к классу текстового описания, выделите класс в браузере или в окне диаграммы. Введите текст в окно документации. Или откройте окно спецификации класса и введите информацию в области Documentation.

8.2. Самостоятельная работа

1. Просмотрите диаграммы последовательности и кооперации, построенные для выбранного вами бизнес-процесса на предыдущем занятии. Выделите классы для активных объектов, представленных на этих диаграммах. Добавьте классы для объектов-сущностей, обрабатываемых или создаваемых в ходе выполнения процесса. Введите обобщенные классы.

2. Для каждого класса определите перечень атрибутов и перечень операций. Операции классов для активных объектов должны быть соотнесены с соответствующими сообщениями, представленными на диаграмме последовательности или кооперации. У классов объектов-сущностей операции вообще могут отсутствовать.

3. Определите отношения, связывающие классы. Для каждого отношения задайте имя, стереотип.

3. Создайте рукописные диаграммы классов для выбранного бизнес-процесса.

8.3. Выполнение индивидуального задания

1. Откройте файл, в котором хранится модель, созданная на практических работах №5-7 по индивидуальному заданию.

2. Создайте диаграммы классов для основных прецедентов, представленных на диаграмме Use case. На диаграммах обязательно должны быть представлены как классы активных объектов (категории business worker), так и классы объектов-сущностей (категории business entity). Обязательно должны присутствовать обобщенные классы (суперклассы). Должны быть показаны отношения коммуникации, использования, обобщения или включения.

САМОСТОЯТЕЛЬНАЯ РАБОТА ПО ИНДИВИДУАЛЬНОМУ ТВОРЧЕСКОМУ ЗАДАНИЮ

Выполнение творческого индивидуального задания заключается в применении некоторой методологии для моделирования конкретного бизнес-процесса и включает три этапа:

1. Освоение методологии моделирования и инструментального средства.
2. Сбор информации о бизнес-процессе, текстовое описание бизнес-процесса.
3. Построение модели бизнес-процесса с помощью инструментального средства, документирование модели.

Варианты заданий:

1. Создание модели бизнес-процесса по методологии IDEF3 с помощью инструментального средства BPWin.
2. Создание модели бизнес-процесса по методологии DFD с помощью инструментального средства BPWin.
3. Создание имитационной модели бизнес-процесса с помощью ПП “Arena”.
4. Исследование динамики показателей бизнес-процесса с помощью имитационного моделирования в iThink.
5. Создание модели eEPC бизнес-процесса с помощью инструментального средства ARIS.
6. Функционально-стоимостной анализ бизнес-процесса с помощью инструмента EasyABC.
7. Построение субъектно-ориентированной модели бизнес-процесса с помощью средства jPASS.
8. Построение модели ключевых показателей результативности с помощью инструментального средства ARIS.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Методология IDEF0. Стандарт. Русская версия. - М.: Метатехнология, 1993. - 107 с.
2. Калянов Г.Н. CASE-технологии: консалтинг в автоматизации бизнес-процессов. – М.: Горячая линия-Телеком, 2000. - 318 с.
3. Маклаков С.В. Моделирование бизнес-процессов с AllFusion Process Modeler (BPwin 4.1). – М.: ДИАЛОГ-МИФИ, 2003. – 240 с.
4. Леоненков А.В. Самоучитель UML. – СПб.: БХВ-Петербург, 2001. – 304 с.
5. Калашян А.Н., Калянов Г.Н. Структурные модели бизнеса: DFD-технологии; Под ред. Г.Н. Калянова. — М.: Финансы и статистика, 2003. – 256 с.
6. Войнов И.В., Пудовкина С.Г., Телегин А.И. Моделирование экономических систем и процессов. Опыт построения ARIS-моделей: Монография. — Челябинск: Изд. ЮУрГУ, 2002. – 392 с.
7. Замятина О.М. Компьютерное моделирование: учеб. пособие. – Томск: изд-во ТПУ, 2007. – 115 с.
8. Шеер А.-В. Моделирование бизнес-процессов. Изд. 2-е, перераб. и доп. — М.: Весть-МетаТехнология, 2000. – 221 с.
9. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М: Финансы и статистика, 1998. – 176 с.
10. Функционально-стоимостной анализ // Корпоративный менеджмент [Электронный ресурс]. – 2008. – Режим доступа: ww.cfin.ru/management/controlling/iso_abc.shtml.
11. 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.
12. JPASS! Business Process Modeling [Электронный ресурс]. – Режим доступа: – <http://www.jcom1.com/cms/jpass.html?&L=1> (дата обращения 24.11.2009).

ВАРИАНТЫ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

1. Ремонт квартир
2. Ремонт автомобилей
3. Проведение праздничных мероприятий (свадеб, юбилеев и т.д.)
4. Организация турпоездки
5. Пошив и ремонт верхней одежды
6. Проведение рекламных компаний
7. Оказание услуг по операциям с недвижимостью
8. Гостиничное обслуживание
9. Организация выставок и ярмарок
10. Издание печатной продукции
11. Продажа и ремонт компьютеров
12. Производство и продажа мебели на заказ
13. Трудоустройство
14. Организация обучения и консультирования
15. Оказание жилищно-коммунальных услуг
16. Оказание услуг по автоперевозкам (пассажирским и/или грузовым)
17. Организация спортивных мероприятий (турниров, игр и т.д.)
18. Изготовление кондитерских изделий (тортов, пирожных)
19. Выпуск газеты
20. Оказание медицинских услуг
21. Оказание маркетинговых услуг
22. Организация выборных компаний
23. Оказание услуг брачного агентства
24. Производство, продажа и сопровождение программной продукции
25. Строительство гаражей, садовых домиков и т.д.