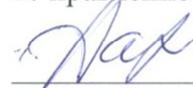


Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Томский государственный университет систем управления и радиоэлектроники»
(ТУСУР)

УТВЕРЖДАЮ

Заведующий кафедрой
«Управление инновациями»



Г.Н. Нариманова
(подпись) (ФИО)

" 28 " февраля 2018 г.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
К ЛАБОРАТОРНЫМ РАБОТАМ

по дисциплине

«Информационные технологии в управлении качеством»

Составлены кафедрой

«Управление инновациями»

Для студентов, обучающихся
по направлению подготовки бакалавров
27.03.02 «Управление качеством»

Форма обучения: очная

Составитель
Доцент каф. УИ, к.ф.-м.н.

Годенова Е.Г.

28 "февраля" 2018 г.

Томск, 2018 г.

СОДЕРЖАНИЕ

МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП	3
ВВЕДЕНИЕ	5
ГЛАВА I. МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ IDEF0.....	10
ЛАБОРАТОРНАЯ РАБОТА № 1	21
ЛАБОРАТОРНАЯ РАБОТА № 2.....	30
ЛАБОРАТОРНАЯ РАБОТА № 3	49
ЛАБОРАТОРНАЯ РАБОТА № 4.....	54
ЛАБОРАТОРНАЯ РАБОТА № 5	63
ГЛАВА II. НОТАЦИЯ VRMN ДЛЯ МОДЕЛИРОВАНИЯ	
БИЗНЕС ПРОЦЕССОВ.....	66
ЛАБОРАТОРНАЯ РАБОТА № 6.....	68
ЛАБОРАТОРНАЯ РАБОТА № 7.....	81
ЛАБОРАТОРНАЯ РАБОТА № 8.....	92
ЛАБОРАТОРНАЯ РАБОТА № 9.....	109
ГЛАВА III. УНИФИЦИРОВАННЫЙ ЯЗЫК МОДЕЛИРОВАНИЯ UML	112
ЛАБОРАТОРНАЯ РАБОТА № 10	113
ЛАБОРАТОРНАЯ РАБОТА № 11	132
ЛАБОРАТОРНАЯ РАБОТА № 12	147
ЛАБОРАТОРНАЯ РАБОТА № 13.....	156
ЛАБОРАТОРНАЯ РАБОТА № 14.....	168
ГЛАВА IV. ОБЩИЕ ПРИНЦИПЫ МЕТОДОЛОГИИ	
И СИСТЕМЫ ARIS	172
ЛАБОРАТОРНАЯ РАБОТА № 15.....	178
ЛАБОРАТОРНАЯ РАБОТА № 16.....	185
ЛАБОРАТОРНАЯ РАБОТА № 17.....	192
ЛАБОРАТОРНАЯ РАБОТА № 18.....	201
ЛАБОРАТОРНАЯ РАБОТА № 19.....	202
ЛАБОРАТОРНАЯ РАБОТА № 20.....	205
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ И ИСТОЧНИКОВ	206

МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Изучению дисциплины «Информационные технологии в управлении качеством» отводится одно из важнейших мест в учебном процессе подготовки бакалавров по направлению 27.03.02 «Управление качеством». Дисциплина входит в вариативную часть (Б1.В.ОД) учебного плана. Изучается в 7 и 8 семестрах в объеме 8 зачетных единиц (з.е.).

Целью курса «Информационные технологии в управлении качеством» является формирование у студентов компетенций в области применения современных информационных технологий в управлении качеством

Задачи курса:

- ✓ ознакомить обучающихся с современными нотациями моделирования бизнес-процессов организаций;
- ✓ сформировать у обучающихся навыки работы программными продуктами для визуализации, оценки и анализа эффективности деятельности организаций;

В результате изучения дисциплины студент должен:

Знать: нотации моделирования и программные продукты в области управления качеством;

Уметь: использовать информационные технологии для моделирования и анализа бизнес-процессов, разработки документооборота предприятий;

Владеть: навыками построения диаграмм для представления, моделирования, реинжиниринга бизнес-процессов, навыками анализа моделей бизнес-процессов.

По курсу «Информационные технологии в управлении качеством» предусмотрены лабораторные работы в двух семестрах, ориентированные на практическое закрепление лекционного материала и самостоятельной работы студентов. Лабораторные работы призваны сформировать у студентов компетенции в области разработки и анализа моделей бизнес-процессов организаций различного уровня с использованием разнообразных нотаций моделирования.

На изучение курса «Информационные технологии в управлении качеством» отводится два семестра. Лабораторные работы первого семестра посвящены изучению нотаций моделирования IDEF0, DFD, IDEF3, BPMN. Они являются базовым фундаментом для освоения принципов моделирования бизнес-процессов. Во втором семестре изучаются более популярные средства моделирования: унифицированный язык моделирования UML и набор методологий, входящий в одну единую методологию ARIS. Набор методологий ARIS также включает в себя элементы нотаций BPMN и UML. Однако в данном курсе они изучаются отдельно, с целью приобретения студентами навыков использования

программного обеспечения разных видов. Задания лабораторных работ направлены на приобретение студентами компетенций в области разработки моделей бизнес-процессов предприятия, как простого уровня, так и комплекса моделей. При этом в ходе выполнения лабораторных работ студенты осваивают различные нотации визуального моделирования бизнес-процессов и знакомятся с несколькими программными продуктами.

Требования к уровню знаний студентов

Для освоения студентами дисциплины «Информационные технологии в управлении качеством и защита информации» рекомендуется изучение дисциплин:

- 1) информатика;
- 2) информационное обеспечение, базы данных;
- 3) средства и методы управления качеством;

Ход выполнения лабораторных работ

Каждая лабораторная работа содержит три части:

- 1) теоретическая часть;
- 2) практическая часть;
- 3) контрольные вопросы по теме.

Для выполнения лабораторной работы студентам необходимо изучить материал, описанный в теоретической части работы. После изучения теоретического материала студенты выполняют задания лабораторной работы, описанные в практической части. После выполнения всех заданий лабораторной работы, студентам необходимо представить файл с отчетом о выполненной работе и защитить результаты лабораторной работы, ответив на контрольные вопросы. К защите лабораторной работы студенты допускаются только после выполнения всех заданий практической части.

ВВЕДЕНИЕ

Говоря об информационных технологиях в области управления качеством можно описать огромное количество сфер применения. Однако из таких сфер это автоматизация систем менеджмента качества (далее СМК), реализация процессного подхода. В качестве бакалавра по направлению «Управление качеством» студент должен обладать знаниями и умениями в области разработки и анализа бизнес-процессов, управления проектами и процессами.

Система менеджмента качества – это система, включающая в себя технические, организационные, информационные, людские, материальные и финансовые элементы. Управление их взаимодействием и совместной работой представляет собой сложную и комплексную задачу. Автоматизация СМК позволяет сделать этот процесс предсказуемым и упорядоченным [1].

Сложность и многообразие требований системы качества не позволяет автоматизировать СМК за счет применения какого-либо одного программного средства.

Различные направления и объекты автоматизации СМК заставляют применять разные продукты. Комплексная автоматизация возможна только за счет интеграции нескольких систем. Наиболее часто при автоматизации СМК выбирают функциональное направление, т.к. оно является «понятным» для сотрудников организации и специалистов по качеству [1].



Рис. 1.1. Средства автоматизации СМК [1]

Средства автоматизации реализуют часть функций системы качества, связанных с обработкой и передачей информации и данных.

К таким средствам относятся [1]:

- ✓ **CRM системы** автоматизируют передачу и обработку информации, связанной с потребителями и внешними сторонами. За счет этих программных средств возможно реализовать только часть требований системы качества. Применение CRM в СМК имеет много ограничений;

- ✓ **ERP системы** автоматизируют элементы системы качества, связанные с производственными процессами: управление производством, закупками, сбытом, обслуживанием и т.п. Исполнение процессов не автоматизируется. Эти системы только обрабатывают и передают информацию о процессах;

- ✓ **системы документооборота** дают возможность автоматизировать управление документацией и данными. Информация поступает из других подсистем (ERP системы, CRM системы) или вводится вручную;

- ✓ **специализированные системы СМК** создаются специально для автоматизации элементов системы качества, но их возможности ограничены. В большинстве случаев они позволяют управлять частью информации по отдельным вопросам системы качества: аудитам СМК, документацией, планами работ.

Для автоматизации исполнения процессов применяют другие системы и комплексы:

- ✓ **автоматизированные линии и оборудование.** Это производственные комплексы или роботизированные линии, которые способны выполнять производственные операции по заданной программе. Они применяются для автоматизации производственных и измерительных процессов и относятся к нижнему уровню автоматизации;

- ✓ **CAD \ CAM системы.** Применяются для автоматизации процессов проектирования. Они построены на основе CALS - технологий и позволяют в автоматическом режиме выполнять различные этапы проектных разработок;

- ✓ **BPMS.** Этот вид систем автоматизации можно отнести к процессным. Они предназначены для автоматизации бизнес-процессов. В полной мере выполнять бизнес-процессы в автоматическом режиме данные системы не могут, но они позволяют автоматизировать значительную часть управленческих операций;

- ✓ **CASE средства.** Предназначены для проектирования процессов. С помощью этих систем можно автоматизировать значительную часть операций по разработке порядка и условий выполнения процессов [1].

МЕТОДОЛОГИИ ОПИСАНИЯ БИЗНЕС-ПРОЦЕССОВ

Описание бизнес-процесса должно быть выполнено на таком языке, который отвечает требованиям выразительности, понятности, соответствия автоматизированному способу сопровождения моделей бизнес-процессов.

Модели бизнес-процессов создаются для определенных целей и отражают интересы определенной группы пользователей моделей. Наиболее удобными принято считать следующие нотации для описания бизнес-процессов [2]:

1. Семейство IDEF;
2. BPMN;
3. UML;

Анализ деятельности предприятий и реорганизация бизнес-процессов – чрезвычайно сложная задача, требующая методической и инструментальной поддержки.

Среди бизнес-аналитиков крупных компаний большую популярность имеет пакет ARIS Toolset (ARIS). Так, применительно к нашему региону, продуктом ARIS Toolset пользуются в своей деятельности такие гиганты как ПАО «СИБУР Ходинг», ООО «Газпром Трансгаз Томск», ООО «Томскнефтехим» и др.

ARIS – архитектура интегрированных информационных систем, методология и тиражируемый программный продукт для моделирования бизнес-процессов организаций. ARIS представляет собой векторный редактор, предназначенный для моделирования бизнес-процессов, в т.ч. для создания функциональных и поведенческих моделей информационных систем [3].

Лабораторные работы по курсу «Информационные технологии в управлении качеством» предусматривают изучение методологий IDEF0, IDEF3, DFD, BPMN, UML и ARIS.

Семейство IDEF

В 1981 году департамент ВВС США разрабатывал программу автоматизации промышленных предприятий, сокращенно называемую ICAM (Integrated Computer Aided Manufacturing). В ходе разработки был выпущен набор стандартов IDEF (ICAM Definition). В частности, одним из стандартов этого набора явилась нотация функционального моделирования IDEF0. Всего же набор IDEF включает семь методологий моделирования для описания различных областей и процессов предприятия: моделирование информационных потоков (IDEF1), построение реляционных структур (IDEF1X), динамического моделирования развития систем (IDEF2), документирования

процессов (IDEF3), построения объектно-ориентированных систем (IDEF4), отологического исследования сложных систем (IDEF5).

Одним из наиболее удобных инструментальных средств функционального моделирования является пакет AllFusion Process Modeler (BPWin). Этот пакет поддерживает три методологии — IDEF0, IDEF3 и DFD, каждая из которых решает свои специфические задачи. Для семейства IDEF негласно закрепились правила применения графических нотаций для различного вида моделей (табл. 1.1). В BPwin возможно построение смешанных моделей, т. е. модель может содержать одновременно диаграммы как IDEF0, так и IDEF3 и DFD. Состав палитры инструментов изменяется автоматически, когда происходит переключение с одной нотации на другую.

Таблица 1.1. Правила применения графических нотаций для различного вида моделей

Задачи описания бизнес-процесса	Тип модели	Используемый формат
1. Описание бизнес -процесса уровня компании	1. Функциональная модель бизнес-процесса (рекомендована). 2. Модель потоков данных (обязательна).	IDEF0 DFD
2. Регламентация бизнес-процесса уровня компании	1. Функциональная модель бизнес-процесса (рекомендована). 2. Модель потока работ (обязательна). 3. Модель потоков данных (обязательна).	IDEF0 DEF3 DFD
3. Описание бизнес-процесса уровня структурного подразделения	1. Функциональная модель бизнес-процесса (рекомендована). 2. Модель потока работ (обязательна). 3. Модель потоков данных (обязательна).	IDEF0 DEF3 DFD
4. Разработка регламента выполнения бизнес-процесса структурного подразделения	1. Модель потока работ (обязательна). 2. Модель потоков данных (рекомендована).	DEF3 DFD
5. Описание бизнес-процесса для рабочего места исполнителя	1. Модель потока работ (обязательна).	DEF3
6. Разработка регламента выполнения бизнес-процесса для рабочего места исполнителя	1. Модель потока работ (обязательна).	DEF3

Модель в BPwin рассматривается как совокупность работ, каждая из которых оперирует с некоторым набором данных. Работа изображается в виде прямоугольников, данные — в виде стрелок. Если щелкнуть по любому объекту модели левой кнопкой

мышь, появляется контекстное меню, каждый пункт которого соответствует редактору какого-либо свойства объекта.

Модели AS-IS и TO-BE

Обычно сначала строится модель существующей организации работы — AS-IS (как есть). Анализ функциональной модели позволяет понять, где находятся наиболее слабые места, в чем будут состоять преимущества новых бизнес-процессов и насколько глубоким изменениям подвергнется существующая структура организации бизнеса. Детализация бизнес-процессов позволяет выявить недостатки организации даже там, где функциональность на первый взгляд кажется очевидной. Найденные в модели AS-IS недостатки можно исправить при создании модели TO-BE (как будет) — модели новой организации бизнес-процессов [4].

Технология проектирования ИС подразумевает сначала создание модели AS-IS, ее анализ и улучшение бизнес-процессов, то есть создание модели TO-BE, и только на основе модели TO-BE строится модель данных, прототип и затем окончательный вариант ИС.

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

ГЛАВА I. МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ IDEF0

На начальных этапах создания модели необходимо понять, как работает организация, которую собираются автоматизировать. Руководитель хорошо знает работу в целом, но не в состоянии вникнуть в детали работы каждого рядового сотрудника. Рядовой сотрудник хорошо знает, что творится на его рабочем месте, но может не знать, как работают коллеги. Поэтому для описания работы предприятия необходимо построить модель, которая будет адекватна предметной области и содержать в себе знания всех участников бизнес-процессов организации [5].

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

Процесс моделирования системы в IDEF0 начинается с создания контекстной диаграммы — диаграммы наиболее абстрактного уровня описания системы в целом, содержащей определение субъекта моделирования, цели и точки зрения на модель. Пример контекстной IDEF0-диаграммы показан на рис. 1.2.

Под субъектом понимается сама система, при этом необходимо точно установить, что входит в систему, а что лежит за ее пределами, другими словами, определить, что будет в дальнейшем рассматриваться как компоненты системы, а что как внешнее воздействие. На определение субъекта системы будут существенно влиять позиция, с которой рассматривается система, и цель моделирования — вопросы, на которые построенная модель должна дать ответ. Другими словами, в начале необходимо определить область моделирования. Описание области как системы в целом, так и ее компонентов является основой построения модели. Хотя предполагается, что в ходе моделирования область может корректироваться, она должна быть в основном сформулирована изначально, поскольку именно область определяет направление моделирования.

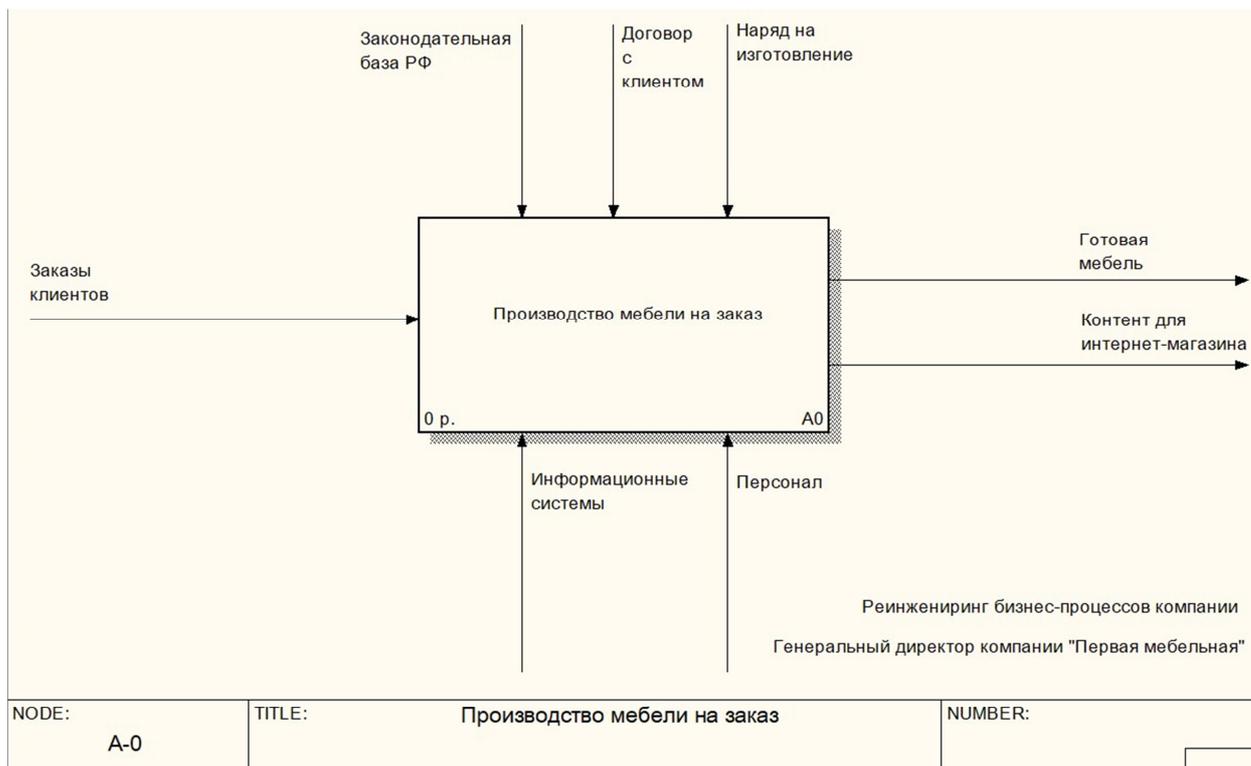


Рис. 1.2. Пример контекстной диаграммы

Цель моделирования

Цель моделирования определяется из ответов на следующие вопросы:

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

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

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

После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется **функциональной декомпозицией**, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются **диаграммами декомпозиции** (рис. 1.3). После декомпозиции контекстной диаграммы проводится

декомпозиция каждого большого фрагмента системы на более мелкие и так далее, до достижения нужного уровня подробности описания. После каждого сеанса декомпозиции проводятся сеансы экспертизы — эксперты предметной области указывают на соответствие реальных бизнес-процессов созданным диаграммам. Найденные несоответствия исправляются, и только после прохождения экспертизы без замечаний можно приступать к следующему сеансу декомпозиции. Так достигается соответствие модели реальным бизнес-процессам на любом уровне модели. Синтаксис описания системы в целом и каждого ее фрагмента одинаков во всей модели.

Работы (Activity) обозначают поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Работы изображаются в виде прямоугольников. Все работы должны быть названы и определены. Имя работы должно быть выражено отглагольным существительным, обозначающим действие (например, "Деятельность компании", "Прием заказа" и т.д.).

Диаграммы декомпозиции содержат родственные работы, т. е. дочерние работы, имеющие общую родительскую работу. Каждая из работ на диаграмме декомпозиции может быть в свою очередь декомпозирована.

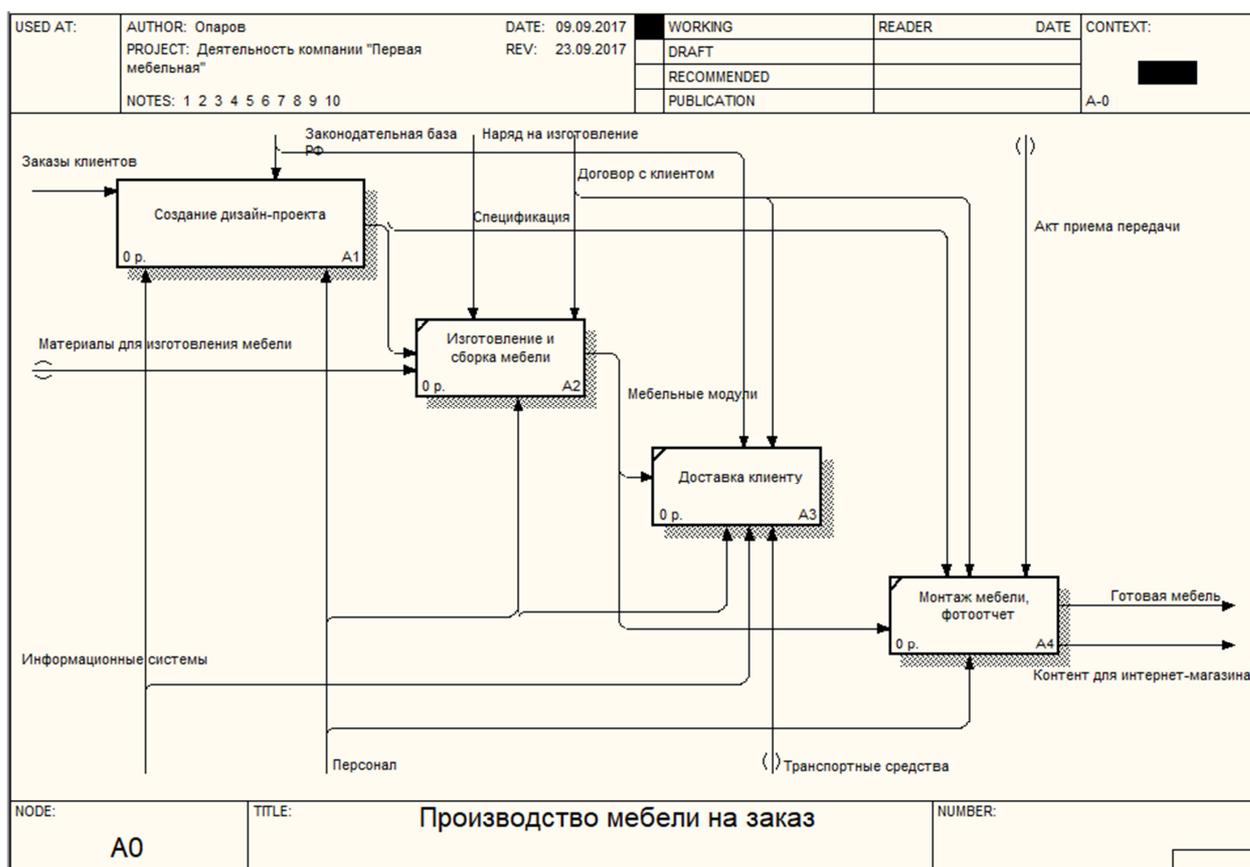
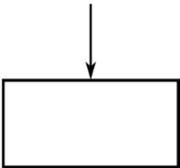
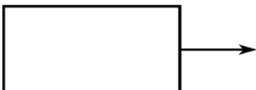
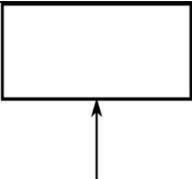
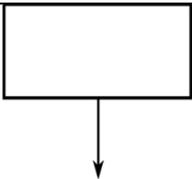


Рис. 1.3. Пример диаграммы декомпозиции работы «Производство мебели на заказ»

Стрелки (Arrow) описывают взаимодействие работ и представляют собой некую информацию (**данные**), выраженную существительными, например, "Звонки клиентов", "Правила и процедуры", "Бухгалтерская система".

В IDEF0 различают пять типов стрелок. Каждый тип стрелок подходит к определенной стороне прямоугольника, изображающего работу, или выходит из нее (табл. 1.2).

Таблица 1.2. Названия и описания стрелок в IDEF0

Название стрелки	Обозначение	Изображение на диаграммах IDEF0	Примечание
Вход (Input)	материал или информация, которые используются или преобразуются работой для получения результата (выхода).		Допускается, что работа может не иметь ни одной стрелки входа
Управление (Control)	правила, стратегии, процедуры или стандарты, которыми руководствуется работа.		Каждая работа должна иметь хотя бы одну стрелку управления. Допускается, что работа может не иметь ни одной стрелки входа.
Выход (Output)	материал или информация, которые производятся работой. Каждая работа должна иметь хотя бы одну стрелку выхода.		Каждая работа должна иметь хотя бы одну стрелку выхода. Работа без результата не имеет смысла и не должна моделироваться.
Механизм (Mechanism)	ресурсы, которые выполняют работу, например, персонал предприятия, станки, устройства и т. д.		Могут не изображаться в модели.
Вызов (Call)	Стрелка вызова рисуется как исходящая из нижней грани работы.		используются в механизме слияния и разделения моделей.

Граничные стрелки на контекстной диаграмме служат для описания взаимодействия системы с окружающим миром. Они могут начинаться у границы

диаграммы и заканчиваться у работы, или наоборот. Такие стрелки называются граничными.

Внутренние стрелки используются для связи работ между собой. Они не касаются границ диаграммы, а начинаются у одной работы и заканчиваются у другой. Для изображения внутренней стрелки необходимо щелкнуть сначала по нужному сегменту одной работы, затем по нужному сегменту другой. В IDEF0 выделяется пять типов связей работ:

1. **Связь по входу (output-input)** – стрелка выхода вышестоящей работы направляется на вход нижестоящей.

2. **Связь по управлению (output-control)** – выход вышестоящей работы направляется на управление нижестоящей. Данная связь указывает на доминирование вышестоящей работы. Данные или объекты выхода вышестоящей работы не меняются в нижестоящей.

3. **Обратная связь по входу (output-input feedback)** – выход нижестоящей работы направляется на вход вышестоящей. Данный вид связи применяется для описания циклов.

4. **Обратная связь по управлению (output-control feedback)** – выход нижестоящей работы направляется на управление вышестоящей. Данный вид связи часто свидетельствует об эффективности бизнес-процесса.

5. **Связь выход-механизм (output-mechanism)** – выход одной работы направляется на механизм другой. Используется эта связь реже, чем прочие, и показывает, что одна работа подготавливает ресурсы, необходимые для другой работы.

Помимо явных стрелок, использующих одну единственную работу как источник и как назначение, существуют и другие стрелки. Разветвляющиеся и сливающиеся стрелки могут иметь в качестве источника или назначения несколько работ. Граничные стрелки, внесенные на диаграмму декомпозиции нижнего уровня, отображаются в квадратных скобках и не показываются на диаграммах верхних уровней. Их можно туннелировать или поднимать до уровня граничных стрелок. Туннельные стрелки изображаются с круглыми скобками на конце. Для туннелирования стрелок необходимо вызвать диалог **Border Arrow Editor**. Если щелкнуть по кнопке **Resolve Border Arrow**, стрелка мигрирует на диаграмму верхнего уровня, если по кнопке **Change To Tunnel** — стрелка будет туннелирована и не попадет на другую диаграмму. Туннелирование может быть применено для изображения малозначимых стрелок. Если на какой-либо диаграмме нижнего уровня необходимо изобразить малозначимые данные или объекты, которые не обрабатываются или не используются работами на текущем уровне, то их необходимо направить на вышестоящий уровень (на родительскую диаграмму). Если эти данные не используются на родительской диаграмме, их нужно направить еще выше, и т. д. В

результате малозначимая стрелка будет изображена на всех уровнях и затруднит чтение всех диаграмм, на которых она присутствует. Выходом является туннелирование стрелки на самом нижнем уровне. Такое туннелирование называется "не-в-родительской-диаграмме".

Другим примером туннелирования может быть ситуация, когда стрелка механизма мигрирует с верхнего уровня на нижний, причем на нижнем уровне этот механизм используется одинаково во всех работах без исключения. (Предполагается, что не нужно детализировать стрелку механизма, т. е. стрелка механизма на дочерней работе именована до разветвления, а после разветвления ветви не имеют собственного имени). В этом случае стрелка механизма на нижнем уровне может быть удалена, после чего на родительской диаграмме она может быть туннелирована, а в комментарии к стрелке или в словаре можно указать, что механизм будет использоваться во всех работах дочерней диаграммы декомпозиции. Такое туннелирование называется "не-в-дочерней-работе".

Несвязанные граничные стрелки (*unconnected border arrow*). При декомпозиции работы, входящие в нее и исходящие из нее стрелки (кроме стрелки вызова) автоматически появляются на диаграмме декомпозиции (миграция стрелок), но при этом не касаются работ. Такие стрелки называются несвязанными и воспринимаются в VPwin как синтаксическая ошибка. Для связывания стрелок входа, управления или механизма необходимо перейти в режим редактирования стрелок, щелкнуть по наконечнику стрелки и потом по соответствующему сегменту работы. Для связывания стрелки выхода необходимо перейти в режим редактирования стрелок, щелкнуть по сегменту выхода работы и затем по стрелке.

Слияние и расщепление моделей

Возможность слияния и расщепления моделей обеспечивает коллективную работу над проектом. Так, руководитель проекта может создать декомпозицию верхнего уровня и дать задание аналитикам продолжить декомпозицию каждой ветви дерева в виде отдельных моделей. После окончания работы над отдельными ветвями все подмодели могут быть слиты в единую модель. С другой стороны, отдельная ветвь модели может быть отщеплена для использования в качестве независимой модели, для доработки или архивирования.

VPwin использует для слияния и разветвления моделей стрелки вызова. Для слияния необходимо выполнить следующие условия:

- ✓ обе сливаемые модели должны быть открыты в VPwin.
- ✓ имя модели-источника, которое присоединяют к модели-цели, должно совпадать с именем стрелки вызова работы в модели-цели.

- ✓ стрелка вызова должна исходить из недекомпозируемой работы (работа должна иметь диагональную черту в левом верхнем углу).
- ✓ имена контекстной работы подсоединяемой модели-источника и работы на модели-цели, к которой мы подсоединяем модель-источник, должны совпадать.
- ✓ модель-источник должна иметь, по крайней мере, одну диаграмму декомпозиции.

Для слияния моделей нужно щелкнуть правой кнопкой мыши по работе со стрелкой вызова в модели-цели и во всплывающем меню выбрать пункт *Merge Model*.

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

ВРwin вносит ICOM-коды автоматически. Для отображения ICOM-кодов следует включить опцию ICOM codes на закладке *Display* диалога *Model Properties* (меню *Model/Model Properties*).

Каркас диаграммы содержит оформление листа диаграммы, заголовок (верхняя часть рамки) и подвал (нижняя часть рамки). Заголовок каркаса используется для отслеживания состояния диаграммы в процессе создания модели. Нижняя часть используется для идентификации и позиционирования в иерархии диаграмм. В табл. 1.3. приведено значение элементов каркаса. Значения полей каркаса задаются в диалоге *Diagram Properties* (меню *Diagram /Diagram Properties*).

Таблица 1.3. Поля каркаса диаграммы и их описание

Поле	Смысл
<i>Used At</i>	Используется для указания на родительскую работу в случае, если на текущую диаграмму ссылались посредством стрелки вызова
<i>Autor, Date, Rev, Project</i>	Имя создателя диаграммы, дата создания и имя проекта, в рамках которого была создана диаграмма. REV-дата последнего редактирования диаграммы
<i>Notes 123456789 10</i>	Используется при проведении сеанса экспертизы. Эксперт должен (на бумажной копии диаграммы) указать число замечаний, вычеркивая цифру из списка каждый раз при внесении нового замечания
<i>Status</i>	Статус отображает стадию создания диаграммы, отображая все этапы публикации

Working	Новая диаграмма, кардинально обновленная диаграмма или новый автор диаграммы
Draft	Диаграмма прошла первичную экспертизу и готова к дальнейшему обсуждению
Recommended	Диаграмма и все ее сопровождающие документы прошли экспертизу. Новых изменений не ожидается
Publication	Диаграмма готова к окончательной печати и публикации
Reader	Имя читателя (эксперта)
Date	Дата прочтения (экспертизы)
Context	Схема расположения работ в диаграмме верхнего уровня. Работа, являющаяся родительской, показана темным прямоугольником, остальные – светлым. На контекстной диаграмме (A-0) показана надпись TOP. В левом нижнем углу показывается номер по узлу родительской диаграммы.
Node	Номер узла диаграммы (номер родительской работы)
Title	Имя диаграммы. По умолчанию — имя родительской работы
Number C-Number	уникальный номер версии диаграммы
Page	Номер страницы, может использоваться как номер страницы при формировании папки

Диаграмма дерева узлов (Node Tree) показывает иерархию работ в модели и позволяет рассмотреть всю модель целиком, но не показывает взаимосвязи между работами (рис. 1.4). Диаграмм дерева узлов может быть в модели сколько угодно много, поскольку дерево может быть построено на произвольную глубину и не обязательно с корня.

Диаграммы "только для экспозиции" (FEO) часто используются в модели для иллюстрации других точек зрения, для отображения отдельных деталей, которые не поддерживаются явно синтаксисом IDEF0. Диаграммы FEO позволяют нарушить любое синтаксическое правило, поскольку, по сути, являются просто картинками — копиями стандартных диаграмм и не включаются в анализ синтаксиса.

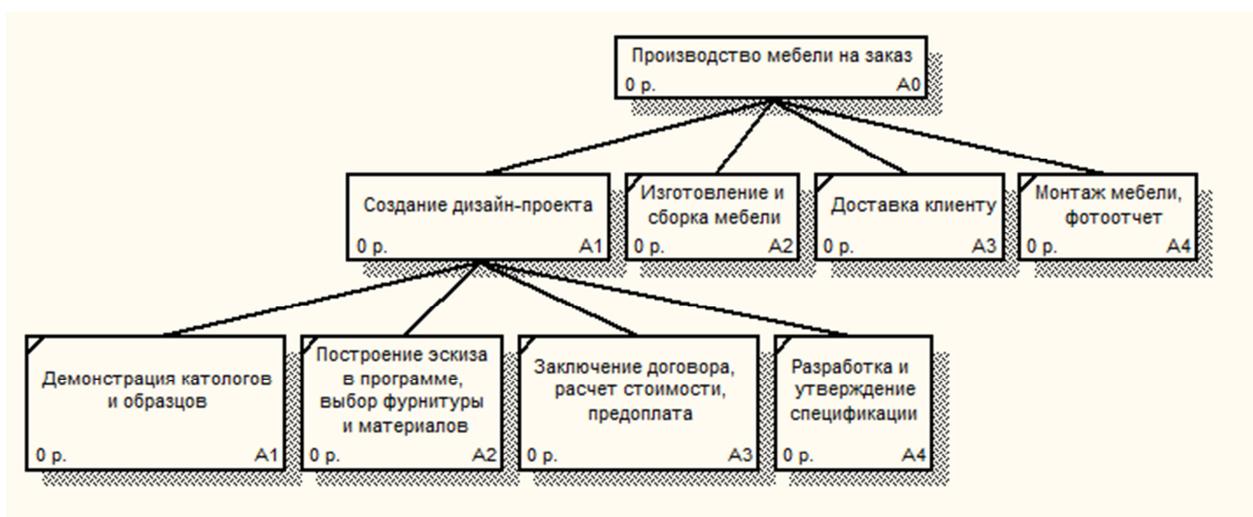


Рис. 1.4. Диаграмма дерева узлов

Общие правила формирования моделей бизнес-процессов в IDEF0

1. Функциональная модель бизнес-процессов (IDEF0) представляет бизнес-процесс как совокупность выполняемых функций (направлений деятельности). Для определяемого в каждом конкретном случае уровня детализации модели, функции должны рассматриваться уже как операции, выполняемые в ходе бизнес-процесса. На нижнем уровне модели в качестве функций рассматриваются отдельные операции.

2. Модель IDEF0 рекомендована для применения в компании при описании бизнес-процессов верхнего уровня.

3. При составлении функциональной модели бизнес-процесса (IDEF0) описываются выполняемые функции и входные, выходные потоки материальных, финансовых ресурсов и информации.

4. При описании бизнес-процесса одновременно могут применяться наборы моделей в различных нотациях (IDEF0, IDEF3, DFD). Диаграмма декомпозиции IDEF0 может быть диаграммой IDEF3 или DFD.

5. Стрелки на диаграмме IDEF0 связывают выполняемые функции.

6. Стрелки-входы обозначают материальные, финансовые и информационные ресурсы, преобразуемые функцией (блоком).

7. Стрелки-выходы обозначают материальные, финансовые и информационные ресурсы, являющиеся результатом выполнения функции (блока). Каждая функция должна иметь хотя бы одну стрелку-выход.

8. Стрелки-управления обозначают правила. Стандарты, указания, нормативные акты и т.п. в соответствии с которыми выполняется функция. Каждая функция должна иметь хотя бы одну стрелку-управления.

9. Стрелки-механизмы обозначают средства выполнения функций: персонал, устройства, оборудование, станки, программное обеспечение и т.п.

10. Стрелки изображаются вертикальными и горизонтальными отрезками прямых с наконечником с одной стороны, пересекающихся под прямым углом и сопряженных дугами.

11. Стрелки соединяются с блоком следующим образом: 1) концы стрелок должны касаться внешней стороны блока, но не пересекать ее; 2) стрелки должны присоединяться к блоку на его сторонах, присоединения в углах не допускается.

12. При изображении стрелок допускается их слияние и разветвление.

13. Именованье стрелок и создание меток при разветвлении подчиняется следующим правилам: а) если стрелка именована до разветвления, а после разветвления

ни одна из ветвей не именована, то подразумевается, что каждая ветвь моделирует те же данные или объекты, что и ветвь до разветвления; б) если стрелка именована до разветвления, а после разветвления какая-либо из ветвей не именована, то подразумевается, что она моделирует те же данные или объекты, что и до разветвления; в) недопустима ситуация, когда стрелка до разветвления не именована, а после не именована какая-либо из ее ветвей.

14. Правила сливающихся стрелок аналогичны правилам разветвляющихся стрелок.

15. В рамках одной диаграммы существуют шесть типов отношений между блоками: 1) доминирование; 2) управление; 3) выход-вход; 4) обратная связь по входу; 5) связь выход-механизм.

16. Блоки, расположенные на диаграмме выше и левее, «доминируют» над блоками, расположенными ниже и правее.

17. Отношение управления возникает тогда, когда выход одного блока служит управляющим воздействием на блок с меньшим доминированием.

18. Отношение «выход-вход» возникает при соединении выхода одного блока с входом другого блока с меньшим доминированием.

19. Отношение обратной связи по входу возникает тогда, когда выход блока становится входом другого блока с большим доминированием.

20. При построении модели бизнес-процесса в IDEF0 используется принцип декомпозиции. Декомпозиция функций производится для более подробного описания выбранной для декомпозиции функции. При декомпозиции функция раскладывается на множество функций, выполнение которых полностью обеспечивает реализацию декомпозированной функции.

21. Диаграмма, представляющая собой результат декомпозиции, называется дочерней диаграммой. Декомпозируемая диаграмма называется родительской диаграммой.

22. Функциональная модель IDEF0 представляется в виде совокупности иерархически упорядоченных диаграмм. Выполнение функции, отображенной на диаграмме верхнего уровня, детализируется на диаграммах нижнего уровня.

23. Моделирование бизнес-процесса в IDEF0 начинается с построения основной (контекстной) диаграммы, которая представляет собой самое общее описание системы и ее взаимодействие с внешней средой. На контекстной диаграмме должна быть представлена цель моделирования и точка зрения, соответствующая цели моделирования.

24. При построении моделей в IDEF0 используются так называемые туннельные стрелки. Туннельные стрелки обозначаются как круглые скобки на конце или в начале

стрелок. Допускается использование квадратных скобок вместо круглых. Стрелка, помещенная в «туннель» там, где она присоединяется к блоку, означает, что данные, выраженные этой стрелкой, не обязательны на следующем уровне декомпозиции. Стрелка, помещаемая в туннель на свободном конце, означает, что выраженные ею данные отсутствуют на родительской диаграмме.

25. Все граничные стрелки на дочерней диаграмме, за исключением стрелок, помещенных в туннель должны соответствовать стрелкам родительского блока.

Правила при составлении моделей IDEF0 [2]:

- ✓ В составе диаграмм должна присутствовать контекстная диаграмма;
- ✓ Блоки на диаграмме должны располагаться по диагонали – от левого верхнего угла диаграммы до правого нижнего в порядке присвоенных номеров.
- ✓ Диаграммы (кроме контекстной) должны содержать не менее 3-х и не более 8-ми блоков;
- ✓ Каждый блок диаграммы получает номер. Помещаемый в правом нижнем углу; порядок нумерации – от верхнего левого к нижнему правому блоку;
- ✓ Каждый блок, не имеющий декомпозиции, помечается небольшой диагональной чертой, расположенной в верхнем углу блока;
- ✓ Имена блоков должны быть уникальными;
- ✓ Стрелки сливаются, если они представляют сходные данные и их источник не указан на диаграмме;
- ✓ Обратные связи по управлению рисуются «верхней петлей». Обратные связи по входу – «нижней петлей»;
- ✓ Стрелки объединяются, если они имеют общий источник либо приемник или если они представляют связанные данные;
- ✓ При соединении большого числа блоков необходимо избегать необязательных пересечений стрелок. Следует минимизировать число петель и поворотов каждой стрелки. При описании функций, преобразующих информационные потоки, на диаграммах нижних уровней названиям стрелок-входов должен соответствовать конкретный документ или перечень документов.
- ✓ Построение стрелок-выходов подчиняется тем же правилам, что и стрелок-входов;
- ✓ Все стрелки-механизмы на диаграммах нижнего уровня должны иметь в своем названии точное название отдела, выполняющего данную функцию.
- ✓ Стрелки управления на диаграммах нижнего уровня должны быть детализованы до названия документа, регламентирующего данное действие;

✓ На диаграммах верхнего уровня разрешается использовать названия групп документов только в том случае, если они раскрываются до названия регламентирующего документа на нижних уровнях декомпозиции. Все прочие условия (кроме регламентирующих документов) должны быть показаны на диаграмме как обычные входы, а не как стрелки управления.

Основываясь на материале, представленном в теоретическом блоке и учитывая правила построения моделей IDEF0, выполните лабораторные работы № 1 и 2.

ЛАБОРАТОРНАЯ РАБОТА № 1

Цель работы: получение навыков применения методологии IDEF0 и программы Ramus Educational для разработки контекстных диаграмм и диаграмм декомпозиции.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Долгое время типичным представителем инструментальных средств функционального моделирования являлся пакет AllFusion Process Modeler. Однако выпуск этого мощного case-средства в 2011 был прекращен и в настоящее время не поддерживается компанией производителем. Для изучения методологии функционального моделирования IDEF0 используется бесплатный обучающий продукт Ramus Educational.

Ramus Educational имеет достаточно простой и интуитивно понятный интерфейс пользователя. При запуске Ramus Educational появляется диалоговое окно, в котором можно создать новый проект, либо открыть уже имеющийся. Для создания нового проекта необходимо выбрать соответствующий переключатель и нажать кнопку «Далее». На экране появиться Мастер «Свойства проекта», где можно выбрать нотацию моделирования и задать свойства проекта. В окне справа появляется описание выбранного окна. Программа Ramus Educational поддерживает две методологии - IDEF0 и DFD, каждая из которых решает свои специфические задачи (рис. 1.5).

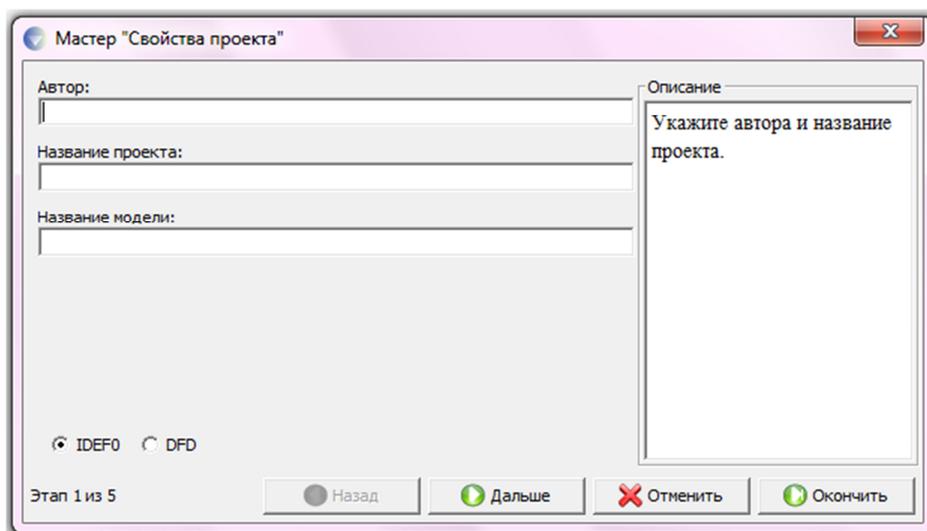


Рис. 1.5. Диалоговое окно мастера «Свойства проекта»

Состав палитры инструментов изменяется автоматически, когда происходит переключение с одной нотации на другую. При дальнейшей настройке свойств проекта нужно указать организацию, для которой разрабатывается данная модель и краткое описание модели. После описания свойств модели можно описать ресурсы, задействованных в проекте. В программе Ramus Educational они называются словом Классификатор (рис. 1.6). Последний шаг создания модели позволяет выбрать собственников проекта из созданных классификаторов. Как правило, собственником проекта считается лицо, ответственное за внедрение и разработку этого проекта в бизнес-процессы фирмы. После нажатия кнопки окончить появляется рабочая область проекта.

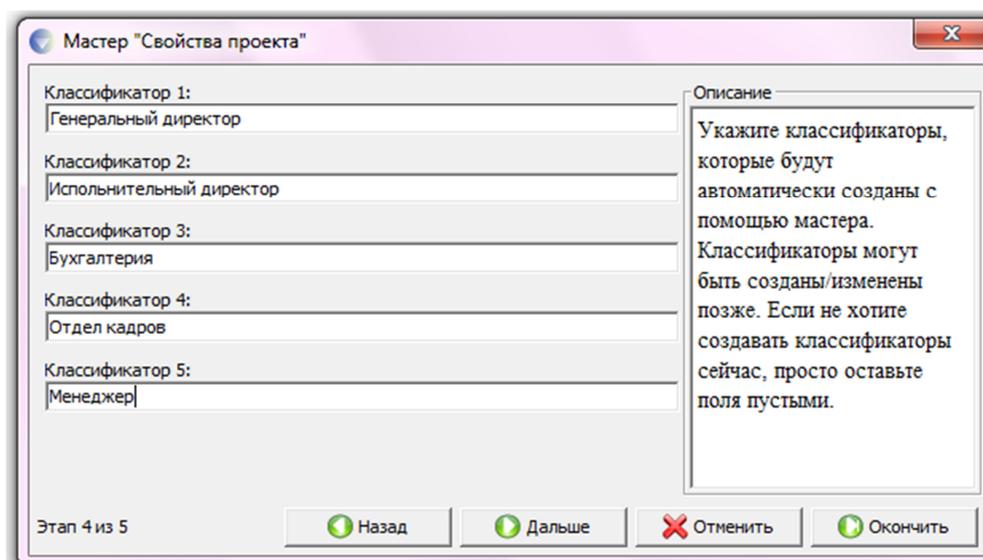


Рис. 1.6. Диалоговое окно для описания классификаторов

Модель в Ramus Educational рассматривается как совокупность работ, каждая из которых оперирует с некоторым набором данных. Работа изображается в виде прямоугольников, данные — в виде стрелок. Если щелкнуть по любому объекту модели

левой кнопкой мыши, появляется контекстное меню, каждый пункт которого соответствует редактору какого-либо свойства объекта.

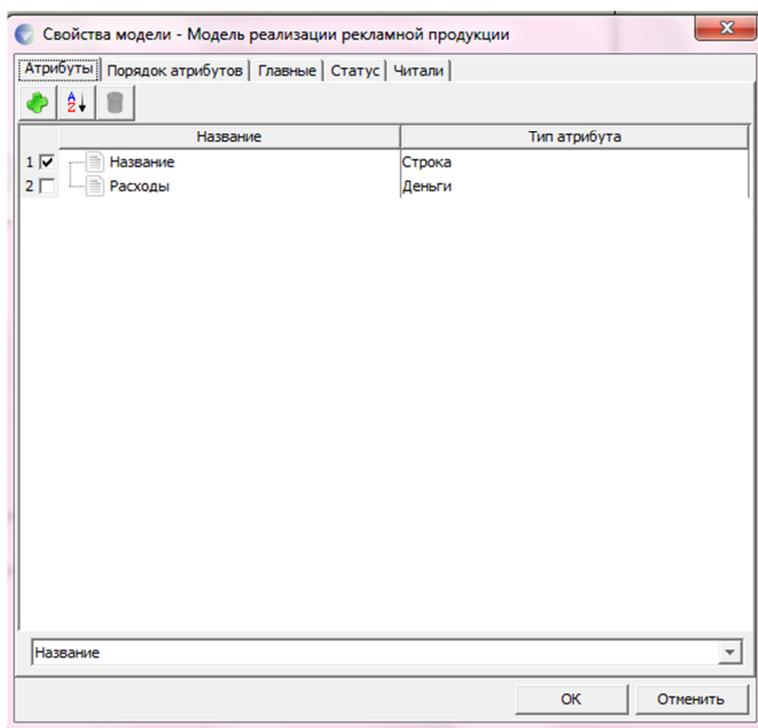


Рис. 1.7. Диалог задания свойств модели

Для внесения имени работы следует щелкнуть по работе правой кнопкой мыши, выбрать в меню **Редактировать активный элемент** и в появившемся диалоге внести имя работы. Для описания других свойств работы служит диалог Параметры стрелки (рис. 1.8).

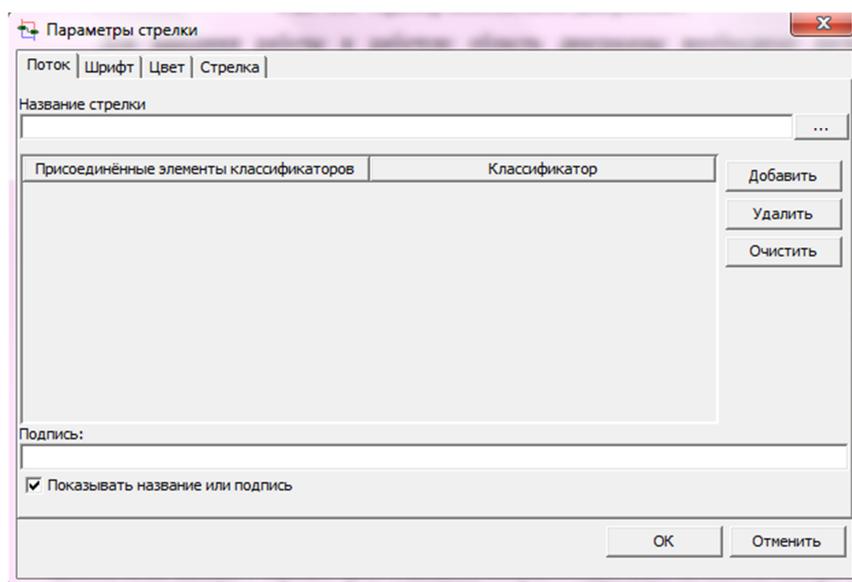


Рис. 1.8. Диалоговое окно Параметры стрелки

Диаграммы декомпозиции содержат родственные работы, т.е. дочерние работы, имеющие общую родительскую работу. Для создания диаграммы декомпозиции следует щелкнуть по кнопке с черным треугольником направленным вниз на панели инструментов. При этом возникает диалог Создания новой диаграммы (рис. 1.9), в котором следует указать нотацию новой диаграммы и количество работ на ней. Остановимся пока на нотации IDEF0 и щелкнем на ОК. Появляется диаграмма декомпозиции (рис.1.9).

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

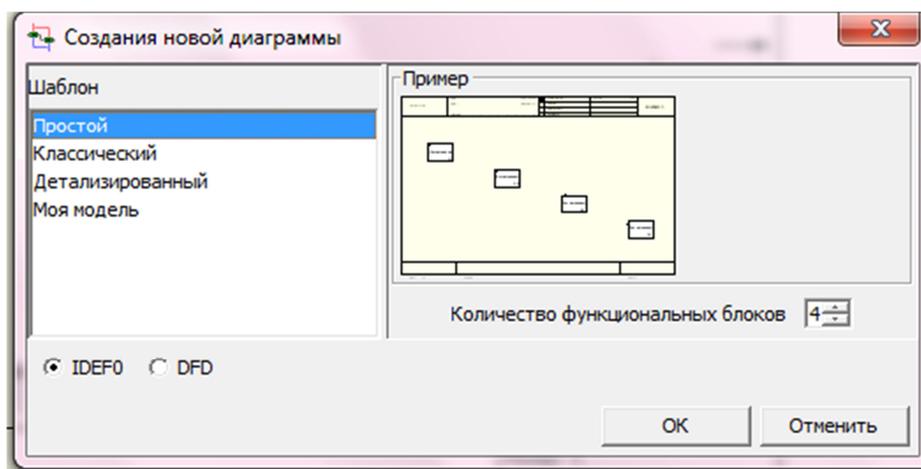


Рис. 1.9. Диалог создания новой диаграммы

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

Такой порядок называется порядком доминирования. Согласно этому принципу расположения в левом верхнем углу помещается самая важная работа или работа, выполняемая по времени первой. Далее вправо вниз располагаются менее важные или выполняемые позже работы. Такое размещение облегчает чтение диаграмм, кроме того, на нем основывается понятие взаимосвязей работ (см. теорию Главы I).

Каждая из работ на диаграмме декомпозиции может быть в свою очередь декомпозирована. На диаграмме декомпозиции работы нумеруются автоматически слева направо. Номер работы показывается в правом нижнем углу. В левом верхнем углу изображается небольшая диагональная черта, которая показывает, что данная работа не была декомпозирована. Так, на рис. 1.3 все работы еще не были декомпозированы.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание № 1. Разработать контекстную диаграмму

1. Запустите программу Ramus Educational;
2. В появившемся диалоговом окне введите имя модели (*Name*) – «Производство мебели на заказ» (рис. 1.2). Укажите тип модели – IDEF0.
3. Для создания новой модели в Ramus нужно пройти 5 шагов. На шаге 1 задается имя модели. Создайте новую модель и задайте ей свойства, указанные на рис. 1.10.
4. Задайте параметры следующих шагов работы мастера:
 - Шаг 2: Используется в компании «Первая мебельная»
 - Шаг 3. Описание «Бизнес-процессы производства корпусной мебели под заказ».
 Описание производится вручную в текстовой области в соответствии с данными таблицы 1.4.

Таблица 1.4. Значение параметров модели

Название вкладки	Значение
<i>General</i> (общие)	AS-IS («Как есть»).
<i>Purpose</i> (цель модели)	Реинжиниринг бизнес-процессов компании
<i>View Point</i> (точка зрения)	Генеральный директор компании «Первая мебельная»
<i>Definition</i> (описание)	Бизнес-процессы производства корпусной мебели под заказ
<i>Scope</i> (границы модели)	Масштаб предприятия
<i>Source</i> (источник)	Данные предпроектного анализа

Шаг 4. Классификатор 1: Генеральный директор.

Шаг 5. Добавить к собственникам проекта Генерального директора.

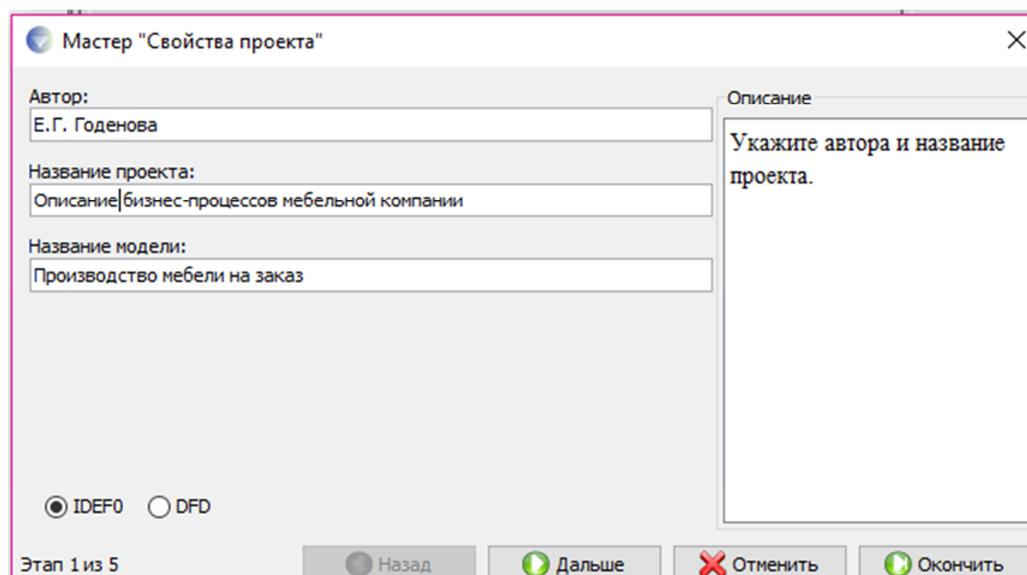


Рис. 1.10. Диалог мастера «Свойства проекта»

5. Создайте контекстную диаграмму, по образцу на рис. 1.2.

6. Создайте стрелки, с параметрами, указанными в табл. 1.6. Для создания стрелки необходимо нажать кнопку . При нажатой кнопке, навести курсор на граничную область до появления черного прямоугольника, выполнить щелчок мышью. Затем аналогично навести курсор на нужный сегмент работы до появления черного треугольника. Все остальные стрелки рисуются аналогично. Направление определяется назначением стрелки. При расположении стрелок учитывайте общие правила формирования моделей бизнес-процессов в IDEF0, рассмотренные в теоретическом материале Главы 1.

Таблица 1.6. Название и описание стрелок, взаимодействующих с работой A0

<i>Arrow Name</i> (Имя стрелки)	<i>Definition</i> (Описание)	<i>Arrow Type</i> (Тип стрелки)
Заказы клиентов	Информация для формирования спецификации	Input (I1)
Законодательная база РФ	Законы, регламентирующие деятельность компании и защиту прав потребителей	Control (C1)
Договор с клиентом	Информация о сроках выполнения заказа, стоимости, особенностях доставки	Control (C2)
Наряд на изготовление	Данные для заготовки деталей корпусной мебели, полученные из спецификации	Control (C3)
Информационные системы	Обработка заказов, планирование производственных заказов, составлении нарядов, учет затрат и результатов	Mechanism (M1)
Персонал	Производственный и вспомогательный персонал	Mechanism (M2)
Готовая мебель	Мебель, собранная в соответствии со спецификацией и договором	Output (O1)
Контент для интернет-магазина	Фото заказа для размещения в каталоге интернет-магазина	Output (O2)

Задание № 2. Выполнить функциональную декомпозицию работы A0

1. Наведите курсор мыши на контекстную диаграмму A0.
2. Перейдите к дочерней диаграмме нажав кнопку  (*Go to Child Diagram*) панели инструментов.
3. В появившемся окне укажите тип диаграммы – IDEF0, число работ – 4.
4. Внесите имена и описания созданных работ в диалоговом окне в соответствии с таблицей 1.7

Таблица 1.7. Данные для диаграммы декомпозиции работы A0

<i>Имя работы (Activity Name)</i>	<i>Определение (Definition)</i>
Создание дизайн-проекта	Демонстрация образцов, консультация клиента, учет пожеланий, определение параметров мебели, заключение договора и составление спецификации, предоплата
Изготовление и сборка мебели	Производственный процесс. Наряд на изготовление производственному цеху. Изготовление деталей из пиломатериалов в соответствии со спецификацией. Сборка мелких деталей до мебельных блоков
Доставка клиенту	Оплата заказа и доставки. Логистика доставки мебели клиентам. Обсуждение с клиентом сроков и времени. Отгрузка со склада. Подъем на этаж заказчика.
Монтаж мебели, фотоотчет	Работа бригады монтажников. Подписание акта приема-передачи. Оплата доставки. Фото для разработки каталога

5. Расположите граничные стрелки согласно рис. 1.3. При расположении стрелок учитывайте общие правила формирования моделей бизнес-процессов в IDEF0, рассмотренные в теоретическом материале Главы 1. Добавить граничные стрелки и выполнить их туннелирование:

- ✓ Стрелка входа «**Материалы для изготовления мебели**» для работы A2;
- ✓ Стрелка-механизм «**Транспортные средства**» для работы A3;
- ✓ Стрелка управления «**Акт приема-передачи**» для работы A4;

6. Создать внутренние стрелки:

Стрелка «**Спецификация**» - выход работы A1, вход для работы A2 и управление для работы A4;

Стрелка «**Мебельные модули**» – выход работы A2, вход работы A3, вход работы A4;

7. Сохраните изменения в модели в исходном файле.

Задание № 3. Создать диаграмму декомпозиции работы A1 «Дизайн проекта»

1. Для работы A1 «Дизайн проекта» создать диаграмму следующего уровня декомпозиции (рис. 1.11). Тип диаграммы – IDEF0, число работ – 4. Перечень работ приведен в таблице 1.8. Добавьте стрелки на диаграмму декомпозиции A1 в соответствии с таблицей 1.9.

Таблица 1.8. Данные для диаграммы декомпозиции работы А1

Имя работы (Activity Name)	Определение (Definition)
Демонстрация каталогов и образцов	Этап демонстрации ассортимента, маркетинг, выяснение пожеланий заказчика
Построение эскиза в программе, выбор фурнитуры и материалов	Разработка эскиза в соответствии с заданными размерами и параметрами, демонстрация каталогов фурнитуры поставщиков
Заключение договора, расчет стоимости, предоплата	Внесение данных клиента в договор, подписание договора, расчет окончательной стоимости работ по договору
Разработка и утверждение спецификации	Разработка спецификации на основе эскиза и выбранных материалов

Таблица 1.9. Данные для стрелок на диаграмме декомпозиции А1

Имя стрелки (Arrow Name)	Источник стрелки (Arrow Source)	Тип источника стрелки (Arrow Source Type)	Назначение стрелки (Arrow Destination)	Тип назначения стрелки (Arrow Destination Type)
Заказы клиентов	Граница	Input	Демонстрация каталогов и образцов	Input
Книга отзывов и предложение	Tunnel	Control	Демонстрация каталогов и образцов	Control
			Построение эскиза в программе, выбор фурнитуры и материалов	Control
Сформированный заказ	Демонстрация каталогов и образцов	Output	Построение эскиза в программе, выбор фурнитуры и материалов	Input
Эскиз	Построение эскиза в программе, выбор фурнитуры и материалов	Output	Заключение договора, расчет стоимости, предоплата	Input
Проект договора	Подготовка договора	Output	Разработка и утверждение спецификации	Input
Спецификация	Разработка и утверждение спецификации	Output	Граница	Output
Законодательная	Граница	Control	Заключение договора, расчет	Control

Имя стрелки (<i>Arrow Name</i>)	Источник стрелки (<i>Arrow Source</i>)	Тип источника стрелки (<i>Arrow Source Type</i>)	Назначение стрелки (<i>Arrow Destination</i>)	Тип назначения стрелки (<i>Arrow Destination Type</i>)
база РФ			стоимости, предоплата	
			Разработка и утверждение спецификации	
Выставочные образцы, каталоги	Tunnel	Mechanism	Демонстрация каталогов и образцов	Mechanism
			Построение эскиза в программе, выбор фурнитуры и материалов	
Персонал	Граница	Mechanism	Демонстрация каталогов и образцов	Mechanism
			Построение эскиза в программе, выбор фурнитуры и материалов	
			Заключение договора, расчет стоимости, предоплата	
			Разработка и утверждение спецификации	
Информационные системы	Граница	Mechanism	Построение эскиза в программе, выбор фурнитуры и материалов	Mechanism

Задание № 4. Выполнить разветвление стрелок механизмов.

1. Разветвите стрелку механизма **Персонал** на всех диаграммах декомпозиции и соедините с выполняемыми работами: грузчик, сборщик, плотник, замерщик, консультант, менеджер.
2. Сохранить изменения в исходный файл.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

- 1) Перечислите несколько нотаций для описания бизнес-процессов;
- 2) Что понимается под термином «CASE-средства»;
- 3) Что представляет собой IDEF0-модель?
- 4) Что понимается под термином «декомпозиция»?
- 5) Каков синтаксис описания систем в IDEF0?
- 6) Что такое работа в терминологии IDEF0?

- 7) Сколько блоков декомпозиции рекомендуется создавать при моделировании и почему?
- 8) Какое смысловое значение имеет расположение работ на диаграммах декомпозиции от верхнего левого угла к правому нижнему?
- 9) Какова функция стрелок на диаграммах IDEF0?
- 10) Назовите назначение ICOM-кодов?
- 11) Наличие какой связи свидетельствует о высокой эффективности бизнес-процесса?
- 12) Какие виды связи могут быть реализованы в нотации IDEF0?
- 13) Объяснить смысл стрелок с квадратными и круглыми скобками на диаграмме IDEF0?
- 14) Может ли диаграмма не иметь стрелок входа, выхода, управления и ресурсов соответственно?

ЛАБОРАТОРНАЯ РАБОТА № 2

Разработка IDEF0-модели «Функционирование службы информации»

Цель работы: закрепление навыков применения методологии IDEF0 и программы Ramus Educational для формирования моделей бизнес-процессов, получение базовых представлений о функционально-стоимостном анализе

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Как уже упоминалось выше, обычно сначала строится функциональная модель существующей в организации работы — AS-IS (как есть). После построения модели AS-IS проводится анализ бизнес-процессов, потоки данных и объектов перенаправляются и улучшаются, в результате строится модель TO-BE. Как правило, строится несколько моделей TO-BE, из которых по какому-либо критерию выбирается наилучшая. Проблема состоит в том, что таких критериев много и непросто определить важнейший. Для того чтобы определить качество созданной модели с точки зрения эффективности бизнес-процессов, необходима система метрики, т.е. качество следует оценивать количественно [4].

Функционально-стоимостный анализ (ФСА) – методология непрерывного совершенствования продукции, производственных технологий, организационных структур. Задачей ФСА является снижение всех видов затрат при одновременном сохранении или повышении качества. Функционально-стоимостный подход к рассмотрению объекта обоснован тем, что потребителя интересует не объект сам по себе, а его функции, качество их выполнения и затраты на приобретение этого качества.

Основным критерием совершенства (конкурентоспособности) объекта с позиции ФСА является его потребительская стоимость, определяемая соотношением качества (полезности) объекта и затрат потребителя [5]. В частности, с помощью ФСА можно решать задачи:

- ✓ анализа затрат (выявление зон неоправданно высоких затрат на всем жизненном цикле объекта);
- ✓ оценки решений (программное обеспечение для количественной оценки новых идей и проектов);
- ✓ оценки конкурентоспособности (определение конкурентоспособной цены и т.д.) [5].

При построении модели средствами пакета VPwin аналитику предоставляется два инструмента для оценки модели — стоимостный анализ, основанный на работах - ABC (*Activity Based Costing*) и свойства, определяемые пользователем - UDP (*User Defined Properties*). *Функциональное оценивание* – ABC – это технология выявления и исследования стоимости выполнения той или иной функции (действия). Исходными данными для функционального оценивания являются затраты на ресурсы (материалы, персонал и т.д.). В сравнении с традиционными способами оценки затрат, при применении которых часто недооценивается продукция, производимая в незначительном объеме, и переоценивается массовый выпуск, ABC обеспечивает более точный метод расчета стоимости производства продукции, основанный на стоимости выполнения всех технологических операций, выполняемых при ее выпуске. *Стоимостный анализ представляет собой соглашение об учете, используемое для сбора затрат, связанных с работами, с целью определить общую стоимость процесса.* Стоимостный анализ основан на модели работ, потому что количественная оценка невозможна без детального понимания функциональности предприятия. Обычно ABC применяется для того, чтобы понять происхождение выходных затрат и облегчить выбор нужной модели работ при реорганизации деятельности предприятия (Business Process Reengineering, BPR). С помощью стоимостного анализа можно решить такие задачи, как определение действительной стоимости производства продукта, определение действительной стоимости поддержки клиента, идентификация наиболее дорогостоящих работ (тех, которые должны быть улучшены в первую очередь), обеспечение менеджеров финансовой мерой предлагаемых изменений и т.д. ABC-анализ может проводиться только тогда, когда модель работы последовательная (следует синтаксическим правилам IDEF0), корректная (отражает бизнес), полная (охватывает всю рассматриваемую область) и стабильная

(проходит цикл экспертизы без изменений), другими словами, когда создание модели работы закончено [5].

Параметры стоимостного анализа задаются на вкладке «ABC Units» окна *Model Properties* (рис. 1.11) [2].

Model Name – отображает название текущей модели.

Currency description – для задания единицы измерения денежных средств из списка. Выбранное название единицы измерения денег будет показано в отчетах и в таблице стоимости в диалоговом окне Свойства Работы (*Activity Properties*). Если в списке выбора отсутствует необходимая валюта (например, рубль), то ее можно добавить. По умолчанию символ валюты извлекается из настроек Windows.

Symbol placement – определение символа единицы измерения денег.

Symbol – вид денежного символа, который отображается возле значения стоимости работы на диаграмме;

Number of decimals in diagrams – количество значащих цифр после запятой (0-9), которые показаны на диаграмме на значении стоимости работ. Например, стоимость работы с двумя десятичными разрядами представляется так: \$ 2.35.

Number of decimals in report – количество значащих цифр после запятой (0-9) применительно к отчетам.

Time unit – временной интервал, который используется при определении таких стоимостных факторов, как продолжительность и частота.

Decimals in frequency values – десятичное число (0-9) для использования в стоимостном значении частоты. Можно установить стоимостное значение частоты в таблице стоимости в диалоговом окне свойств работы.

Decimals in duration values – аналогично предыдущему пункту, но в отношении продолжительности.

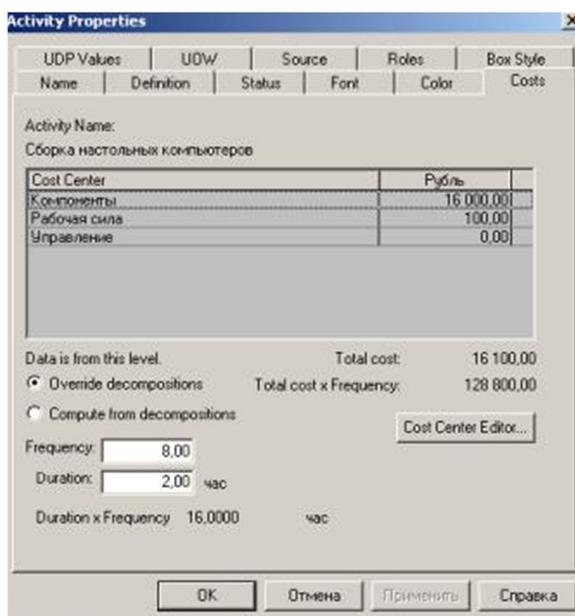


Рис. 1.11. Задание стоимости работ в диалоге Activity Properties/Cost [2]

Далее описываются *центры затрат (cost centers)*. Для внесения центров затрат необходимо вызвать диалог *Cost Center Editor* из меню Model (рис. 1.12). Если в процессе назначения стоимости возникает необходимость внесения дополнительных центров затрат, диалог *Cost Center Editor* вызывается прямо из диалога *Activity Properties/Cost* соответствующей кнопкой.

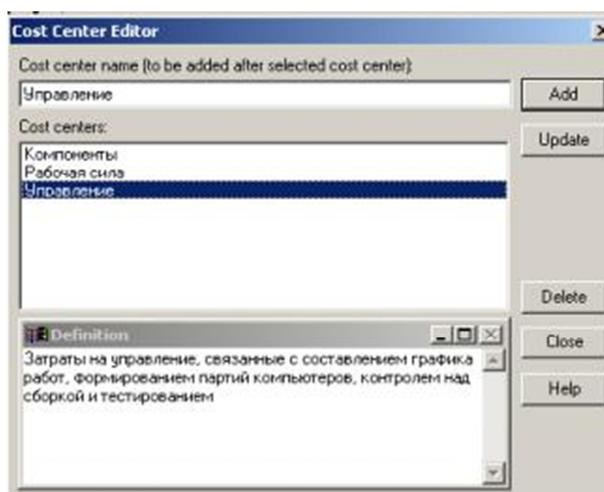


Рис. 1.12. Диалог Cost Center Editor [2]

Каждому центру затрат следует дать подробное описание в окне *Definition*. Список центров затрат упорядочен. Порядок в списке можно менять при помощи стрелок, расположенных справа от списка. Задание определенной последовательности центров затрат в списке, во-первых, облегчает последующую работу при присвоении стоимости работам, а во-вторых, имеет значение при использовании единых стандартных отчетов в разных моделях. Хотя VPwin сохраняет информацию о стандартном отчете в файле BPWINRPT.INI, информация о центрах затрат и UDP сохраняется в виде указателей, т. е.

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

Общие затраты по работе рассчитываются как сумма по всем центрам затрат. При вычислении затрат вышестоящей (родительской) работы сначала вычисляется произведение затрат дочерней работы на частоту работы (число раз, которое работа выполняется в рамках проведения родительской работы), затем результаты складываются. Если во всех работах модели включен режим *Compute from Decompositions*, подобные вычисления автоматически проводятся по всей иерархии работ снизу вверх (рис 1.13). Этот достаточно упрощенный принцип подсчета справедлив, если работы выполняются последовательно.

Обычно все расходы, рассмотренные в модели, можно разбить на две части. В первую войдут затраты, не связанные с конкретными этапами бизнес-процесса. Это, например, оклады сотрудников, оплата за электроэнергию и т.д. Их относят к контекстной диаграмме. Для внесения этих характеристик в модель их описывают в таблице на вкладке *Costs*, причем переключатель *Data is from level* устанавливают в положение *Override Decomposition* [5].



Рис. 1.13. Вычисление затрат родительской работы [4]

Многие диаграммы нижнего уровня можно описать стоимостными характеристиками, присущими данному этапу модели. Так, отметив эти издержки в *Costs*, установив переключатель *Data is from level* в положение *Compute from decompositions*, можно учесть расходы на диаграммах более высоких уровней. Таким образом, на контекстной диаграмме можно просмотреть общие расходы на этапы нижнего уровня (*Override Decomposition*), а также расходы, переданные «вверх» от декомпозиций нижнего уровня (*Compute from decompositions*).

Свойства, определяемые пользователем (UDP)

ABC позволяет оценить стоимостные и временные характеристики системы. Если стоимостных показателей недостаточно, имеется возможность внесения собственных метрик — свойств, определенных пользователем — (*User Defined Properties*, UDP). UDP позволяют провести дополнительный анализ, хотя и без суммирующих подсчетов [5].

Для описания UDP служит диалог *User-Defined Property Editor* (меню Model/UDP Definition Editor) (рис. 1.14). В верхнем окне диалога вносится имя UDP, в списке выбора *Datatype* описывается тип свойства. Имеется возможность задания 18 различных типов UDP, в том числе управляющих команд и массивов, объединенных по категориям. Для внесения категории следует задать имя категории в окне *New Keyword* и щелкнуть по кнопке *Add Category*. Для присвоения свойства категории необходимо выбрать UDP из списка, затем категорию из списка категорий и щелкнуть по кнопке *Update*. Одна категория может объединять несколько свойств, в то же время одно свойство может входить в несколько категорий. Свойство типа *List* может содержать массив предварительно определенных значений. Для определения области значений UDP типа *List* следует задать значение свойства в окне *New Keyword* и щелкнуть по кнопке *Add Member*. Значения из списка можно редактировать и удалять.

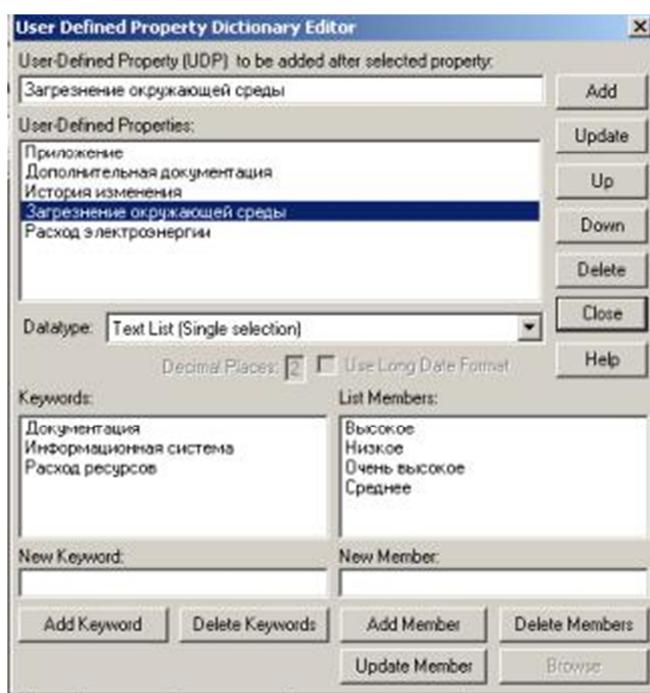


Рис. 1.14. Диалог описания UDP [4]

Каждой работе можно поставить в соответствие набор UDP. Для этого следует щелкнуть правой кнопкой мыши по работе и выбрать пункт меню UDP. В закладке *UDP Values* диалога IDEF0 Activity Properties можно задать значения UDP. Результат задания

можно проанализировать в отчете *Diagram Object Report* (меню *Tools/Report/Diagram Object Report*) [5].

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1. Создать контекстную диаграмму «Функционирование службы новостей»

Описание проблемы: в телевизионных компаниях служба информации – это, как правило, структурное подразделение, занимающееся подготовкой и выпуском программ и новостей в эфир. В зависимости от масштабов и зоны вещания телерадиокомпаний варьируются штат, техническое оснащение, частота выходов в эфир. Однако практически во всех службах действуют общие принципы функционирования, призванные обеспечить оперативную выдачу информации в эфир и наполняемость выпусков [5].

Редакции новостей центральных (федеральных) каналов хорошо организованы и технически обеспечены. Региональные информационные службы (как государственные, так и частные) зачастую ограничены в ресурсах, что не может не сказываться на качестве репортажей. Кроме того, подчас оставляет желать лучшего и порядок работы. Причина – недостаток квалифицированных кадров при неуклонном росте числа телерадиокомпаний, появляющихся в российских регионах. В сложившейся ситуации для оптимизации затрат, а также технических ресурсов и штата в соответствии с потребностями отделов актуально моделирование бизнес-процессов.

1. Запустите Ramus Educational и создайте новую модель «Функционирование службы информации».
2. Заполнить диалоговые окна мастера создания модели в соответствии с данными таблицы 1.10.

Таблица 1.10. Данные модели «Функционирование службы информации»

Название вкладки	Значение
<i>General</i> (общие)	AS-IS («Как есть»).
<i>Purpose</i> (цель модели)	Оптимизация затрат и технических ресурсов
<i>View Point</i> (точка зрения)	Генеральный продюсер
<i>Definition</i> (описание)	Производство и выдачи в эфир выпусков новостей
<i>Scope</i> (границы модели)	Масштаб службы информации
<i>Source</i> (источник)	Данные анализа работы отделов

3. Разместите на контекстной диаграмме «Функционирование службы новостей» стрелки в соответствии с табл. 1.11. При расположении стрелок учитывайте общие правила формирования моделей бизнес-процессов в IDEF0.

Таблица 1.11. Название и описание стрелок, взаимодействующих с работой A0

<i>Arrow Name</i> (Имя стрелки)	<i>Definition</i> (Описание)	<i>Arrow Type</i> (Тип стрелки)
Источники информации	Справочники, газеты, ленты новостей интернета, звонки телезрителей	Input (I1)
Эфирная сетка	Время вещания контента	Control (C1)
Должностные обязанности	Должностные инструкции, приказы, распоряжения, назначения на должности	Control (C2)
Технические ресурсы	Компьютеры, камеры, аппаратно-студийный блок, аппаратная видеозаписи и т.д.	Mechanism (M1)
Штат	Журналисты, редакторы, операторы и режиссеры	Mechanism (M2)
Выпуски новостей	Готовые для выдачи в эфир выпуски новостей	Output (O1)

4. Сохранить в файл *Фамилия_Novosti.bp1*.

Задание 2. Создание диаграммы декомпозиции работы A0 «Функционирование службы информации»

1. На основе знаний, полученных в ходе выполнения лабораторной работы № 1, выполните функциональную декомпозицию контекстной диаграммы. Для создаваемой диаграммы декомпозиции задайте нотацию IDEF0, число работ – 4. Внесите имена и описания созданных работ в диалоговом окне в соответствии с таблицей 1.12.

Таблица 1.12. Данные для диаграммы декомпозиции работы A0

<i>Имя работы (Activity Name)</i>	<i>Определение (Definition)</i>
Планировать эфир (A1)	Планировать выход в эфир рекламы и анонсов, компоновать рекламные блоки, блоки анонсов, новостные блоки
Снимать репортажи (A2)	Составлять график съемок, утверждать сотрудников, выезжать на место съемок
Изготавливать репортажи (A3)	Осуществлять монтаж, утверждать сетку вещания
Эфир (A4)	Назначать ведущих, написание текста репортажей, утверждение хронометража эфира

2. Создайте внутренние и граничные стрелки, взаимодействующие с работами в соответствии с таблицей 1.13. **ВНИМАНИЕ!** Стрелки *Штат* и *Технические ресурсы*, наследуемые от родительской диаграммы, разветвляются и имеют разные имена при соединении с разными работами. Так стрелка *Штат* имеет базовое название *Редактор, корреспонденты, ведущий*. После разветвления появляются три ветви: 1. Редактор, ведущий; 2. Корреспонденты; 3. Администратор. В таблице 1.13, в графе Назначение стрелки указано в скобках название ветви для каждой работы на диаграмме. При

стыковании стрелок и работ необходимо учесть наиболее целесообразное расположение ветвей, чтобы не загромождать диаграмму.

Таблица 1.13. Данные для стрелок на диаграмме декомпозиции А0

Имя стрелки (<i>Arrow Name</i>)	Источник стрелки (<i>Arrow Source</i>)	Тип источника стрелки (<i>Arrow Source Type</i>)	Назначение стрелки (<i>Arrow Destination</i>)	Тип назначения стрелки (<i>Arrow Destination Type</i>)
Источники информации	Граница	Input	Планировать эфир	Input
Эфирная сетка	Граница	Input	Планировать эфир	Input
		Control	Эфир	Control
Должностные обязанности	Граница	Control	Все четыре работы (А1-А4)	Control
График работ	Планировать эфир	Output	Снимать репортажи	Input
			Эфир	Control
Планы съемок	Планировать эфир	Output	Снимать репортажи	Input
Видеоряд в редакции	Снимать репортажи	Output	Изготавливать репортажи	Input
Репортажи	Изготавливать репортажи	Output	Эфир	Input
Штат	Граница	Mechanism	Все четыре работы (А1-А4)	Mechanism
Технические ресурсы	Граница	Mechanism	Все четыре работы (А1-А4)	Mechanism
Выпуски новостей	Эфир	Output	Граница	Output

Задание 3. Создание диаграммы декомпозиции работы А1 «Планировать эфир»

На данном этапе происходит поиск информационных поводов, которые могут стать темами для будущих сюжетов новостей. Каждый вечер накануне эфирного дня, а также утром редактор выпусков вместе с ведущими и корреспондентами просматривает газеты, ленты новостей интернета и поиска информации. Из всех найденных сообщений выбираются те, которые способны заинтересовать зрителей и соответствуют общей концепции вещания. На диаграмме декомпозиции такие критерии указываются стрелками, «помещенными в тоннель». Составляется график, в котором определены временные пределы съемок, написания текста и монтажа. При этом назначаются исполнители –

журналисты, операторы, водители. Корреспонденты консультируются с редактором о том, как лучше сделать сюжет, собирают информацию по теме, договариваются об интервью по телефону и составляют план съемок [5].

1. Выполните декомпозицию работы A1 «Планировать эфир». В диалоговом окне создания диаграммы укажите нотацию IDEF0, число работ – 4. Внесите имена и описания созданных работ в диалоговом окне в соответствии с таблицей 1.14.

Таблица 1.14. Данные для диаграммы декомпозиции работы A1 «Планировать эфир»

<i>Имя работы (Activity Name)</i>	<i>Определение (Definition)</i>
Поиск информационных поводов (A11)	Просмотр редактором, ведущими и корреспондентами газет, новостных лент и сводок.
Отбор тем (A12)	Выбор из найденных информационных поводов тем для сюжета будущих новостей
Составление графика работ (A13)	Определение временных пределов съемок, написания текста и монтажа. Назначение исполнителей работ – журналистов, операторов, водителей.
Подготовка к съемкам (A14)	Консультация корреспондентов с редактором, сбор информации по теме, планирование интервью, составление плана съемок.

2. Создайте внутренние и граничные стрелки, взаимодействующие с работами в соответствии с таблицей 1.15.

Таблица 1.15. Данные для стрелок на диаграмме декомпозиции A1 «Планировать эфир»

<i>Имя стрелки (Arrow Name)</i>	<i>Источник стрелки (Arrow Source)</i>	<i>Тип источника стрелки (Arrow Source Type)</i>	<i>Назначение стрелки (Arrow Destination)</i>	<i>Тип назначения стрелки (Arrow Destination Type)</i>
Источники информации	Граница	Input	Поиск информационных поводов	Input
Интерес зрителя	Tunnel	Control	Отбор тем	Input
Должностные обязанности	Граница	Control	Все четыре работы (A11-A14)	Input
Концепция вещания	Tunnel	Control	Отбор тем	Input
Информацио	Поиск	Output	Отбор тем	Input

Имя стрелки (<i>Arrow Name</i>)	Источник стрелки (<i>Arrow Source</i>)	Тип источника стрелки (<i>Arrow Source Type</i>)	Назначение стрелки (<i>Arrow Destination</i>)	Тип назначения стрелки (<i>Arrow Destination Type</i>)
нные поводы	информационных поводов			
Темы сюжетов	Отбор тем	Output	Составление графика работ	Input
График работ	Составление графика работ	Output	Граница	Output
			Подготовка к съемкам	Input
Планы съемок	Подготовка к съемкам	Output	Граница	Output
Штат (<i>Разделение и именование стрелок у работ</i>)	Граница	Mechanism	Поиск информационных поводов (1. Редактор, ведущий; 2. Корреспонденты)	Mechanism
			Отбор тем (1. Редактор, ведущий)	
			Составление графика работ (1. Редактор, ведущий; 2. Администратор)	
			Подготовка к съемкам (1. Редактор, ведущий; 2. Корреспонденты)	
Технические ресурсы (<i>Разделение и именование стрелок у работ</i>)	Граница	Mechanism	Поиск информационных поводов (1. Телефоны, 2. Компьютеры + интернет)	Mechanism
			Составление графика работ (Компьютеры + планировщик)	
			Подготовка к съемкам (Телефоны)	
Задания на съемки	Составление графика работ	Output	Подготовка к съемкам	Input
Эфирная сетка	Граница	Input	Отбор тем	Input
			Составление графика работ	Input

3. Сохраните изменения в исходный файл.

Задание 4. Создание диаграммы декомпозиции работы A2 «Снимать репортажи»

Съемка новостей осуществляется на основе плана, а также временных ограничений. Съёмочная группа выезжает на место, корреспондент уточняет предварительную информацию, записывает интервью, оператор снимает «картинки», используя телевизионный журналистский комплект (ТЖК) – камеру, микрофон, штатив и светильник. После завершения работы съёмочная группа возвращается в редакцию [5].

1. Выполните декомпозицию работы A2 «Снимать репортажи». В диалоговом окне создания диаграммы укажите нотацию IDEF0, число работ – 4. Внесите имена и описания созданных работ в диалоговом окне в соответствии с таблицей 1.16.

Таблица 1.16. Данные для диаграммы декомпозиции работы A2 «Снимать репортажи»

<i>Имя работы (Activity Name)</i>	<i>Определение (Definition)</i>
Прибытие к месту съемок (A21)	Уточнение времени и места, информирование водителя и транспортного отдела
Уточнение информации (A22)	Сверка новостного сюжета, подготовка ведущих
Съемки (A23)	Видеосъемка новостного сюжета, репортаж ведущего
Возвращение в редакцию (A24)	Уточнение времени и места, информирование водителя и транспортного отдела

2. Создайте внутренние и граничные стрелки, взаимодействующие с работами в соответствии с таблицей 1.17.

Таблица 1.17. Данные для стрелок на диаграмме декомпозиции A2 «Снимать репортажи»

<i>Имя стрелки (Arrow Name)</i>	<i>Источник стрелки (Arrow Source)</i>	<i>Тип источника стрелки (Arrow Source Type)</i>	<i>Назначение стрелки (Arrow Destination)</i>	<i>Тип назначения стрелки (Arrow Destination Type)</i>
Планы съемок	Граница	Input	Прибытие к месту съемок	Input
			Уточнение информации	
			Съемки	
Достижение места съемок	Прибытие к месту съемок	Output	Уточнение информации	Control
Уточненная информация	Уточнение информации	Output	Съемки	Input
Видеоряд «Сырье»	Съемки	Output	Возвращение в редакцию	Input

Имя стрелки (<i>Arrow Name</i>)	Источник стрелки (<i>Arrow Source</i>)	Тип источника стрелки (<i>Arrow Source Type</i>)	Назначение стрелки (<i>Arrow Destination</i>)	Тип назначения стрелки (<i>Arrow Destination Type</i>)
Видеоряд в редакции	Возвращение в редакцию	Output	Граница	Output
График работ	Граница	Control	Все четыре работы (A21-A24)	Control
Должностные обязанности	Граница	Control	Все четыре работы (A21-A24)	Control
Штат (<i>Разделение и именование стрелок у работ</i>)	Граница	Mechanism	Прибытие к месту съемок (1. Водители; 2. Корреспонденты; 3. Операторы)	Mechanism
			Уточнение информации (1. Корреспонденты)	
			Съемки (1. Корреспонденты; 2. Операторы)	
			Возвращение в редакцию (1. Водители; 2. Корреспонденты; 3. Операторы)	
Технические ресурсы (<i>Разделение и именование стрелок у работ</i>)	Граница	Mechanism	Прибытие к месту съемок (Автомобили)	Mechanism
			Съемки (ТЖК)	
			Возвращение в редакцию (Автомобили)	

3. Сохраните изменения в исходный файл.

Задание 5. Создание диаграммы декомпозиции работы А3 «Изготавливать репортажи»

Построение данной диаграммы позволит понять, какие штатные сотрудники участвуют в создании сюжета после съемок. Прежде всего, журналист просматривает видео в просмотрной и расписывает монтажный лист – содержание видеоряда и интервью. Для видео он пишет текст, который в случае необходимости правит редактор.

Сюжет монтируется по проверенному тексту. Иногда во время монтажа журналиста консультирует режиссер.

1. Выполните декомпозицию работы АЗ «Изготавливать репортажи». В диалоговом окне создания диаграммы укажите нотацию IDEF0, число работ – 4. Внесите имена и описания созданных работ в диалоговом окне в соответствии с таблицей 1.18.

Таблица 1.18. Данные для диаграммы декомпозиции работы АЗ «Изготавливать репортажи»

<i>Имя работы (Activity Name)</i>	<i>Определение (Definition)</i>
Написание монтажного листа (А31)	Описание содержания видеоряда и интервью
Написание текста (А32)	Написание текста для видео
Редактура (А33)	Правка текста для видео редактором
Монтаж (А34)	Монтирование сюжета по отредактированному тексту

2. Создайте внутренние и граничные стрелки, взаимодействующие с работами в соответствии с таблицей 1.19.

Таблица 1.19. Данные для стрелок на диаграмме декомпозиции АЗ «Изготавливать репортажи»

<i>Имя стрелки (Arrow Name)</i>	<i>Источник стрелки (Arrow Source)</i>	<i>Тип источника стрелки (Arrow Source Type)</i>	<i>Назначение стрелки (Arrow Destination)</i>	<i>Тип назначения стрелки (Arrow Destination Type)</i>
Видеоряд в редакции	Граница	Input	Написание монтажного листа	Input
			Монтаж	
Монтажный лист	Написание монтажного листа	Output	Написание текста	Input
Авторский текст репортажа	Написание текста	Output	Редактура	Input
Окончательный вариант текста	Редактура	Output	Монтаж	Input
Репортажи	Монтаж	Output	Граница	Output
Должностные обязанности	Граница	Control	Все четыре работы (А31-А34)	Control
График работ	Граница	Control	Все четыре работы (А31-А34)	Control

Имя стрелки (<i>Arrow Name</i>)	Источник стрелки (<i>Arrow Source</i>)	Тип источника стрелки (<i>Arrow Source Type</i>)	Назначение стрелки (<i>Arrow Destination</i>)	Тип назначения стрелки (<i>Arrow Destination Type</i>)
Штат (<i>Разделение и именование стрелок у работ</i>)	Граница	Mechanism	Написание монтажного листа (Корреспонденты)	Mechanism
			Написание текста (Корреспонденты)	
			Редактура (1. Редактор; 2. Корреспонденты)	
			Монтаж (1. Режиссер; 2. Корреспонденты; 3. Монтажер)	
Технические ресурсы (<i>Разделение и именование стрелок у работ</i>)	Граница	Mechanism	Написание монтажного листа (Просмотровая)	Mechanism
			Написание текста (Фабрика новостей)	
			Монтаж (Монтажная)	

3. Сохраните изменения в исходный файл.

Задание 6. Создание диаграммы декомпозиции работы А4 «Эфир»

Верстка выпуска предполагает определение порядка следования репортажей, написание подводок, проверку на соответствие временным лимитам. При этом используется сетевая компьютерная программа СУБД «Фабрика новостей». Заполняется микрофонная папка, которая визируется главным редактором. Режиссер проверяет готовность сюжетов и составляет эфирный лист с порядком следования сюжетов и номерами кассет для сотрудников аппаратно-студийного блока (АСБ) и аппаратной видеозаписи (АВЗ). Во время эфира ведущий находится в студии, техническая поддержка осуществляется АСБ, сюжеты запускаются в АВЗ [5].

1. Выполните декомпозицию работы А4 «Эфир». В диалоговом окне создания диаграммы укажите нотацию IDEF0, число работ – 3. Внесите имена и описания созданных работ в диалоговом окне в соответствии с таблицей 1.20.

Таблица 1.20. Данные для диаграммы декомпозиции работы А4 «Эфир»

<i>Имя работы (Activity Name)</i>	<i>Определение (Definition)</i>
Верстка выпусков (А41)	Определение порядка следования репортажей, написание подводок, проверку на соответствие временным лимитам.
Подготовка к эфиру (А42)	Проверка готовности сюжетов и составление эфирного листа с порядком следования сюжетов и номерами кассет для служб технической поддержки
Выходы в эфир(А43)	Работа ведущего в студии, техническая поддержка эфира сотрудниками аппаратно-студийного блока, запуск сюжетов в аппаратной видеозаписи.

2. Создайте внутренние и граничные стрелки, взаимодействующие с работами в соответствии с таблицей 1.21.

Таблица 1.21. Данные для стрелок на диаграмме декомпозиции А4 «Эфир»

<i>Имя стрелки (Arrow Name)</i>	<i>Источник стрелки (Arrow Source)</i>	<i>Тип источника стрелки (Arrow Source Type)</i>	<i>Назначение стрелки (Arrow Destination)</i>	<i>Тип назначения стрелки (Arrow Destination Type)</i>
Репортажи	Граница	Input	Верстка выпусков	Input
Микрофонная папка	Верстка выпусков	Output	Подготовка к эфиру	Input
Эфирный лист для АВЗ	Подготовка к эфиру	Output	Выходы в эфир	Input
Завизированная микрофонная папка	Подготовка к эфиру	Output	Выходы в эфир	Input
Выпуски новостей	Выходы в эфир	Output	Граница	Output
Должностные обязанности	Граница	Control	Все три работы (А41-А43)	Control
График работ	Граница	Control	Верстка выпусков	Control
Эфирная сетка	Граница	Control	Все три работы (А41-А43)	Control
Штат (Разделение и	Граница	Mechanism	Верстка выпусков (1.Редактор	Mechanism

Имя стрелки (<i>Arrow Name</i>)	Источник стрелки (<i>Arrow Source</i>)	Тип источника стрелки (<i>Arrow Source Type</i>)	Назначение стрелки (<i>Arrow Destination</i>)	Тип назначения стрелки (<i>Arrow Destination Type</i>)
именование стрелок у работ)			4. Ведущий, режиссёр)	
			Подготовка к эфиру (1. Ведущий, режиссер; 2. Главный редактор)	
			Выходы в эфир (Ведущий, режиссер)	
Технические ресурсы (Разделение и именование стрелок у работ)	Граница	Mechanism	Верстка выпусков (Фабрика)	Mechanism
			Подготовка к эфиру (Фабрика)	
			Выходы в эфир (1. Студия) (2. АСБ) (3. АВЗ)	

3. Сохраните изменения в исходный файл.

Задание 7. Создание диаграммы декомпозиции работы A11 «Поиски информационных поводов»

1. Выполните декомпозицию работы A11 «Поиски информационных поводов». В диалоговом окне создания диаграммы укажите нотацию IDEF0, число работ – 5. Внесите имена и описания созданных работ в диалоговом окне в соответствии с таблицей 1.22.

Таблица 1.22. Данные для диаграммы декомпозиции работы A11 «Поиск информационных поводов»

Имя работы (Activity Name)	Определение (Definition)
Поиск информационных поводов в газетах (A111)	Дать самостоятельно
Поиск информационных поводов в Интернете (A112)	Дать самостоятельно
Определение информационных поводов по звонкам (A113)	Дать самостоятельно
Рассмотрение памятных дат (A114)	Дать самостоятельно
Определение информационных поводов по пресс-релизам (115)	Дать самостоятельно

4. Дайте описание работам, на примере описания в соответствии с таблицами, рассмотренными ранее.

5. Создайте внутренние и граничные стрелки, взаимодействующие с работами в соответствии с рис. 1.15.

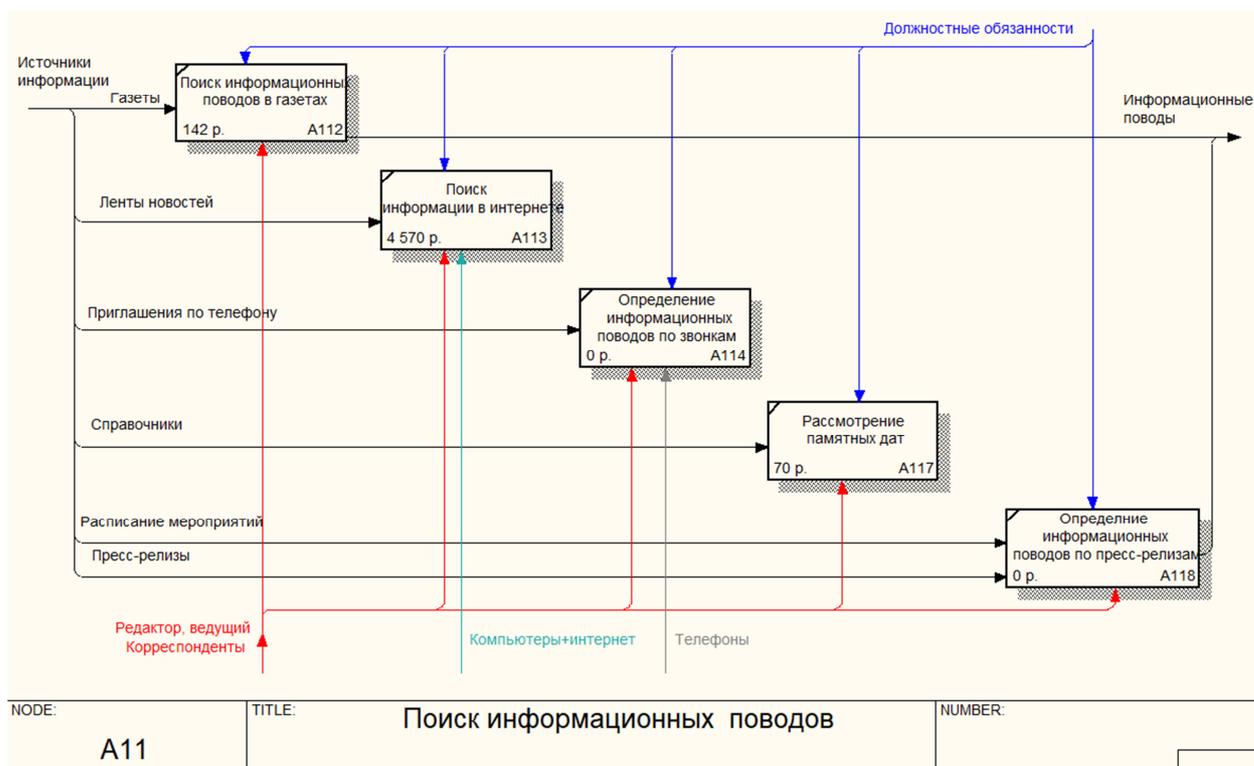


Рис.1.15. Диаграмма декомпозиции «Поиск информационных поводов»

6. Составьте таблицу Стрелок и типов источников на примере таблиц 1.13, 1.15, 1.17, 1.19, 1.21.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

- 1) Какие существуют критерии для определения момента завершения моделирования;
- 2) Чем отличается метод функционально стоимостного анализа от традиционных финансовых методов;
- 3) Что необходимо предпринять, в случае если стоимостных показателей системы ABC недостаточно?
- 4) Что означает выбор переключателя *Data is from level* в положения *Override Decomposition* и *Compute from Decomposition*?
- 5) При каком условии можно начинать функционально-стоимостный анализ?
- 6) Какие характеристики необходимо указать, прежде чем приступить к анализу стоимости работы?
- 7) Какова основная задача ФСА?
- 8) Чем обоснован функционально-стоимостный подход к рассмотрению объекта?
- 9) Что является основным критерием совершенствования с позиции ФСА?

- 10) Какие задачи можно решить при помощи метода ФСА?
- 11) Зачем в BPWin существует функция слияния моделей?
- 12) Зачем в BPWin существует функция расщепления моделей?
- 13) В чем заключается удобство применения FEO-диаграмм для создания моделей TO-BE?

ЛАБОРАТОРНАЯ РАБОТА № 3

Моделирование потоков данных

Цель работы: получение навыков применения нотации DFD для моделирования документооборота.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Диаграмма потоков данных – **DFD (Data Flow Diagramming)** используется для описания схемы документооборота, технологических процессов обработки информации и движения материальных потоков в качестве дополнения к моделям IDEF0. Для описания DFD используются элементы [2]:

Работы (Activities) – функции обработки информации;

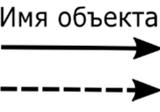
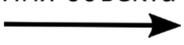
Стрелки (Arrows) – документы, объекты, сотрудники, отделы, участвующие в обработке информации;

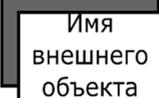
Внешние ссылки (External References) – обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;

Хранилище данных (Data Store) – таблица для хранения документов.

Для построения диаграмм потоков данных в большей степени используются нотации Гейна-Сарсона или Йордона - ДеМарко. Типовые элементы обеих нотаций представлены в табл. 1.23.

Таблица 1.23. Типовые элементы графических нотаций диаграмм DFD

Элемент	Описание	Нотация Йордона - ДеМарко	Нотация Гейна - Сарсона
Функция	Работа		
Поток данных	Объект, над которым выполняется работа. Может быть логическим или управляющим. Управляющие потоки обозначаются пунктирной	<p>Имя объекта</p> 	<p>Имя объекта</p>  <p>(Понятие управляющего потока отсутствует)</p>

Элемент	Описание	Нотация Йордона - ДеМарко	Нотация Гейна - Сарсона
	линией со стрелкой		
Хранилище данных	Структура для хранения информация объектов		
Внешняя сущность	Внешний по отношению к системе объект, обменивающийся с ней потоками		

Работы – функции системы, преобразующие входы в выходы, изображаются прямоугольниками со скругленными углами, смысл их совпадает со смыслом работ в IDEF0 [2].

Внешние сущности – входы в систему и/или выходы из нее, изображаются в виде прямоугольников с тенью. Данный графический элемент может использоваться многократно на одной или нескольких диаграммах [2].

Стрелки (потоки данных) – описывают движение объектов из одной части системы в другую. В отличие от IDEF0, стороны работ на диаграммах DFD не имеют четкого назначения, стрелки могут подходить и выходить из любой грани прямоугольника. Для описания диалогов типа «команда-ответ» применяются двунаправленные стрелки между работами, между работой и внешней сущностью и между внешними сущностями, а также между работой и хранилищем данных [2].

Хранилище данных используются для изображения объектов «в покое»; в материальных системах хранилища данных изображаются там, где объекты ожидают обработки, например, в очереди. В системах обработки информации хранилища данных являются механизмом, который позволяет сохранить данные для последующих процессов. Пример хранилища данных - база данных, архив документов и т.п. [2].

Слияние и разветвление стрелок используется также в диаграммах DFD. Это позволяет описать декомпозицию стрелок. Каждому новому сегменту сливающейся или разветвляющейся стрелки можно задать собственное имя [2].

Построение диаграмм DFD начинается с построения **контекстной диаграммы**. Контекстная диаграмма состоит из прямоугольника работы, изображающей систему в целом и внешние сущности, взаимодействующие с системой. В качестве внешних сущностей используются источники и приемники информации, объекты управления (вещи, люди, процессы). Номер работы может включать префикс, номер родительской работы и номер объекта. Номер объекта является уникальным номером работы на диаграмме. Например, А.10.3. Хранилища данных и внешние сущности также могут иметь

уникальные номера, независимо от их расположения на диаграмме. Хранилища данных имеют префикс *D*, а внешние сущности префикс *E*.

На диаграмме DFD отображаются работы, входящие в состав описываемого бизнес-процесса, а также входы и выходы каждой работы – информационные и материальные потоки. Например, от поставщиков поступают материалы и информация, которые затем становятся внешними входами. Эти внешние входы уходят наружу к потребителю в виде результатов деятельности, выходной информации. При этом временная последовательность работ не обязательно должна совпадать с направлением движения потоков в бизнес-процессе. При построении DFD-схемы бизнес-процесса рекомендуется отмечать подразделения и должности, участвующие и отвечающие за выполнение всех работ, входящих в процесс. Существует правило, по которому необходимо формировать название работы:

**НАЗВАНИЕ РАБОТЫ = ДЕЙСТВИЕ + ОБЪЕКТ, НАД КОТОРЫМ
ДЕЙСТВИЕ ОСУЩЕСТВЛЯЕТСЯ.**

Название материальных и информационных потоков образуется следующим образом:

**НАЗВАНИЕ ПОТОКА = ОБЪЕКТ, ПРЕДСТАВЛЯЮЩИЙ ПОТОК + СТАТУС
ОБЪЕКТА**

На диаграммах DFD используются межстраничные ссылки (*Off-Page Refrence*) и внешние сущности (*External Refrence*). Для межстраничной ссылки в DFD нужно создать новую граничную стрелку, которая автоматически появится с квадратными скобками. Далее следует указать ее тип Off-Page Refrence. Для внешней сущности создается новая граничная стрелка, для которой указывается тип External Reference. Затем выбирается имя внешней сущности или же задается имя новой сущности.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание № 1. Создать словарь структурного подразделения

1. Запустите Ramus Educational;
2. В диалоговом окне введите имя модели – Документооборот и укажите тип диаграммы – DFD; В описании укажите: описание документооборота при изготовлении мебели по заказу клиента.

Задание № 2. Создать контекстную диаграмму DFD

На контекстную диаграмму необходимо добавить:

Работу – Изготовить мебель по заказу клиента;

Внешние ссылки – Клиент, Склад, Менеджер;

Входные потоки – Дизайн-проект, Заказ клиента, Материалы для изготовления.

Выходные потоки – Готовая мебель.

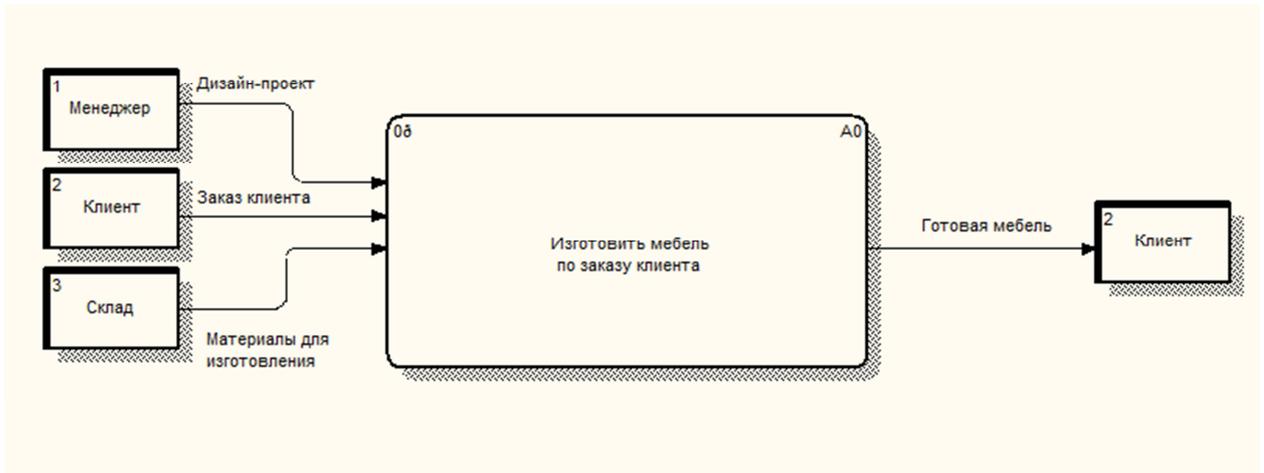


Рис. 1.16. Контекстная DFD диаграмма

Задание № 3. Создать диаграмму декомпозиции DFD

1. По аналогии с диаграммой IDEF0 создайте диаграмму декомпозиции DFD.
2. На диаграмму декомпозиции добавьте элементы (рис. 1.17):

Внешние ссылки:

- ✓ Консультант.

Хранилища данных:

- ✓ БД Заказы.
- ✓ Шаблоны документов.
- ✓ Наряд на изготовление.

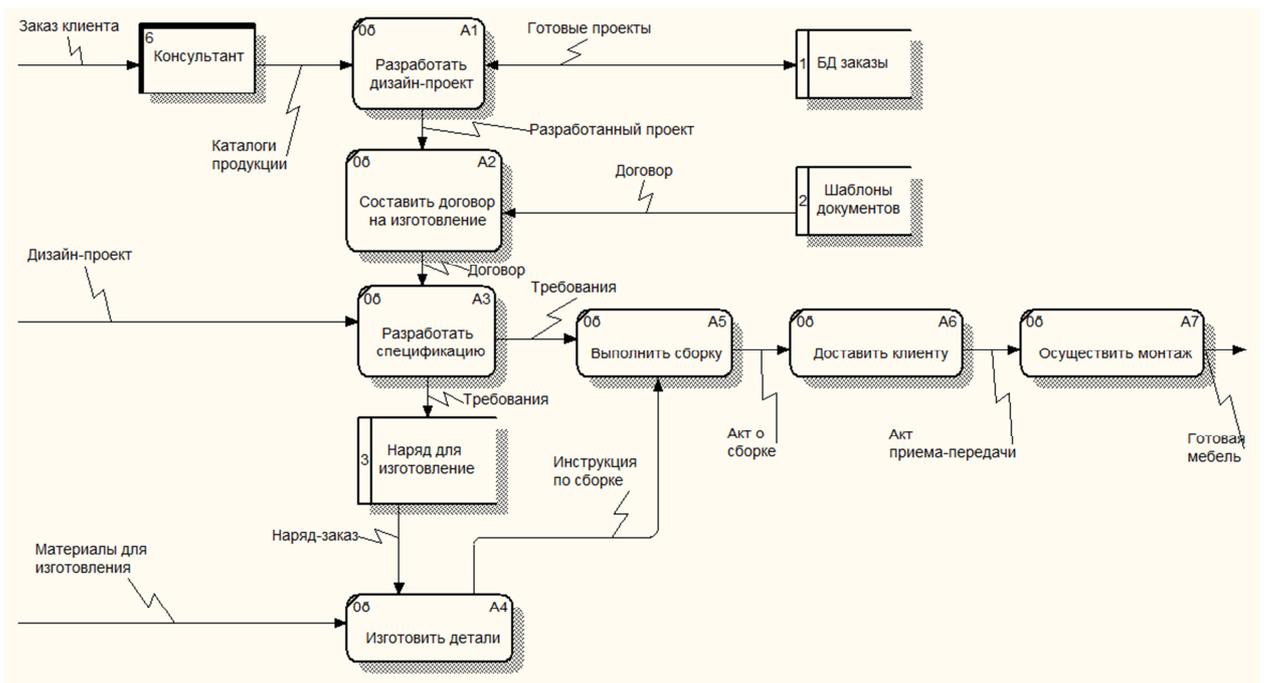


Рис. 1.17. Диаграмма декомпозиции DFD

Диаграмма декомпозиции демонстрирует процесс документооборота при выполнении работы «Изготовить мебель по заказу клиента»:

1. От клиента поступает заказ на изготовление мебели. Консультант демонстрирует каталоги продукции и на основе готовых образцов из базы данных «Заказы» разрабатывает дизайн-проект. Разработанный проект также заносится в БД Заказы.

2. На основе проекта менеджер составляет договор, беря типовой договор из хранилища шаблонов.

3. После разработки договора и на основе дизайн-проекта составляется спецификация на изготовление мебели.

4. При оформлении наряда на изготовление учитываются требования спецификации. Составляется наряд-заказ для производственного цеха на изготовление деталей.

5. После изготовления деталей составляется инструкция по сборке. При выполнении сборки учитываются требования спецификации.

6. После выполнения сборки составляется акт о сборке и отправляется в отдел доставки

7. После доставки заказа клиенту акт приема-передачи подписывается двумя сторонами.

8. Затем осуществляется монтаж мебели.

Задание № 4. Разработайте собственную DFD диаграмму

На основе примера создайте собственную DFD диаграмму.

Задание № 5. Подготовить отчет

Подготовьте отчет по лабораторной работе в MS Word, добавьте в него контекстную диаграмму, диаграмму декомпозиции, описание процесса документооборота, а также таблицу исполнителей ролей (табл. 1.24).

Таблица 1.24. Исполнители ролей

Разработать дизайн-проект	Менеджер
Составить договор на изготовление	Менеджер
Разработать спецификацию	Замерщик
Изготовить детали	Плотник
Выполнить сборку	Сборщик
Доставить клиенту	Грузчик
Осуществить монтаж	Сборщик

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

1. Для чего используются диаграммы потоков данных?
2. Какие элементы применяются при построении DFD диаграмм?
3. Что описывает элемент «Работа» в нотации DFD?
4. Что описывает элемент «Внешняя сущность» в нотации DFD?
5. Что описывает элемент «Хранилище данных» в нотации DFD?
6. По какой формуле формируется название работы?
7. По какой формуле формируется название потока?
8. Каков порядок построения DFD диаграмм?
9. Может ли диаграмма декомпозиции от родительской IDEF0-диаграммы быть диаграммой DFD?

ЛАБОРАТОРНАЯ РАБОТА № 4

Динамическое IDEF3-моделирование

Цель работы: получение навыков применения нотации IDEF3 для моделирования вариантов последовательностей выполнения операций.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

IDEF3 – методология моделирования и стандарт документирования процессов, происходящих в системе. Метод документирования технологических процессов представляет собой механизм документирования и сбора информации о процессах. IDEF3 показывает причинно-следственные связи между ситуациями и событиями в понятной эксперту форме, используя структурный метод выражения знаний о том, как функционирует система, процесс или предприятие [2].

Методология IDEF3 находит широкое применение при разработке информационных систем. При этом используется инструмент визуального моделирования бизнес-процессов [2]. Процессная модель IDEF3 позволяет: а) отразить последовательность процессов; б) показать логику взаимодействия процессов системы. Основной целью IDEF3 является предоставление аналитикам возможности описать ситуацию, когда процессы выполняются в определенной последовательности, а также объекты, участвующие совместно в одном процессе.

Также как во всех рассмотренных выше нотациях IDEF3-модель представляет собой набор диаграмм. При этом основными компонентами IDEF3-модели являются:

- 1) единицы работ;

- 2) связи;
- 3) перекрестки;
- 4) объекты ссылок.

Единица работы

В стандарте IDEF3 вводится понятие «единица работы» - Unit of Work (UOW). Единицу работу изображают прямоугольником, который имеет имя и номер (рис. 1.18). Имя работы является словосочетанием с отглагольным существительным, изображающим действие (выполнение, изготовление) или инфинитивом глагола (изготовить продукцию). Номер работы имеет вид: номер родительской работы + порядковой номер на текущей диаграмме [2].



Рис. 1.18. Единица работы

Связи

Связи показывают взаимоотношения работ. Особенностью связей в IDEF3 является то, что они однонаправленные и могут быть направлены куда угодно. Однако, обычно, при изображении диаграмм связи направляют слева направо.

Связи, которые могут быть установлены между единицами работ и их описания показаны в табл. 1.25.

Таблица 1.25.. Типы связей в нотации IDEF3

Название стрелки	Описание
Старшая (Precedence)	Сплошная линия слева направо, сверху вниз. UOW в начале связи должен закончиться раньше, чем начнется UOW, стоящий в конце связи
Стрелка отношений (Relation Link)	Пунктирная линия, для изображения связей между UOW с объектами ссылок; работа-источник не обязательно должна заканчиваться прежде начала работы-цели
Потоки объектов (Object Flow)	Описание того факта, что объект используется в 2 или более UOW. Часто изображается стрелкой с двумя наконечниками

Наличие связей типа Relation Link следует интерпретировать следующим образом: существует неопределенная временная зависимость между началом и окончанием этих работ (рис. 1.19) [2].

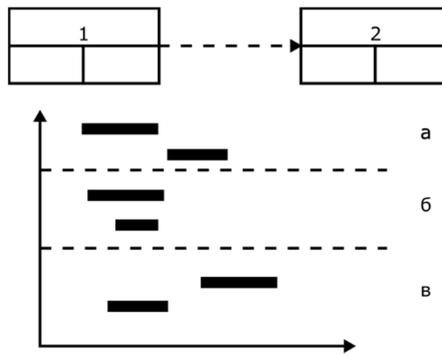


Рис. 1.19. Наличие связей типа Relation Link в IDEF3

Вариант «а» означает, что UOW1 заканчивается ранее UOW2; вариант «б» означает, что UOW2 может заканчиваться ранее UOW1; вариант «в» означает, что UOW1 начинается позднее UOW2.

Перекрестки

Перекрестки (Junction) обеспечивают отображение логики взаимодействия стрелок при слиянии и разветвлении (табл. 1.26).

Таблица 1.26. Обозначение перекрестков в IDEF3

Обозначение	Наименование	Смысл в случае слияния стрелок (Fan-in-Junction)	Смысл в случае разветвления стрелок (Fan-out-Junction)
	Асинхронное «И»	Все предшествующие процессы должны быть завершены для начала следующего процесса	Все следующие процессы должны быть запущены в любое время
	Синхронное «И»	Все предшествующие процессы должны быть завершены одновременно для начала следующего процесса	Все следующие процессы должны быть запущены, при этом в одно и то же время
	Асинхронное «ИЛИ»	Один или более предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены, время запуска - произвольное
	Синхронное «ИЛИ»	Один или более предшествующих процессов завершены одновременно	Один или несколько следующих процессов должны быть запущены, при этом в одно и то же время
	Исключающее «ИЛИ»	Только один из предшествующих процессов выполняется	Только один из следующих процессов запускается

Для нумерации перекрестков указывается префикс **J**. Стрелки могут сливаться или разветвляться ТОЛЬКО через перекрестки. Для создания перекрестков в IDEF3 существуют правила:

1. Каждому перекрестку для слияния должен предшествовать перекресток для разветвления.
2. Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа синхронного или асинхронного «ИЛИ».
3. Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа, исключающего «ИЛИ».
4. Перекресток для слияния типа, исключающего «ИЛИ», не может следовать за перекрестком для разветвления типа «И».
5. Перекресток, имеющий одну стрелку на одной стороне, должен иметь более одной стрелки на другой стороне.

Перекрестки на диаграммах могут образовывать комбинации (рис. 1.20). При этом правила создания перекрестков справедливы и для комбинаций.

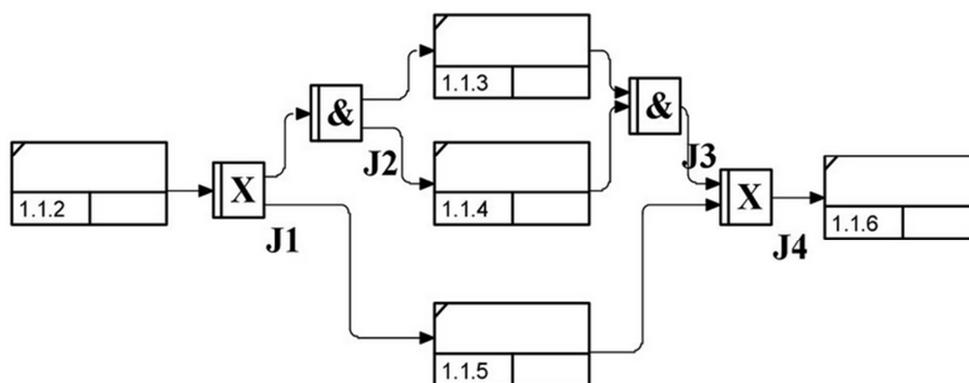


Рис. 1.20. Пример комбинации перекрестков

Объект ссылки

Когда при создании IDEF3-модели необходимо разработчику необходимо сделать акцент на важной детали и добавить заметку, используются объекты ссылок (*Referent*). Они не связаны связями с работами или перекрестками, а призваны лишь отобразить концепцию, идею или данные. Типы объектов ссылок показаны в табл. 1.27.

Таблица 1.27. Типы объектов ссылок

Типы объектов ссылок	Описание
ОБЪЕКТ	Описывает участие важного объекта в работе
GOTO	Циклический переход, может ссылаться на перекресток
UOB	Множественное нециклическое использование работы
NOTE	Документирование важной работы
ELAB	Детальное описание разветвления или слияние стрелок на перекрестках

Декомпозиция работ

Также как при построении IDEF0-диаграмм, диаграммы нотации IDEF3 можно декомпозировать. Допускается многократная декомпозиция модели, при этом в одной модели можно описать разные сценарии (потоки). Особенностью IDEF3 является то, что декомпозиция может быть сценарием или описанием.

- ✓ Сценарий - описывает только один путь развития процесса.
- ✓ Описание - включает все возможные пути развития процесса.

Взаимодействие точек входа со сценарием или декомпозицией указано в таблице 1.28.

Таблица 1.28.

	Сценарий	Декомпозиция
Точка входа	Одна	Одна
Точка выхода	Несколько	Одна

Каждая работа имеет номер, состоящий из номера родительской работы, номера декомпозиции, номера работы на текущей диаграмме. Например, A11.1.4 означает A11-номер родительской работы, 1 – первая декомпозиция, 4- четвертая работа на текущей диаграмме.

Сценарий

Работа над сценарием предусматривает передачу экспертом аналитику текстовое описание сценария. Аналитик составляет список:

- ✓ Кандидатов на работы (выражаются отглагольным существительным, обозначающим процесс, одиночное или в составе именного словосочетания);
- ✓ Кандидатов на объекты (существительные, обозначающие результат выполнения работы).

В IDEF3 логику согласования модели допустимо выразить в виде комбинации перекрестков или в виде ссылки типа ELAB.

В результате дополнения диаграмм IDEF0 диаграммами IDEF3 и DFD может быть создана смешанная диаграмма. AllFusion Process Modeler допускает следующие переходы от модели к модели:

1. IDEF0→DFD (удалить все граничные стрелки, создать соответствующие внешние сущности, хранилища данных, создать внутренние стрелки, начинающиеся с внешних сущностей, тоннелировать стрелки на диаграмме IDEF0);
2. IDEF0→IDEF3;
3. DFD→IDEF3.

Чтобы выполнить процесс моделирования на основе стандарта IDEF3, с помощью программы AllFusion Process Modeler, нужно выполнить установку нескольких опций в меню *Tools*→*Preferences* (рис. 4.4).

Программа DIA для построения IDEF3-моделей

Для построения IDEF3-моделей в данной лабораторной работе используется свободно распространяемый кроссплатформенный редактор диаграмм, блок-схем, сетевых диаграмм, UML-диаграмм под названием «DIA». Небольшая проблема заключается в том, что стандартного набора элементов для построения диаграмма в графической нотации IDEF3 в DIA не предусмотрено, тем не менее все нужные блоки в программе есть. Их нужно просто сгруппировать вручную. Для этого нажимаем в меню: "Файл -> категории и объекты". В открывшемся окне нажимаем кнопку "Создать". Откроется ещё одно окошко, в котором выбираем пункт "Название категории" и вписываем туда "idef3". Процесс создания категории выглядит примерно так [6] (рис. 1.21):

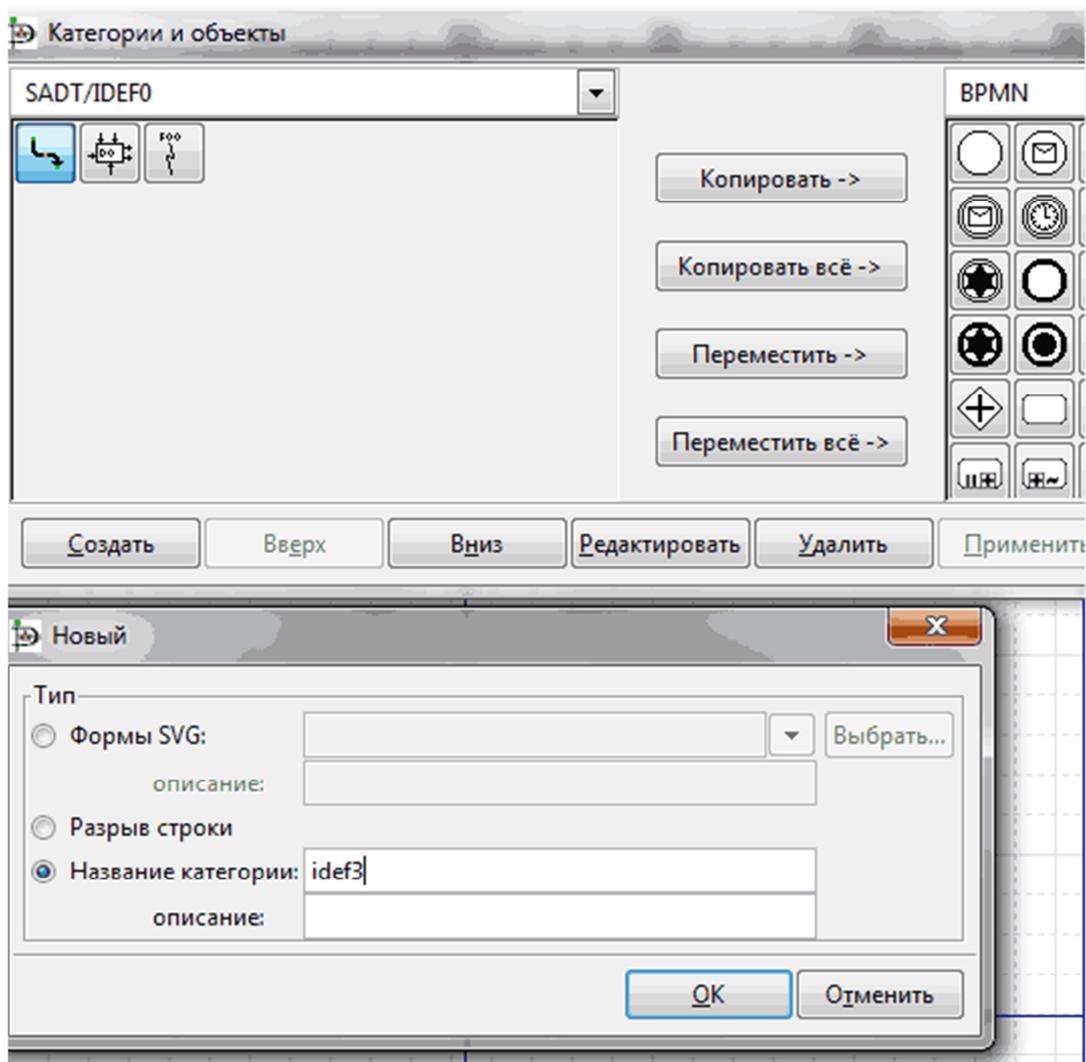


Рис. 1.21. Добавление элементов IDEF0 в категорию IDEF3 в DIA

Так как вы только что создали эту категорию - естественно она пуста. Нам нужно переместить в нее нужные элементы схем. Поэтому:

- ✓ для начала справа выбираем категорию SADT/IDEF0, там будет 3 элемента. Нажимаем кнопку "копировать все" (смотрите не нажмите случайно "переместить все");
- ✓ теперь справа выбираем категорию "RE-Джексона" и копируем фигуру "прямоугольник" с одинарной линией слева;
- ✓ теперь выбираем категорию "Блок-схема" и оттуда добавляем прямоугольник с линиями по бокам (рис. 1.22);
- ✓ и наконец, выбираем категорию UML и добавляем оттуда чёрную изогнутую линию (при её использовании придётся постоянно менять ее опции, возможно есть линии получше в других типах диаграмм, но я не нашел);

В итоге должно получиться вот так:

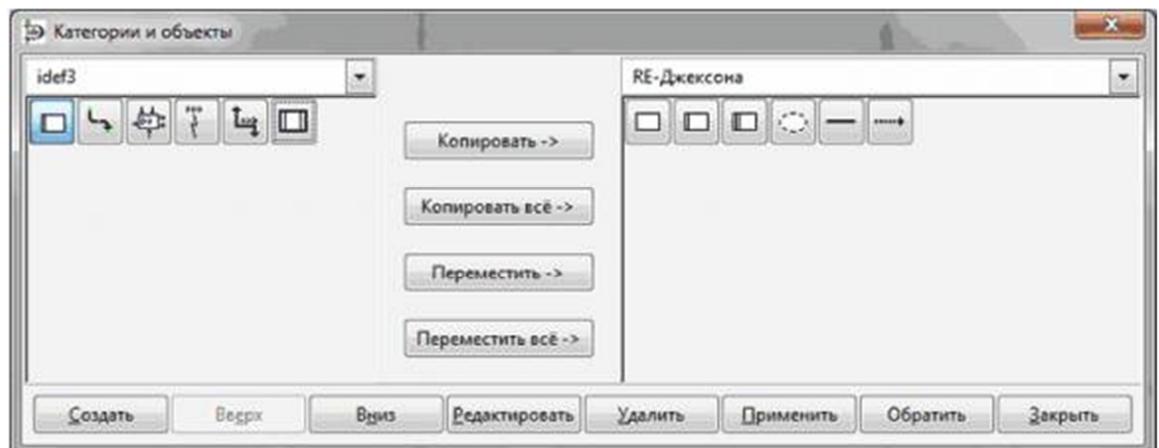


Рис. 1.22. Добавление прямоугольника с линиями по бокам, для синхронных перекрестков

Жмем кнопку "Применить", "Закрыть" окошко и подборка элементов для нотации IDEF3 готова. Заходим в "другие библиотеки элементов" и выбираем там созданную нами графическую нотацию «IDEF3» (она располагается в положенной ей месте по алфавиту). Кстати, чтобы писать в блоках, удобно использовать клавишу F2. Конечно, это не идеальный инструмент, но этот способ позволяет создавать диаграммы IDEF3 максимально приближенно к их точной графической нотации.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание № 1. Создать IDEF3 диаграмму

- 1) Запустите программу DIA;

- 2) С помощью библиотеки элементов «IDEF3» создайте IDEF3-модель «Демонстрация каталогов и образцов», показанную на рис. 1.23.
- 3) Добавьте объекты ссылки. Имена объектов: «ОБЪЕКТ / Клиент», «NOTE / Обновлять продукцию 1 раз в 3 месяца», «NOTE / Запрашивать обновления у поставщика 1 раз в месяц», «NOTE / Запрашивать обновления у поставщика 1 раз в месяц».
- 4) Свяжите объекты, так как показано на рис. 1.23.
- 5) Добавьте описания работ и объектов, путем вставки комментариев на рабочее поле программы. Комментарии можно добавить из библиотеки элементов UML.

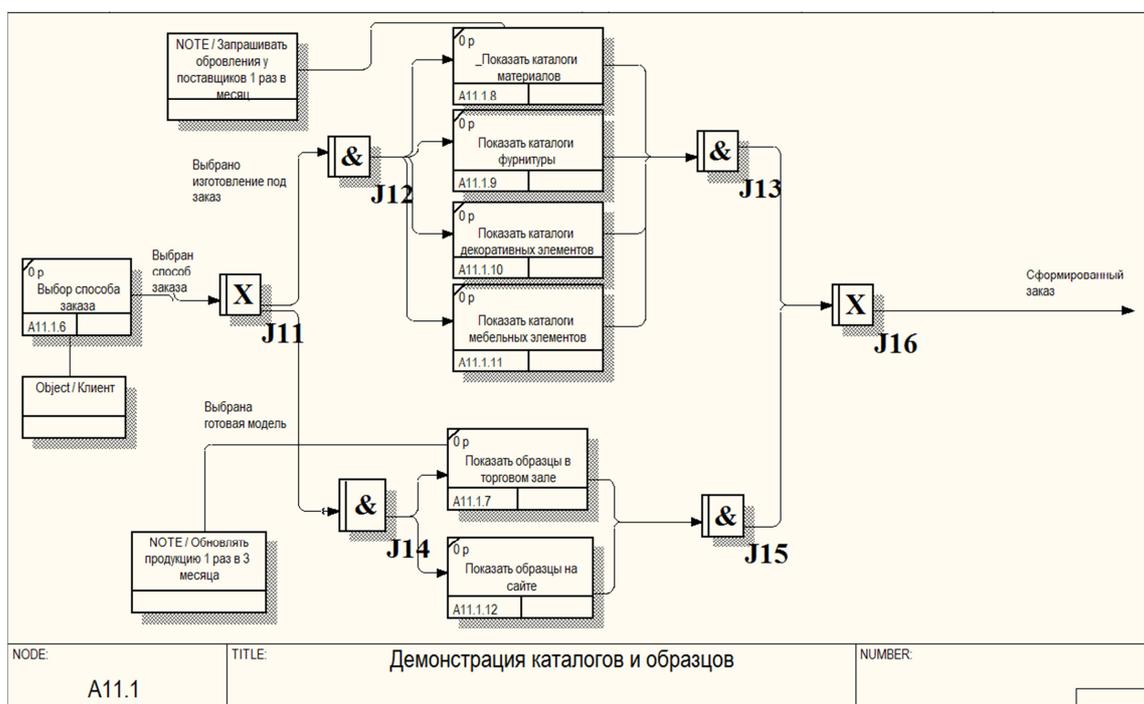


Рис. 1.23. IDEF3-диаграмма «Демонстрация каталогов и образцов»

Заполнить свойства работ на диаграмме A11.1 согласно таблице 1.29.

Таблица 1.29.

Номер работы	Название	Описание Definition	Объект Object
A11.1.1	Выбор способа заказа	Клиенту предлагается способ приобретения мебели: изготовление на заказ или выбор из готового ассортимента компании	Клиент
A11.1.2	Показать каталоги материалов	Демонстрируются каталоги с образцами материалов для изготовления мебели	Каталог «Пленки ПВХ», каталог фасады «МДФ», каталог фасады «ДСП», каталог «Фасады дерево

Номер работы	Название	Описание Definition	Объект Object
			массив», каталог «Столешницы искусственный камень», каталог «Столешницы постформинг, каталог «УФ Фотопечать».
A11.1.3	Показать каталоги фурнитуры	Демонстрируются каталоги с образцами фурнитуры	Каталог «Мебельные ручки», каталог «Мойки», каталог «Ножки мебельные», каталог «Держатели и доводчики мебельные», каталог «Мебельные механизмы»
A11.1.4	Показать каталоги декоративных элементов	Демонстрируются каталоги с образцами декора	каталог «Декоративные элементы», каталог «Витражи и декоративные стекла», каталог «Карнизы»
A11.1.5	Показать каталоги мебельных элементов	Демонстрируются каталоги мебельных элементов	Каталог «Вытяжки», каталог «Спинки кровати», каталог «Дверные накладки»,
A11.1.6	Показать образцы в торговом зале	Клиенту демонстрируются образцы продукции компании, находящиеся в торговом зале	Готовая продукция: кухни, шкафы-купе, мебель для дома, мебель для офиса
A11.1.7	Показать образцы на сайте	Клиенту демонстрируются образцы продукции компании, находящиеся на сайте компании.	Образцы продукции, ранее изготавливаемые на заказ другим клиентам и впоследствии помещенные в каталог на сайте компании

6) Создать IDEF3 диаграмму декомпозиции работы A13 «Заключение договора, расчет стоимости, предоплата» самостоятельно.

7) Создать IDEF3 диаграмму декомпозиции работы A14 «Разработка и утверждение спецификации» самостоятельно.

8) Продумайте и разместите логически необходимые перекрестки и объекты ссылок на созданных диаграммах декомпозиции.

9) Сохранить модель в файл Модель_IDEF3

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

1. В чем состоит отличие IDEF0 и IDEF3 диаграмм?

2. Какие стрелки используются для связей единиц работ?
3. Каковы основные компоненты IDEF3-моделей?
4. Что показывают связи работ?
5. Каковы особенности связей в IDEF3?
6. Каков смысл использования перекрестка асинхронное «И» в случае слияния и разветвления стрелок?
7. Каков смысл использования перекрестка синхронное «И» в случае слияния и разветвления стрелок?
8. Каков смысл использования перекрестка асинхронное «ИЛИ» в случае слияния и разветвления стрелок?
9. Каков смысл использования перекрестка синхронное «ИЛИ» в случае слияния и разветвления стрелок?
10. Каков смысл использования перекрестка исключаящее «ИЛИ» в случае слияния и разветвления стрелок?
11. Какие типы объектов ссылок существуют в IDEF3?
12. Каковы правила создания перекрестков?
13. В чем особенность декомпозиции IDEF3-диаграмм?
14. Что означает номер работы A2.2.3 на IDEF3-диаграмме?

ЛАБОРАТОРНАЯ РАБОТА № 5

Разработка модели бизнес-процессов предприятия с использованием методологий IDEF0, DFD, IDEF3

Цель работы: закрепить навыки применения методологии функционального моделирования для разработки моделей бизнес-процессов, потоков данных и документооборота компании.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Выполнение лабораторных работ №1-№4 позволяет получить базовые навыки по разработке моделей бизнес-процессов и реализации процессного подхода. Лабораторные работы были посвящены трем графическим нотациям. Методология функционального моделирования IDEF0 позволяет выполнить формализацию и полное описание бизнес-процессов, с акцентом на соподчинённость работ. Диаграммы потоков данных (DFD) делают удобным детальное описание документооборота как отдельного процесса, так и предприятия в целом. Диаграммы потоков работ (IDEF3) позволяют моделировать

временные последовательности выполнения бизнес-процессов. А также проанализировать несколько вероятных сценариев развития моделируемого процесса.

Особенностью программного продукта All Fusion Process Modeler является одновременная поддержка трех графических нотаций. Таким образом, диаграмма декомпозиции может быть DFD- или IDEF3-диаграммой. Это очень удобно для полного описания деятельности компании или какого-либо ее подразделения. Можно не только детально выделить все основные и вспомогательные процессы компании, но и установить хронологию исполнения, описать движение документооборота в каждом процессе. В частности, при разработке СМК, подобные возможности является несомненным преимуществом.

В данной лабораторной работе необходимо самостоятельно разработать комплексную модель одного бизнес-процесса с использованием всех трех изученных нотаций.

ПРАКТИЧЕСКАЯ ЧАСТЬ

1) Распределиться по группам (2 человека) и выбрать тему для разработки модели. Темы для разработки представлены ниже.

2) Используя ресурсы сети Интернет кратко изучите предметную сферу выбранной темы;

3) Разработать модель по выбранной теме с использованием методологии IDEF0. Разработанная модель должна удовлетворять следующим требованиям:

4) Содержать контекстную диаграмму;

5) Содержать не менее двух уровней функциональной декомпозиции;

6) Каждая работа и стрелка должна иметь название и описание;

7) На главной странице с контекстной диаграммой должен быть заполнен «подвал»;

8) Разработайте DFD-диаграмму документооборота компании для одной из работ по своему выбору;

9) Разработайте IDEF3 диаграмму для одной из работ по своему выбору;

10) Проведите функционально-стоимостный анализ основной деятельности компании;

11) Создайте FEO-диаграмму основного процесса и попробуйте перестроить процессы с целью их улучшения.

Примерные темы для разработки моделей

- 1) Деятельность компании доставки пиццы;
- 2) Деятельность компании доставки суши;
- 3) Деятельность службы заказа такси;
- 4) Деятельность салона красоты;
- 5) Деятельность компании по организации праздников;
- 6) Деятельность химчистки;
- 7) Деятельность библиотеки;
- 8) Деятельность книжного интернет-магазина;
- 9) Деятельность кредитного отдела банка;
- 10) Деятельность автосервиса;
- 11) Деятельность турфирмы;
- 12) Деятельность отдела сбыта.
- 13) Тема по выбору студента (предварительно согласовывается с преподавателем).

ГЛАВА II. НОТАЦИЯ BPMN ДЛЯ МОДЕЛИРОВАНИЯ БИЗНЕС ПРОЦЕССОВ

В 70-80-х годах прошлого века началось массовое снижение конкурентоспособности американских бизнес-компаний. В частности, японские компании стали успешно конкурировать с американскими прямо на внутреннем рынке США. В поисках путей повышения эффективности американского бизнеса в начале 1990-х годов в США появилась новая парадигма организации бизнеса, ориентированная на процессы. В результате, в лексикон бизнеса и IT-технологий вошли такие термины, как бизнес-процесс (business process), реинжиниринг бизнеса (business reengineering), реинжиниринг бизнес-процессов (business process reengineering), моделирование бизнес-процессов (business process modeling).

Нотация по моделированию бизнес-процессов **BPMN** (The Business Process Modeling Notation) - это новый стандарт для моделирования бизнес процессов и сетевых услуг, который впервые был выпущен BPMI Notation Working Group в мае 2004 года. Последняя версия нотации BPMN 2.0 вышла в 2010 году. Оригинальная спецификация (на английском языке) изготовлена группой компаний «Object Management Group» [7].

Людам, занимающимся бизнесом, крайне удобно работать с бизнес-процессами, отображаемыми в виде блок-схем. Множество бизнес-аналитиков проектируют и описывают бизнес-процессы компаний с помощью простых диаграмм в нотации BPMN, т.к. язык нотации понятен даже на уровне пользователя. При этом модели процессов, описанных в нотации BPMN, являются ИСПОЛНЯЕМЫМИ (т.е. реализуются в любой BPM-системе), а не только документируются. Для детального описания процессов существуют программные решения, которые способны преобразовать диаграммы в исполняемые процессы, эти процессы затем могут быть запущены и работать в реальном времени.

Нотация BPMN описывает условные обозначения для отображения бизнес-процессов в виде диаграмм бизнес-процессов. BPMN ориентирована как на технических специалистов (разработчиков, ответственных за реализацию процессов), так и на бизнес-пользователей (бизнес-аналитиков, создающих и улучшающих процессы) и менеджеров, следящих за процессами и управляющих ими. Следовательно, BPMN призвана служить связующим звеном между фазой дизайна бизнес-процесса и фазой его реализации. Для этого язык использует базовый набор интуитивно понятных элементов, которые позволяют определять сложные семантические конструкции.

Процесс с точки зрения бизнеса - это отдельная деятельность (часть бизнес-процесса), выполняемая компанией или организацией. В терминологии BPMN процесс является сложным действием, которое, в свою очередь, состоит из действий, переходов между ними и т. д. Процесс можно вызывать, приостанавливать, прерывать, также он может завершаться сам, процессы могут выполняться параллельно и обмениваться сообщениями.

Моделирование в BPMN осуществляется посредством диаграмм с небольшим числом графических элементов. Это помогает пользователям быстро понимать логику процесса. Выделяют четыре основные категории элементов:

- ✓ **Объекты потока управления:** события, действия и логические операторы;
- ✓ **Соединяющие объекты:** поток управления, поток сообщений и ассоциации;
- ✓ **Роли:** пулы и дорожки;
- ✓ **Артефакты:** данные, группы и текстовые аннотации;

Элементы этих четырёх категорий позволяют строить простейшие диаграммы бизнес процессов (ДБП). Для повышения выразительности модели спецификация разрешает создавать новые типы объектов потока управления и артефактов.

Таким образом, процесс в BPMN может состоять из нескольких конструкций, описанных в таблице 2.1.

Таблица 2.1. Конструкции BPMN

Сущности (flows objects):	Связи (connecting objects) - соединяют разные действия и данные в единый поток исполнения, могут быть следующих видов:	Участники (swimlanes) процесса:	Артефакты (artifacts) процесса
<ul style="list-style-type: none"> ▪ действие (activity); ▪ шлюзы (gateway); ▪ событие (event); 	<ul style="list-style-type: none"> ▪ поток исполнения (sequence flow) - переход от одного действия к другому; ▪ поток сообщений (message flow) - обмен сообщениями между разными участниками процесса; ▪ ассоциация (association) - определяет переход между действиями в особых ситуациях (например, при возникновении исключений); может использоваться для "прикрепления" комментариев, данных и пр.; 	<ul style="list-style-type: none"> ▪ внешние (pools); ▪ внутренние (lanes); 	<ul style="list-style-type: none"> ▪ данные (data object), группы (groups), комментарии (annotations).

При выполнении лабораторных работ главы II эти 4 типа конструкций будут рассмотрены подробнее.

В курсе лабораторных работ можно использовать любое свободно распространяемое программное обеспечение, реализующее в своем наборе элементов нотацию BPMN. Примерами такого ПО могут быть Aris Express, Bizagi Modeler. Данное ПО доступно для скачивания Free Download (свободное скачивание) при условии регистрации в сообществе пользователей. В описании лабораторных работ приводятся диаграммы в нотации BPMN, выполненные в программе Aris Express. Применение данного программного продукта не является строгим и допускается выполнение лабораторных работ на любом другом ПО, поддерживающем эту нотацию. Также существуют наборы инструментов для построения BPMN-диаграмм, работающие бесплатно в режиме онлайн, т.е. через браузер. Их главным плюсом является то, что не требуется установка компьютера. Главным системным требованием их использования является наличие современного браузера. Примером такого инструментария может служить BPMN Studio, доступный по ссылке <https://bpmn.studio/ru>.

ЛАБОРАТОРНАЯ РАБОТА № 6

Введение в нотацию BPMN

Цель: изучение основных элементов категории «Сущность» в нотации BPMN

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1. События

Любой процесс, описанный в нотации BPMN, представляет собой последовательное или параллельное выполнение различных действий (операций) с указанием определённых бизнес-правил. Рассмотрим простой пример процесса «Обработка заказа» (рис. 2.1), который может реализовываться в рамках продажи и аренды электроинструментов через интернет-магазин [7]. Чтение процесса всегда начинается со **Стартового события** (не закрашенный кружок с тонкой линией) рис. 2.2.

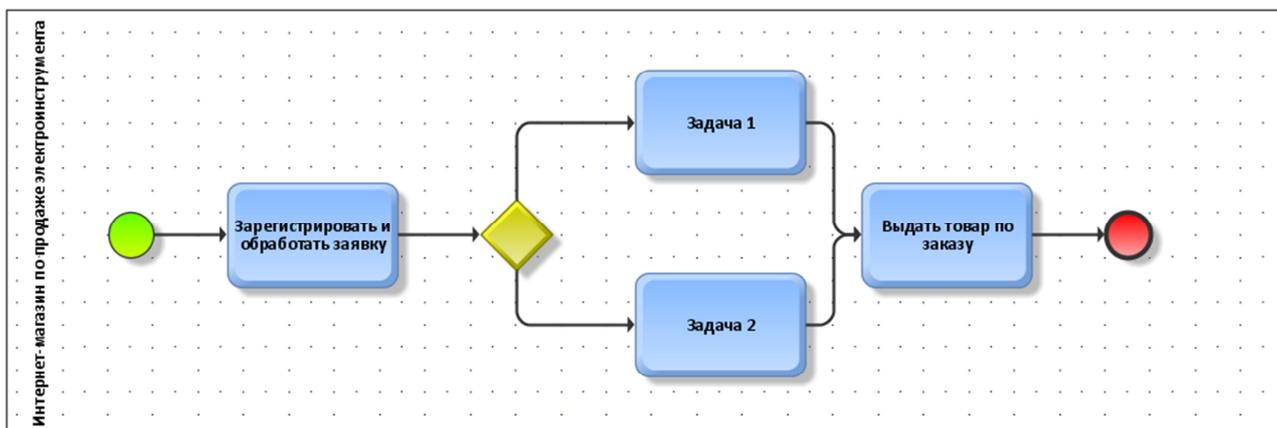


Рис. 2.1. Процесс «Обработка заказа»

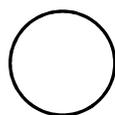


Рис.2.2. Стартовое событие

Как видно из названия, **Стартовое событие** указывает на то, в какой точке берет начало тот или иной процесс. В контексте потока операций **Стартовое событие** является начальной точкой в процессе; это означает, что никакой входящий поток операций не может быть соединен со стартовым событием. Например, стартовым событием процесса-примера является звонок или письмо от клиента на сайт компании (интернет-магазина).

Далее от **Стартового события** выполнение процесса идет по линиям (Поток операций) до **Конечного события** (не закрашенный кружок с утолщенной линией), их может быть несколько (рис. 2.3).



Рис.2.3. Конечное событие

Конечное событие указывает на то, в какой точке завершается тот или иной процесс. В контексте Потока операций **Конечное событие** завершает ход Процесса; это означает, что никакой Исходящий поток операций не может быть соединен с **Конечным событием**.

Конечное событие представляет собой круг, выполненный одиночной, жирной линией. Толщина линии должна быть жирной настолько, чтобы без труда можно было отличить **Конечное событие** от **Стартового**. В разных системах моделирования начальное и конечное состояния могут отличаться также по цвету, например, в системе ELMA. Конечное событие процесса «Обработка заказа» отображает завершение процесса – выдачей заказанного товара.

Кроме **Стартового** и **Конечного** события, в описании бизнес-процессов используются **Промежуточные события**. Промежуточное событие влияет на ход процесса, однако, не может являться началом или завершением процесса и само по себе не является полноценным действием. Примерами **Промежуточных событий** являются: ожидание определённого времени, события, письма. Промежуточное событие изображается в виде круга со свободным центром (рис. 2.4). Для отличия от **Стартового** и

Конечного типов событий, изображение круга Промежуточное событие выполнено двойной тонкой линией.

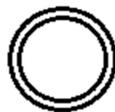


Рис.2.4. Промежуточное событие

VRMN выделяет несколько типов каждого **События**: Сообщение, Таймер, Эскалация и другие - их достаточно большое количество. Для определения типа **События** используются различные маркеры, позволяющие отличить данный тип **События** от другого.

Маркер – это специальный значок, рисуемый в центре круга **События**. Он влияет на характер работы **События**.



- простое **Стартовое событие**.



- **Стартовое событие -таймер** позволяет запустить процесс по таймеру в определённый момент времени (пример с совещанием в 9.00час.), т.е. меняется характер запуска бизнес-процесса.



- **Стартовое событие-сообщение** показывает, что от участника поступает Сообщение, которое инициирует запуск Процесса (например, от клиента приходит оплата товара, приходит сообщение об оплате и запускается процесс выдачи товара).



- простое **Конечное событие**. Данный тип **Конечного события** не подразумевает какой-то определенный результат.



- **Конечное событие-сообщение** служит для указания того, что Участник отправил Сообщение в момент завершения Процесса.



- обычное **Промежуточное событие**.



- маркер часов показывает, что используется событие-таймер. В ход процесса при этом останавливается на определённое время.



- маркер конверта определяет **Промежуточное событие** типа «Сообщение», которое используется для отправки сообщения другому участнику Процесса. Данный вид **Промежуточного события** может стать инициатором **Стартового события-сообщения** другого процесса.

Наиболее часто для описания бизнес-процессов применяют **Промежуточное событие-таймер**, которое позволяет моделировать моменты времени, периоды и таймауты. **Промежуточное событие** данного типа графически изображается с аналоговыми часами внутри круга.

Промежуточное событие-таймер используется для того, чтобы приостановить ход процесса до определённого времени либо задать определённую цикличность выполнения действия (например, планирование и информирование о совещании каждую неделю в понедельник в 9.00). На рис. 2.5. представлен расширенный вариант процесс «Обработка заказа». В приведённом примере процесса используется промежуточное событие-таймер с целью приостановить ход процесса пока не придёт заказанный с фабрики товар. т.е. пока заказанный товар не придёт на склад магазина, формировать заявку и рассматривать вопрос о добавлении подарка не начнут. **Таймер** (Промежуточное событие-таймер) позволяет обозначить на диаграмме процесса это ожидание.

Второй пример использования промежуточного события-таймер в рамках процесса - ожидание момента, когда заказанный товар придёт клиенту; после чего с ним необходимо связаться, чтобы выяснить удовлетворённость клиента работой интернет-магазина либо попросить оставить отзыв на сайте компании-продавца (обратная связь).

2.2. Действия

Вся логика работы (ход) процесса выражается во всевозможных элементах, расположенных между **Стартовым** и **Конечным** событием. Основным элементом, отражающим деятельность, выполняемую внутри процесса, являются **Действия**. **Действия** – это точки выполнения работ в ходе Процесса. Они относятся к выполняемым

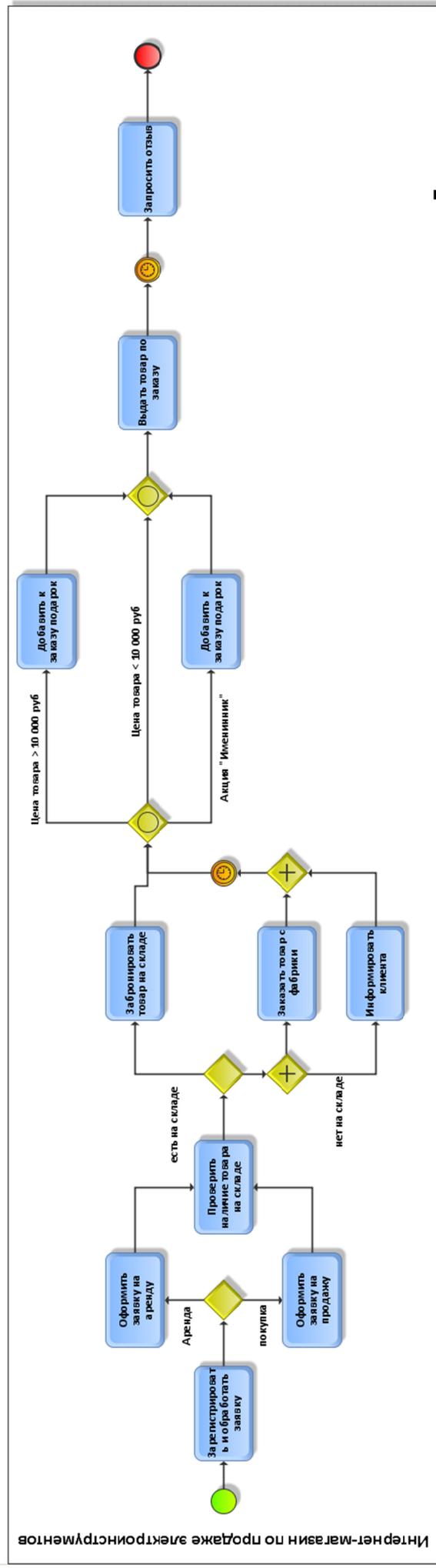


Рис. 2.5. Расширенный вариант процесс «Обработка заказа»

элементам Процесса BPMN. Элементарное **Действие** выражается в выполнении одной единственной **Задачи**. В BPMN существует три типа действия:

1. задача;
2. свернутый подпроцесс;
3. развернутый подпроцесс.

Задача (task) - это атомарное действие процесса, неделимое на более элементарные части. На диаграмме задача изображается, как показано на рис. 2.6а. В BPMN могут быть заданы три вида задач: циклическая задача, множественная задача и откат.

Циклическая задача (loop) - это задача, которая выполняется в цикле. В параметрах этой задачи можно указать, какой цикл имеется в виду - с пред- или постусловием, определить это условие и указать некоторые дополнительные свойства цикла (рис. 2.6 б).

Множественная задача (multiple instance) - это циклическая задача, которая выполняет в цикле целый набор однотипных задач. Текстовыми параметрами можно задать условие цикла, количество однотипных задач, а также порядок их выполнения (последовательный или параллельный) (рис. 2.6 в).

Откат (compensation) - задача, которая вызывается в случае отмены другой задачи (рис. 2.6 г). Например, клиент отказался от забронированного отеля - тогда система должна освободить соответствующую бронь; пример приводится на рис. 2.7.

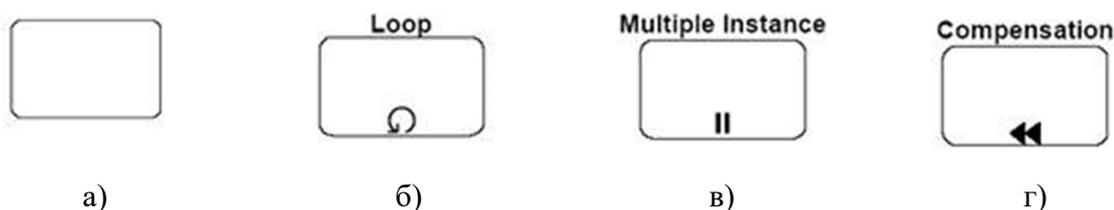


Рис. 2.6. Графическое изображение задач разного типа в BPMN

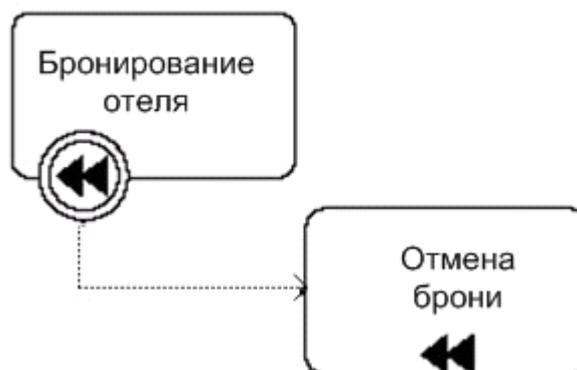


Рис. 2.7. Пример задачи с откатом

Самой распространённой **Задачей** является типичная для технологического процесса задача, где человек участвует в качестве исполнителя. Такие **Задачи** называются Пользовательскими.

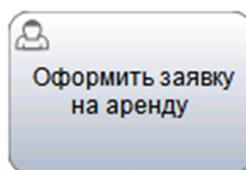


Рис.2.8. Пользовательская задача

Также у каждой задачи есть атрибут, который может иметь одно из следующих значений:

- ✓ **Service** – задача является сторонним программным сервисом, вызываемым WE (это значение имеют по умолчанию все задачи); например, вызывается Web-сервис, вычисляющий погоду, курс валюты или еще что-нибудь;
- ✓ **Receive** – задача является ожиданием внешнего для данного бизнес-процесса события, часто является началом бизнес-процесса;
- ✓ **Send** – задача является посылкой сообщения во внешний для данного бизнес-процесса контекст;
- ✓ **User** – задача выполняется человеком или группой, при этом используется некоторая сторонняя IT-технология или сервис; в параметрах можно задать как исполнителей так и используемую ими ПО;
- ✓ **Script** – задача является скриптом, который WE выполняет полностью автоматически;
- ✓ **Manual** – задача, которая выполняется без помощи WE или другой IT- технологии или сервиса, например, посредством личного общения менеджера с заказчиком;
- ✓ **Reference** – задача является ссылкой на другую задачу;
- ✓ **None** – значение данного атрибута не задано.

Эти значения не имеют графического представления и могут быть отражены, например, в имени задачи. Список этих атрибутов может быть расширен.

На рис. 2.5. в рамках процесса «Обработка заказа» основными действиями являются задачи «Зарегистрировать и обработать заявку», «Оформить заявку на покупку» и «Оформить заявку на аренду».

2.3. Шлюзы

Другой элемент нотации, часто используемый в описании процессов – **Шлюзы** (Условия). Графический элемент **Шлюза** представляет собой небольшой ромб, используемый во многих нотациях схем бизнес-процессов для изображения ветвления и знакомый большинству инструментов моделирования (рис. 2.9). Фактически **Шлюз** - есть совокупность входов и выходов.



Рис.2.9. Шлюз

Шлюзы используются для контроля расхождений и схождений потока операций в рамках процесса. Термин шлюз подразумевает пропускное устройство, которое либо позволяет осуществлять переход через шлюз, либо нет. На рис. 2.1., в приведённом примере, в зависимости от желания клиента (купить или арендовать электроинструмент), заявка оформляется в формате покупки либо аренды соответственно. В данном процессе Шлюз указывает, что процесс может пойти только в одном из описанных направлений, т.е. либо покупка, либо аренда.

Элемент **Шлюз (Условия)** так же имеет несколько вариаций. В первом примере (рис. 2.1) был использован *Исключающий Шлюз «ИЛИ» (Эксклюзивный)*. *Исключающие Шлюзы* включаются в состав бизнес-процесса для разделения Потока операций на несколько альтернативных маршрутов. Для процесса с *исключающим* типом **Шлюза** может быть выбран лишь один из предложенных маршрутов (поэтому определяется как Шлюз «ИЛИ»).



Рис.2.10. Исключающий Шлюз

Графический элемент *Исключающий Шлюз* не имеет внутренних маркеров.

Условие можно представить себе в виде вопроса, который появляется в какой-то точке процесса и предполагает несколько вариантов ответов. Каждый из предлагаемых ответов связан с определённым направлением потока операций. В процессе-примере *Исключительный Шлюз* используется два раза: первый Шлюз определяется вопросом «Покупают электроинструмент или арендуют?» и в зависимости от ответа процесс протекает либо в одном направлении «Оформить заявку на покупку» либо в другом – «Оформить заявку на аренду»; второй Шлюз определяется вопросом «Есть ли заказанный товар на складе?», по результатам которого товар либо бронируют на складе, либо, в случае отсутствия на складе, делают заказ на фабрику-производитель. Обе задачи выполняться одновременно не будут.

Второй тип **Шлюза**, так же часто используемый в описании процессов – *Параллельный Шлюз «И»*. Данный тип **Шлюза** используется для создания параллельных маршрутов и их синхронизации (объединения).



Рис. 2.11. Параллельный Шлюз

Графический элемент *Параллельный Шлюз* содержит внутренний маркер, выполненный в виде знака «+», что позволяет отличить данный тип **Шлюза** от других (рис. 2.11).

С помощью *Параллельного Шлюза* параллельные маршруты создаются без необходимости проверки каких-либо условий. При разветвлении все исходящие потоки (маршруты) активизируются одновременно. Закрывающий **Шлюз** используется для синхронизации, т.е. он ожидает завершения выполнения всех входящих ветвей (маршрутов) и только затем активирует выходной поток. рассматриваемом примере, на рис. 2.5., Параллельный Шлюз разъединяет процесс на два параллельных маршрута с операциями «Заказать товар на фабрике» и «Информировать клиента», которые выполняются одновременно, не исключая друг друга: сотрудник может связаться с фабрикой и заказать нужный велосипед, и предупредить клиента о задержке по выдаче товара. Второй - закрывающий Параллельный Шлюз - используется для синхронизации потока операций, т.е. обязательно ожидается завершение обоих действий.

Ещё один, но менее распространённый тип **Шлюза** – *Неисключающий (Неэксклюзивный) Шлюз «И/ИЛИ»*. *Неисключающие Шлюзы* используются для разделения потока операций на несколько альтернативных и параллельных маршрутов.

Для данного экземпляра процесса может быть выбран лишь один из предложенных маршрутов – или параллельный (т.е. оба маршрута выполняются параллельно) или альтернативный (т.е. процесс пойдёт только по одному, соответствующему условиям, маршруту).



Рис.2.12. Неисключающий Шлюз

Графический элемент *Неисключающий Шлюз* содержит внутренний маркер, выполненный в виде круга, что позволяет отличить данный тип **Шлюза** от других (рис. 2.12). На диаграмме 2.5 описаны несколько возможных маршрутов потока операций, разделённые Неисключающим Шлюзом:

Два возможных альтернативных маршрута:

1. При условии, что сумма заказанного товара (велосипеда) больше 10 000 руб., то к заказу добавляется подарок. При этом дополнительного товара заказано не было. Маршрут идёт только в одном направлении.

2. Если сумма товара меньше 10 000 руб., и в заказе доптовара нет, то подарок не добавляется. Маршрут также идёт только в одном направлении.

Два возможных параллельных маршрута:

1. Если велосипед заказан на сумму больше 10 000руб., и дополнительно заказаны товары на сумму от 5 000руб., то клиент получит два подарка. Ход процесса идёт параллельно по двум маршрутам.

2. Если сумма велосипеда меньше 10 000руб., но дополнительные товары превышают 5000 руб., то клиент получает подарок. Также процесс проходит по двум направлениям параллельно. Закрывающий Неисключающий Шлюз, также, как и Параллельный Шлюз, синхронизирует потоки операций.

ПРАКТИЧЕСКАЯ ЧАСТЬ

- 1) Запустить программу Aris Express. В главном окне программы выбрать пиктограмму BPMN diagram (рис. 2.13 а).
- 2) Откроется главное окно, содержащее рабочую область и набор элементов Symbols для построения диаграмм (рис. 2.13 б).

Соответствующий участник процесса добавляется путем щелчка на палитре инструментов и последующего щелчка на поле для построения диаграммы.

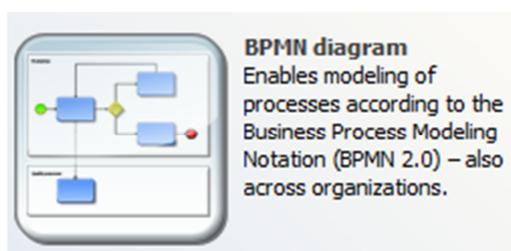


Рис. 2.13а. Пиктограмма для запуска палитры инструментов нотации BPMN

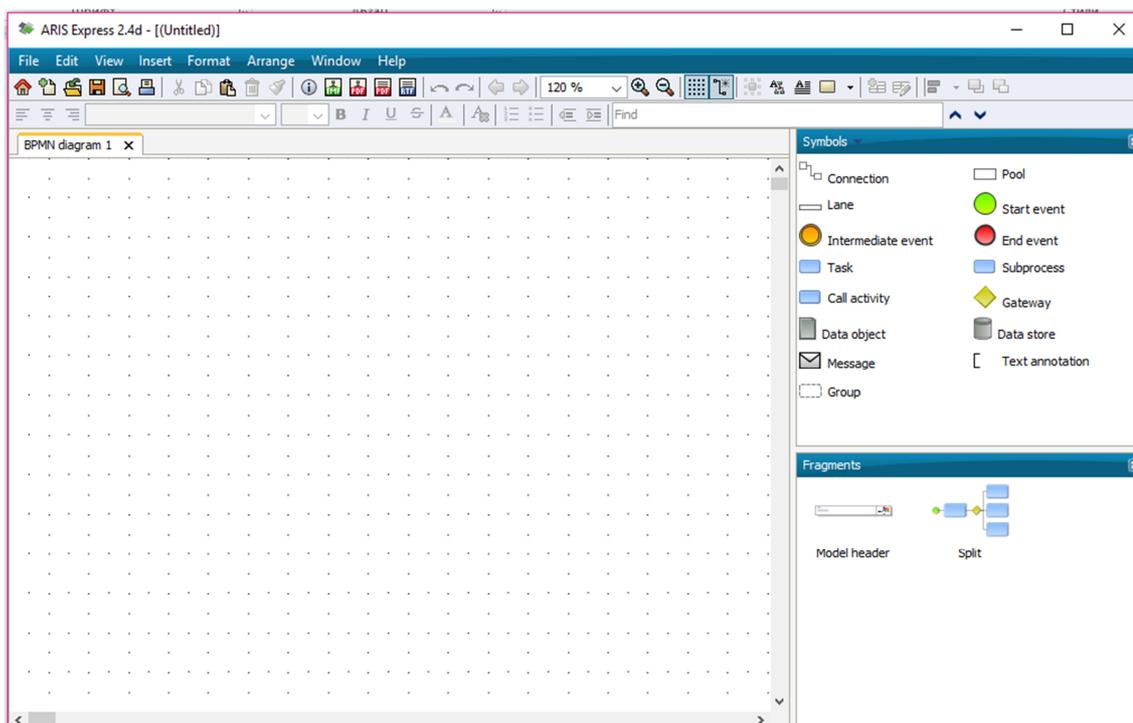


Рис. 2.136. Главное окно программы ARIS Express для работы с нотацией BPMN

Задание 1.

Создать диаграмму процесс «Обработка заказа», показанную на рис. 2.1.

Задание 2.

Создать диаграмму расширенный вариант процесса «Обработка заказа», показанную на рис. 2.5.

Задание 3.

На основании текстового описания бизнес-процесса разработать диаграмму процесса «Прием нового сотрудника» в медицинском учреждении в нотации BPMN.

Текстовое описание бизнес-процесса «Прием нового сотрудника»

Цель: директор клиники «Евромед» желает автоматизировать процесс приема на работу сотрудников и довести его до всех участников данного процесса.

Методы получения информации: интервью, изучение нормативных документов и должностных инструкций.

Работа строится на основе серии проведенных интервью и их анализа. В результате интервью и анализа документов были выявлены 8 участников данного процесса:

1. директор частной медицинской клиники «Евромед»,
2. руководитель структурного подразделения (в которое устраивается работник),
3. начальник кадрово-правового отдела (КПО),
4. зам. начальника кадрово-правового отдела (КПО),

5. бухгалтер,
6. юрисконсультант,
7. эпидемиолог,
8. инженер по охране труда.

Процесс включает около 16-20 функций и длится примерно 3 месяца. Между участниками процесса идет активный обмен документами.

Процесс трудоустройства на основании серии интервью можно описать так:

1. Руководитель структурного подразделения сообщает о трудностях в выполнении работы, которые обусловлены отсутствием необходимых человеческих ресурсов и просит принять еще одного или нескольких сотрудников.

2. Директор либо подтверждает необходимость устройства нового сотрудника, либо нет, и тогда процесс заканчивается.

3. Если директор подтвердил необходимость устройства нового сотрудника, то зам. начальника КПО составляет список вакансий, в котором указаны и требования к кандидатам на должность.

4. Зам. начальника КПО при участии начальника КПО и руководителя подразделения оценивают кандидата.

5. Если кандидат подходит, то зам. начальника КПО знакомит его с необходимыми документами, если не подходит, то процесс завершается.

6. После того, как кандидат ознакомлен с необходимыми документами, зам. начальника КПО принимает документы для трудоустройства за подписью начальника структурного подразделения и бухгалтера.

7. Начальник КПО проверяет документы на прием, если все верно, то передает директору для заключения трудового договора, если не верно, то возвращает кандидата вновь на стадию приема документов.

8. Директор рассматривает вопрос заключения трудового договора. Здесь может быть три взаимоисключающих сценария развития событий: а) директор отказывает в заключении договора и процесс заканчивается; б) директор дает предписание начальнику КПО еще раз проверить документы на трудоустройство до тех пор, пока директора не устроит их качество; в) директор одобряет заключение трудового договора;

9. После одобрения директором заключения трудового договора руководитель структурного подразделения знакомит нового сотрудника с должностной инструкцией, а начальник КПО готовит приказ о трудоустройстве, юрисконсультант готовит трудовой договор.

10. После того, как юрисконсультант подготовит трудовой договор, зам. начальника ТПО выполняет допуск нового сотрудника к работе. Процесс допуска к работе объединяет в себе две задачи: санэпидемконтроль, который проводит эпидемиолог и вводный инструктаж, который проводит инженер по охране труда.

11. После получения допуска к работе руководитель структурного подразделения составляет индивидуальный план работы на испытательный срок, который длится 3 месяца. На этот срок работнику выделяется наставник.

12. Через 3 месяца руководитель подразделения оценивает работу сотрудника.

13. На основании оценки руководителем структурного подразделения директор принимает решение о продлении трудового договора.

14. Если решение о продлении трудового договора положительное, то юрисконсультант подготавливает дополнительное соглашение, где установлен новый срок договора, в противном случае зам. начальника ТПО возвращает сотруднику его документы и на работу его не устраивают.

Задание 4.

В нотации BPMN описывается несколько типов **Шлюзов**, однако здесь мы привели три наиболее распространённые при описании бизнес-процессов элементы Условия. Самостоятельно изучите типы шлюзов:

1. Основанный на событиях,
2. Параллельный, основанный на событиях,
3. Комплексный

По результатам лабораторной работы оформите отчет, где отобразите выполнение всех заданий. Отчет оформляется в формате doc.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

- 1) Что такое нотация BPMN и каково ее назначение?
- 2) В чем преимущества нотации BPMN перед остальными нотациями моделирования бизнес-процессов?
- 3) Какие основные 4 категории элементов выделяют в BPMN?
- 4) Какие бывают виды сущностей (flows objects) в BPMN?
- 5) Что такое действие (activity)?
- 6) Чем действие отличается от задачи?
- 7) Какие три типа действия используются в BPMN?
- 8) Какие атрибуты может иметь задача в BPMN?
- 9) Что такое шлюз (gateway) и каково его назначение?

- 10) Какие типы шлюзов используются в BPMN?
- 11) Что такое событие (event) и каково его назначение?
- 12) Какие типы событий используются в BPMN?
- 13) В каких случаях используется Исключающий шлюз?
- 14) В каких случаях используется Параллельный шлюз?
- 15) В каких случаях используется Неисключающий шлюз?
- 16) Какая задача называется пользовательской?

ЛАБОРАТОРНАЯ РАБОТА № 7

Зоны ответственности и подпроцессы в нотации BPMN

Цель: изучение основных элементов категорий «Участники» и «Подпроцесс» в нотации BPMN.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

В данной лабораторной работе рассматриваются следующие графические элементы спецификации BPMN и их использование при описании бизнес-процессов: **Пул**, **Дорожка**, а также рассматривается элемент **Подпроцесс**.

2.4. Участники процесса (swimlanes)

Любой описываемый процесс включает в себя участников этого процесса. Для отображения взаимодействия между участниками бизнес-процесса в нотации BPMN используются два типа участников:

- ✓ внешние (pools) - **Пулы**;
- ✓ внутренние (lanes) – **Дорожки**.

Чаще, при описании процессов в BPMN-системах (системах управления бизнес-процессами) в определение элемента **Пул** вкладывают понятие области процесса (совокупность всех действий и ответственных за их выполнение лиц). Пулы не отражают конкретные внутренние процессы участников, они скорее показывают глобальные взаимодействия и зависимости между участниками процесса.

Как правило, элемент **Дорожка** часто используют в качестве внутренних ролей (Зоны Ответственности), что представляет собой распределение обязанностей среди участников процесса (менеджер, директор и т.п.). В области одного **Пула** могут находиться несколько **Дорожек** (участников процесса) (рис. 2.14, 2.15).

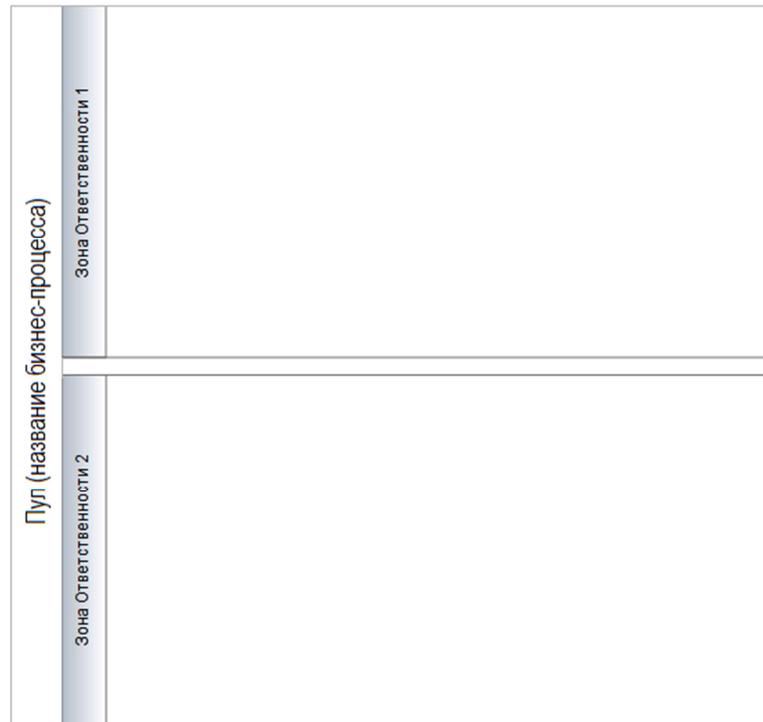


Рис.2.14. Графическое изображение Пула и Дорожек (Зон Ответственности)

Дорожка представляет собой прямоугольник, в котором описываются все действия ответственного за выполнение задач лица. Дорожки в нотации BPMN могут располагаться как вертикально, так горизонтально.

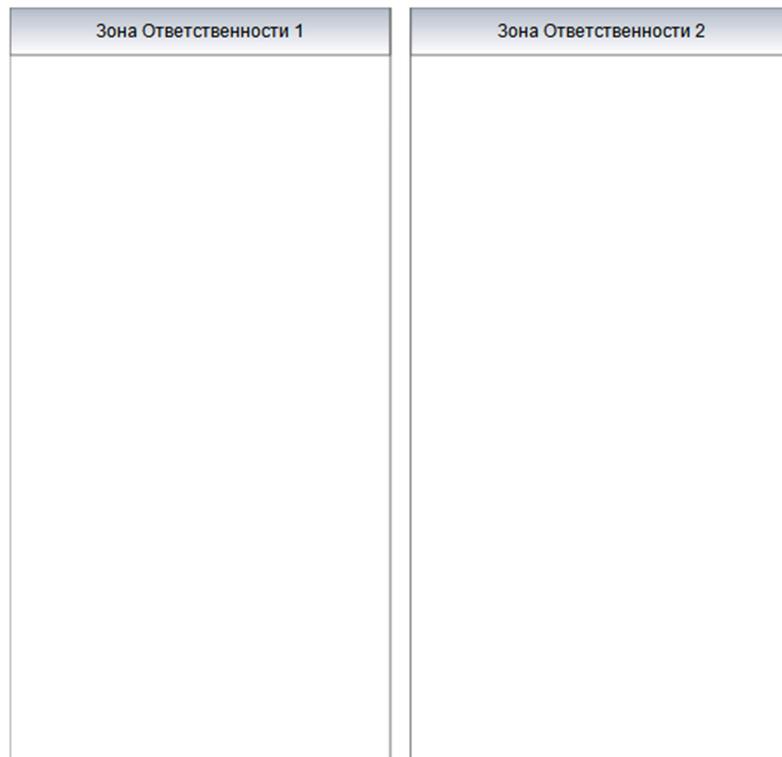


Рис.2.15. Возможное расположение Дорожек в процессе.

В случае если в системах BPMS каждый бизнес-процесс описывается отдельно (на одном листе описывается один процесс), то **Пул**, чаще всего, не визуализируют.

В примере-процессе на рис. 2.16. выделены две **Дорожки** – участники процесса «Офис-менеджер» и «Менеджер по продажам». Офис-менеджер отвечает за регистрацию и обработку заказа. Далее процесс переходит в зону ответственности Менеджера по продажам. Разработчики BPMS для дополнительного удобства сделали возможность задавать зону ответственности динамической, т.е. она не определяет конкретного сотрудника, а лишь показывает роль (должность) ответственного за выполнение задач. В

рамках примера «Менеджер по продажам» задаётся динамической зоной, т.к. менеджеров в компании может быть много, а задачи будут выполняться одним из этих сотрудников.

Дорожка фактически является зоной ответственности участника: любой элемент, помещенный в дорожку, выполняется исполнителем, прописанным в заголовке дорожки. Так, например, на рисунке 2.16 видно, что поток операций переходит из дорожки «Офис-менеджер» в дорожку «Менеджер по продажам». **Задачу**, находящуюся в дорожке «Офис-менеджера» исполняет именно офис-менеджер. В одном процессе может быть неограниченное количество **Дорожек**. Таким образом, можно описать всех участников

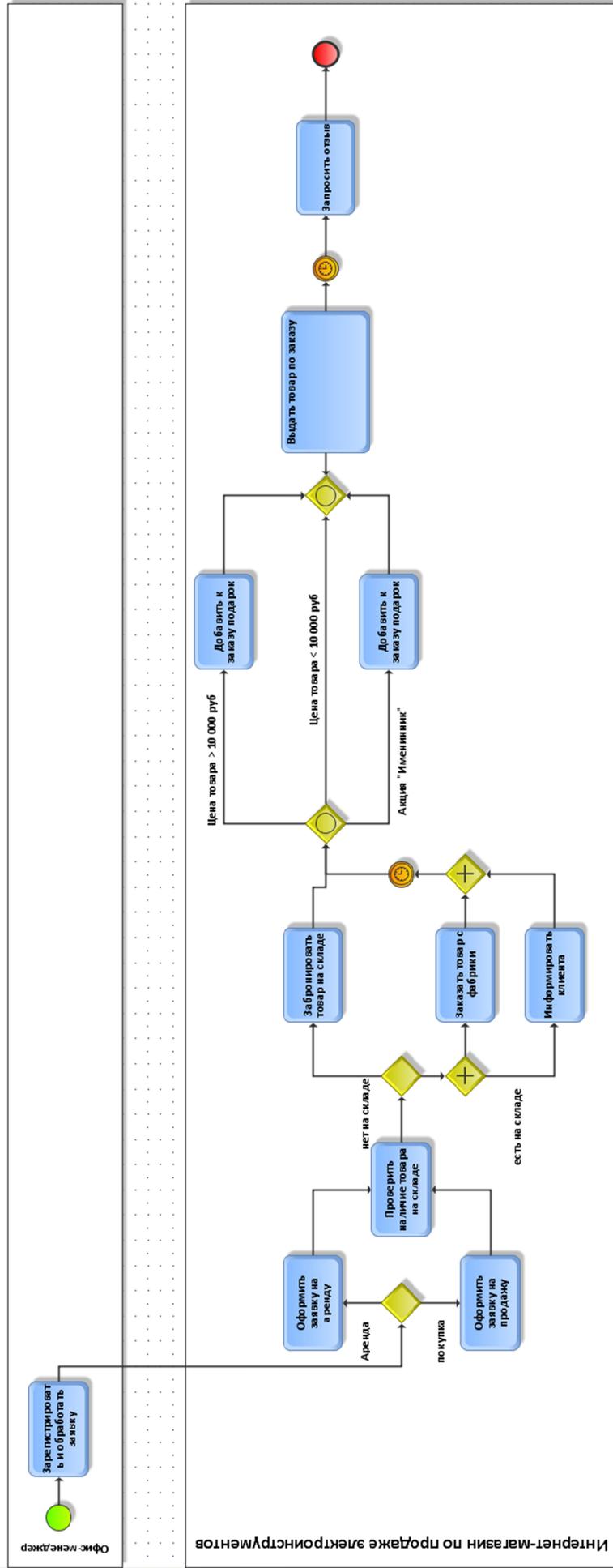


Рис. 2.16. Использование дорожек в процессе «Обработка заказа»

процесса – поток операций будет определять, какие задачи, кем, в какой момент, и в каком порядке будут выполняться в рамках процесса.

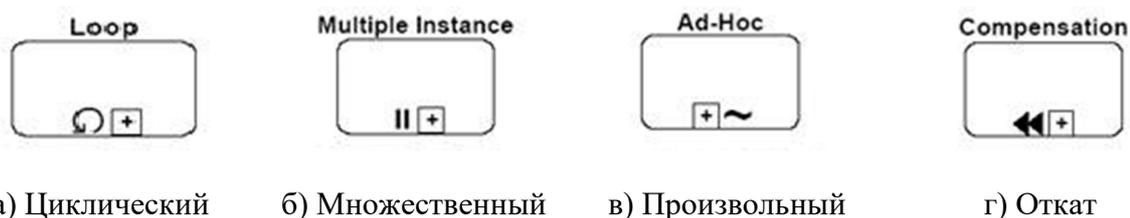
2.5. Подпроцесс (Subprocess)

В предыдущей работе рассматривался элемент **Задача**, как элементарное **Действие** процесса. Еще одним типом действия является **ПОДПРОЦЕСС (subprocess)**. Он позволяет разбить сложные процессы на более мелкие. Графически изображается прямоугольником со скругленными углами и символом «+», означающим, что подпроцесс может быть развернут (рис. 2.17).



Рис.2.17. Простой свернутый подпроцесс

Так же, как и задачи, подпроцессы могут быть циклическими, множественными и с откатом, но кроме того, могут иметь еще маркер *произвольный* (ad hoc) - см. рис. 2.18в. Он означает, что задачи и другие подпроцессы, входящие в состав данного, исполняются в произвольном порядке. Свернутый подпроцесс является ссылкой на другую диаграмму, где он определяется в виде задач и, возможно, других подпроцессов.



а) Циклический б) Множественный в) Произвольный г) Откат

Рис. 2.18. Разновидности подпроцессов в BPMN

Подпроцессы бывают *свернутые* (collapsed subprocesses) - см. рис. 2.18 - и *развернутые* (expanded subprocesses) - см. рис. 2.19. Развернутый подпроцесс позволяет задать на диаграмме второй этаж (а, возможно, третий и т. д. - все зависит от того, насколько модель "глубока"). Это означает, что прямо на родительской диаграмме один или несколько процессов детализированы, как показано на рис. 2.20.

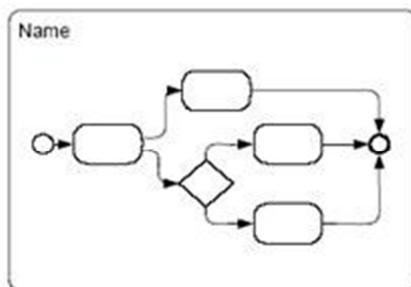


Рис. 2.19. Развернутый подпроцесс

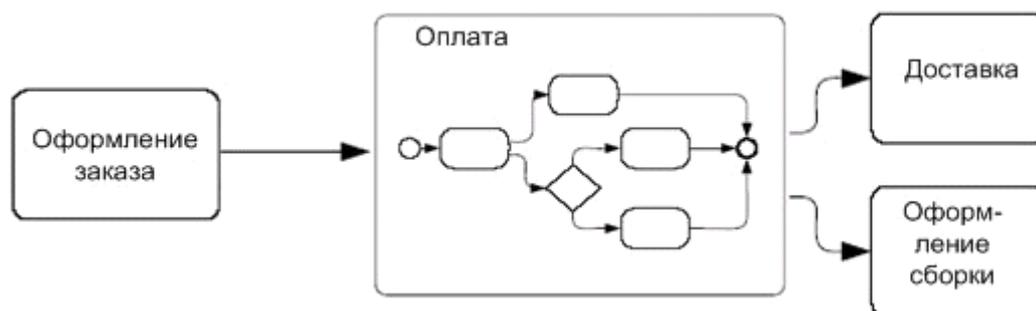


Рис. 2.20. Пример развернутого подпроцесса

Пример использования подпроцесса «Выдать заказ клиенту» показан на рис. 2.21.

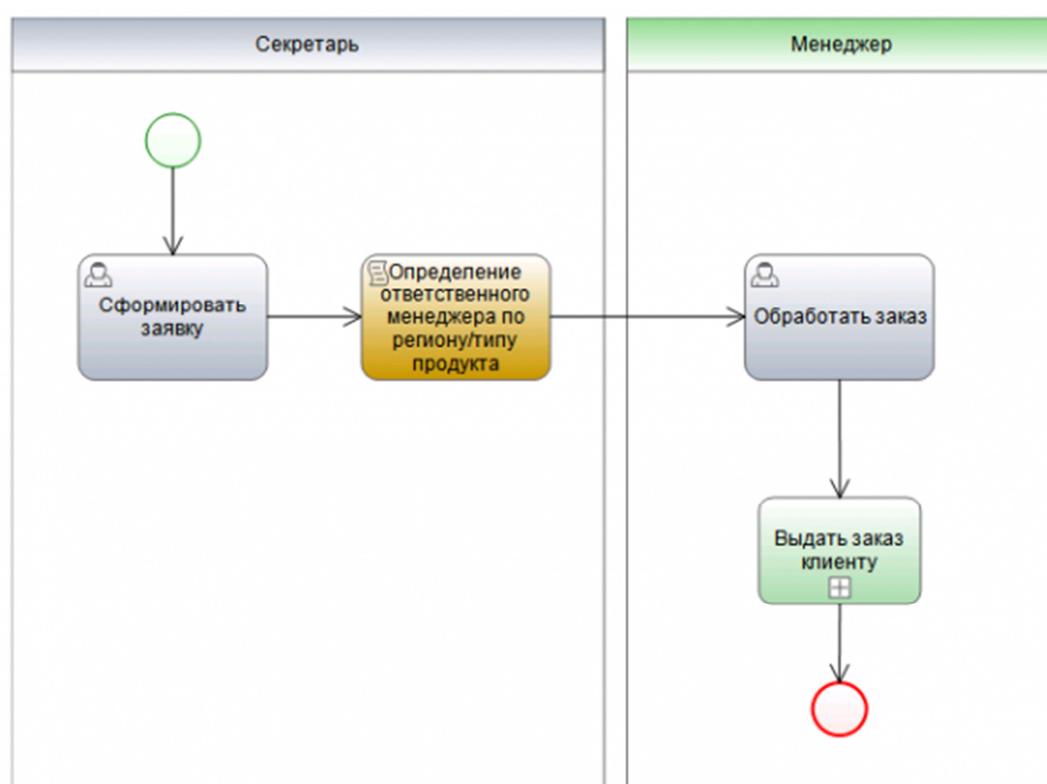


Рис. 2.21. Использование подпроцесса «Выдать заказ клиенту» в описании бизнес-процесса

В практике описания бизнес-процессов элемент нотации BPMN **Подпроцессы** используется в основном в двух случаях [8]:

1. Для декомпозиции и повышения читаемости и наглядности схем (диаграмм);
2. Для описания повторяющихся действий. Единожды описанный **Подпроцесс** может многократно вызываться (использоваться) внутри различных процессов.

Рассмотрим первый случай использования **Подпроцессов** – Декомпозиция процесса. Довольно часто при описании бизнес-процессов компании для наглядности

используют схемы (диаграммы), отражающие верхние уровни организации работы. В этом случае диаграмма отображает «суть» процессов и нацелена на понимание логики процесса без знания деталей. Примером такого бизнес-процесса верхнего уровня может служить – процесс «Найм персонала». На верхнем уровне этот процесс будет выглядеть следующим образом (рис. 2.22):

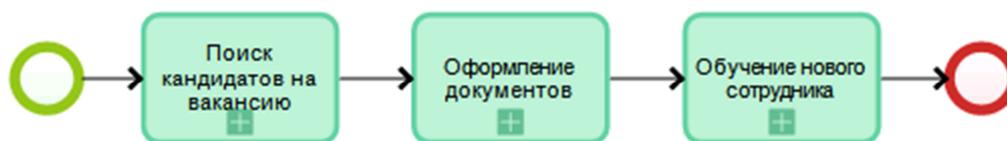


Рис. 2.22. Процесс верхнего уровня «Найм персонала».

Такая «прорисовка» процесса легка для восприятия любого бизнес-пользователя, т.к. отображает только последовательность основных действий в рамках процесса без утяжеления информацией. Любая схема (диаграмма) процесса представляет собой последовательность функциональных блоков, декомпозиция которых позволяет создать процесс верхнего уровня. При этом каждый **Подпроцесс** описывается уже на более низком уровне с полной детализацией элементов BPMN (действия, условий и исполнителей). **Подпроцессы** являются комплексными задачами в рамках основного процесса. Однако стоит отметить, что **Подпроцессы**, как элемент BPMN, являются не самостоятельными задачами, а лишь ссылкой к другому процессу.

Декомпозиция процесса (разбивка на подпроцессы) позволяет моделировать и вносить изменения в рамках каждого **Подпроцесса**, не изменяя весь основной процесс целиком. При детализации каждого отдельного **Подпроцесса** описываются необходимые условия выполнения: участники, действия, бизнес-правила и т.д.

При детализации **Подпроцессов** приведённого примера процесса «Найм персонала» получим следующие процессы:

1. Поиск кандидатов на вакансию.
2. Оформление документов нового сотрудника.
3. Обучение нового сотрудника.

Рассмотрим каждый **Подпроцесс** отдельно.

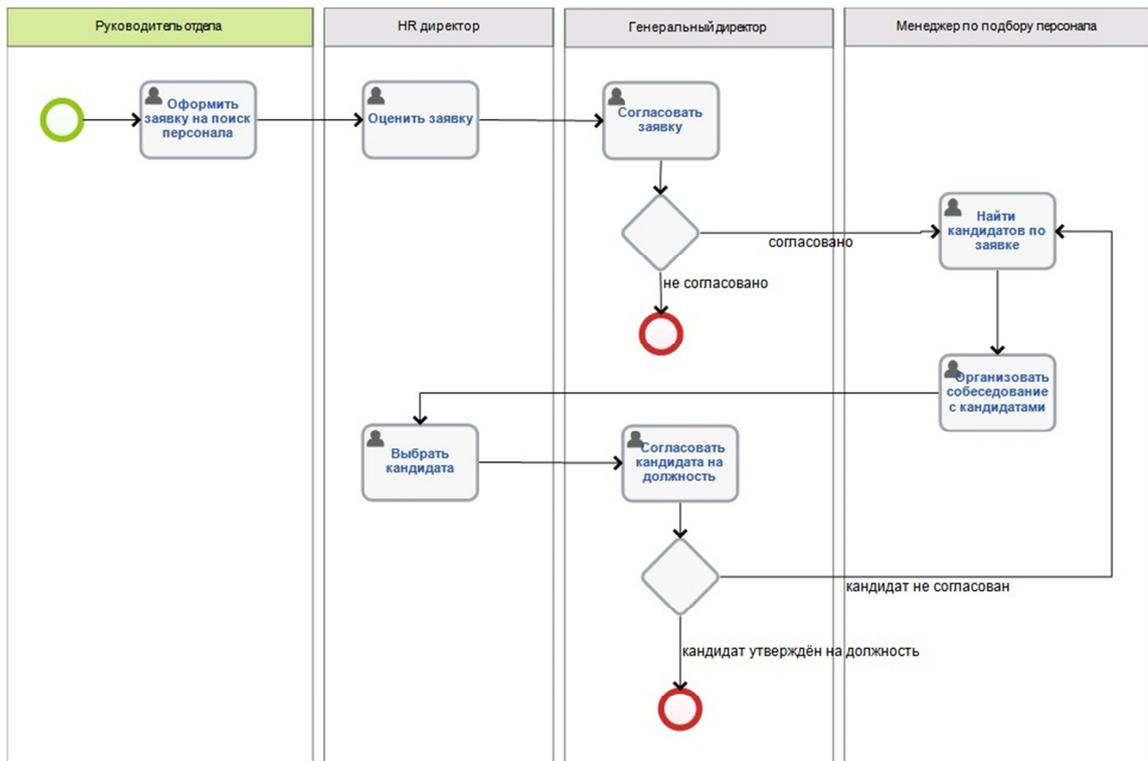


Рис.2.23. Подпроцесс «Поиск кандидатов на вакансию».

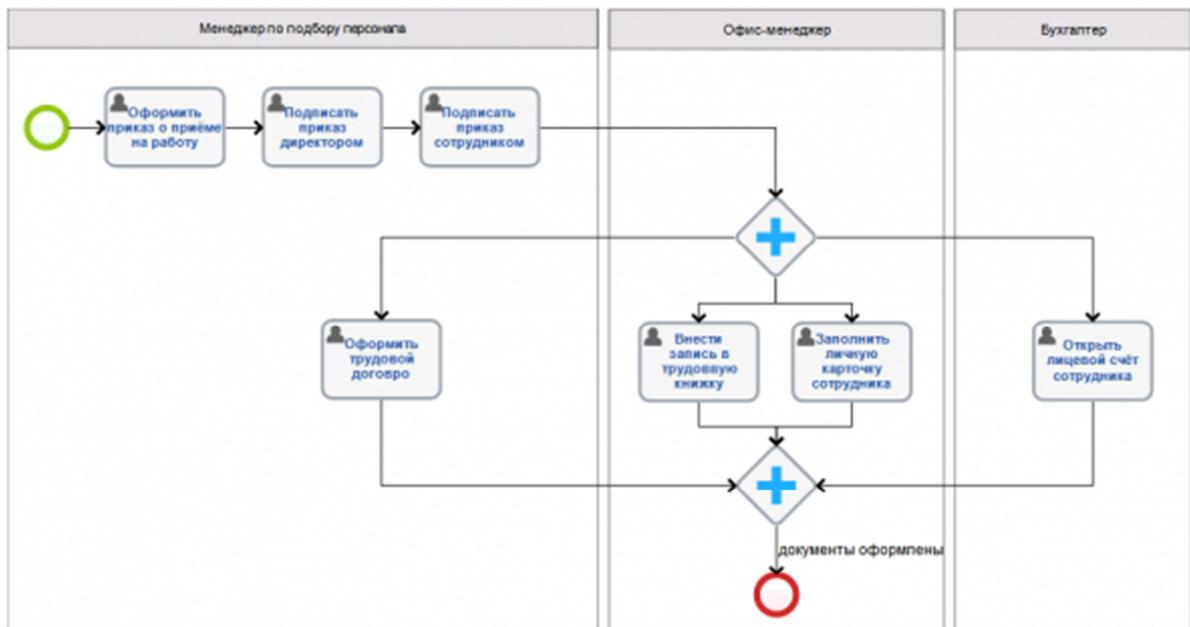


Рис.2.24. Подпроцесс «Оформление документов».

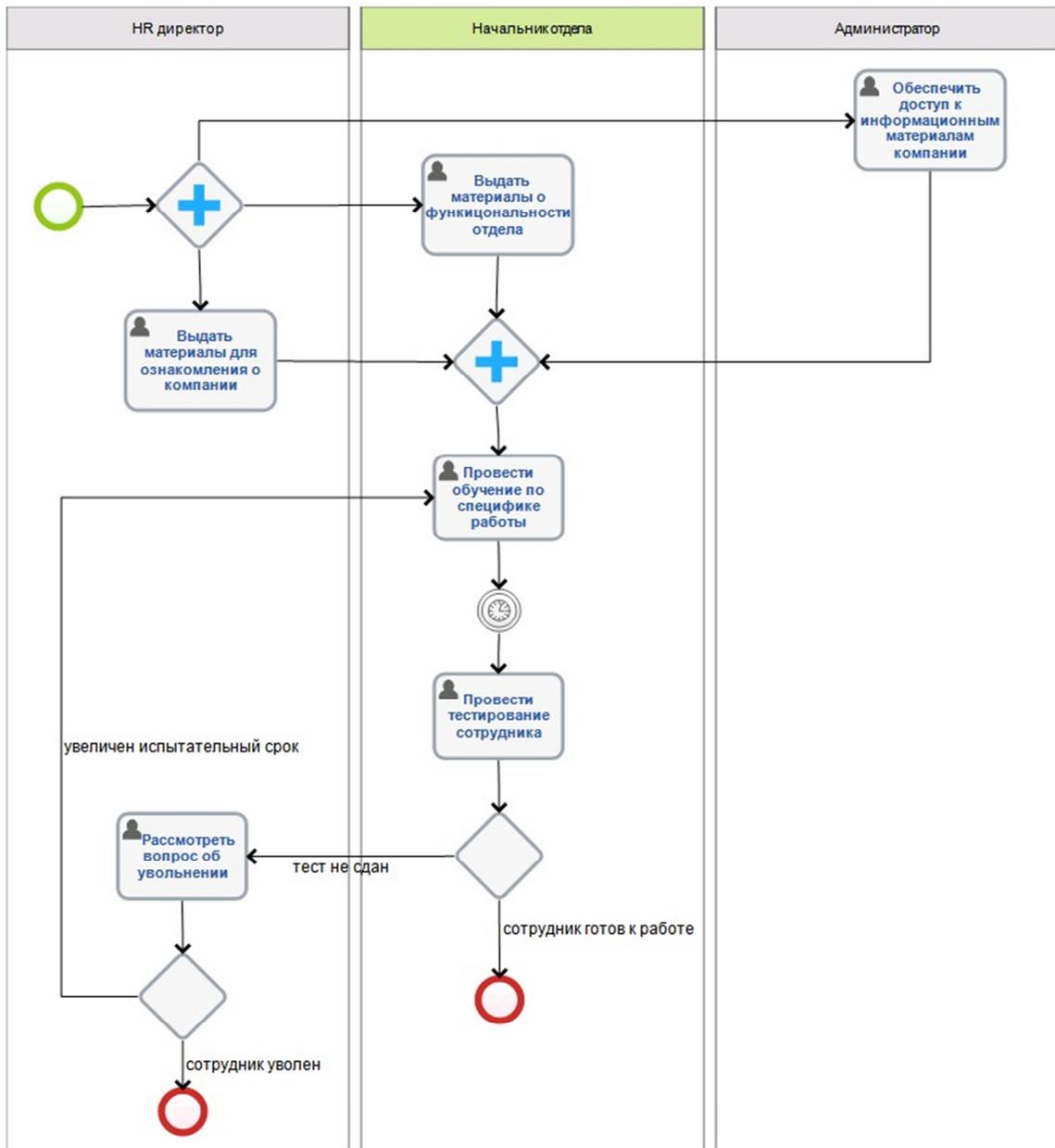


Рис. 2.25. Подпроцесс «Обучение нового сотрудника».

Вот таким образом можно описать довольно большой бизнес-процесс компании. А теперь представьте, если все активности и исполнители процесса «Найм персонала» будут отображены в рамках одного процесса. Сделать это сложно, а «читать» процесс будет ещё сложнее. Поэтому, используя декомпозицию (разбивку на подпроцессы) при описании сложных, но важных процессов компании, вы получаете продукт (процесс), который будет понятен любому бизнес-пользователю и легко изменяемый при моделировании и совершенствовании в будущем.

Ещё одним большим плюсом при использовании Подпроцессов является возможность их повторного использования. В рамках одного основного процесса могут повторяться одни и те же действия. **Подпроцессы** позволяют ссылаться на один и тот же

Подпроцесс (функциональный блок) сколько угодно раз в одном бизнес-процессе и в абсолютно разных по сути процессах.

При внесении изменений в **Подпроцесс** нет необходимости перерисовывать все процессы, ссылающиеся на данный **Подпроцесс**. Сопровождением и актуализацией **Подпроцесса** занимаемся только его владелец, что позволяет сократить время внесения изменений в процессы, снизить риск ошибок и иметь постоянно актуальные решения. **Изменения вносятся в одном месте и один раз!**

Повторно-используемый **Подпроцесс** используется для вызова предопределенного **Подпроцесса**. Примером повторно-используемого **Подпроцесса** может служить процесс «Информирования контрагентов» в рамках основных процессов «Выпуск нового продукта» и «Открытие нового филиала».

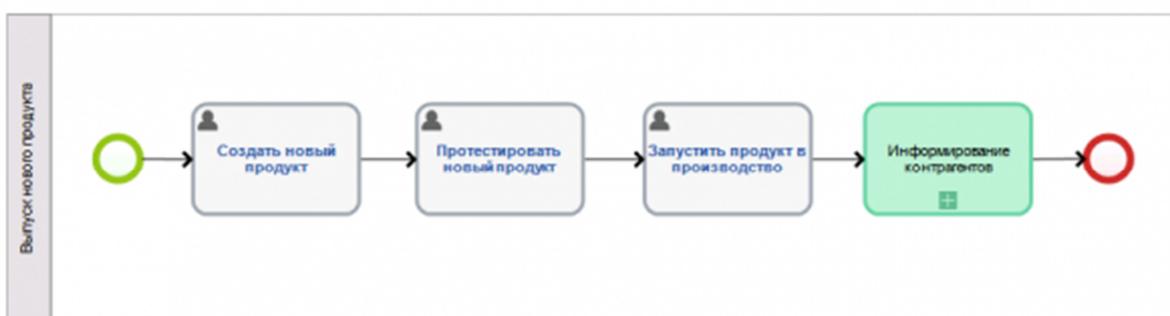


Рис. 2.26. Пример использования повторно-используемого Подпроцесса.



Рис. 2.27. Пример использования повторно-используемого Подпроцесса.



Рис. 2.28. Повторно-используемый Подпроцесс.

При появлении нового информационного канала или механизма рекламирования в Подпроцессе «Информирование контрагентов» изменения вносятся один раз и только в данный Подпроцесс, не затрагивая основные процессы компании.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1.

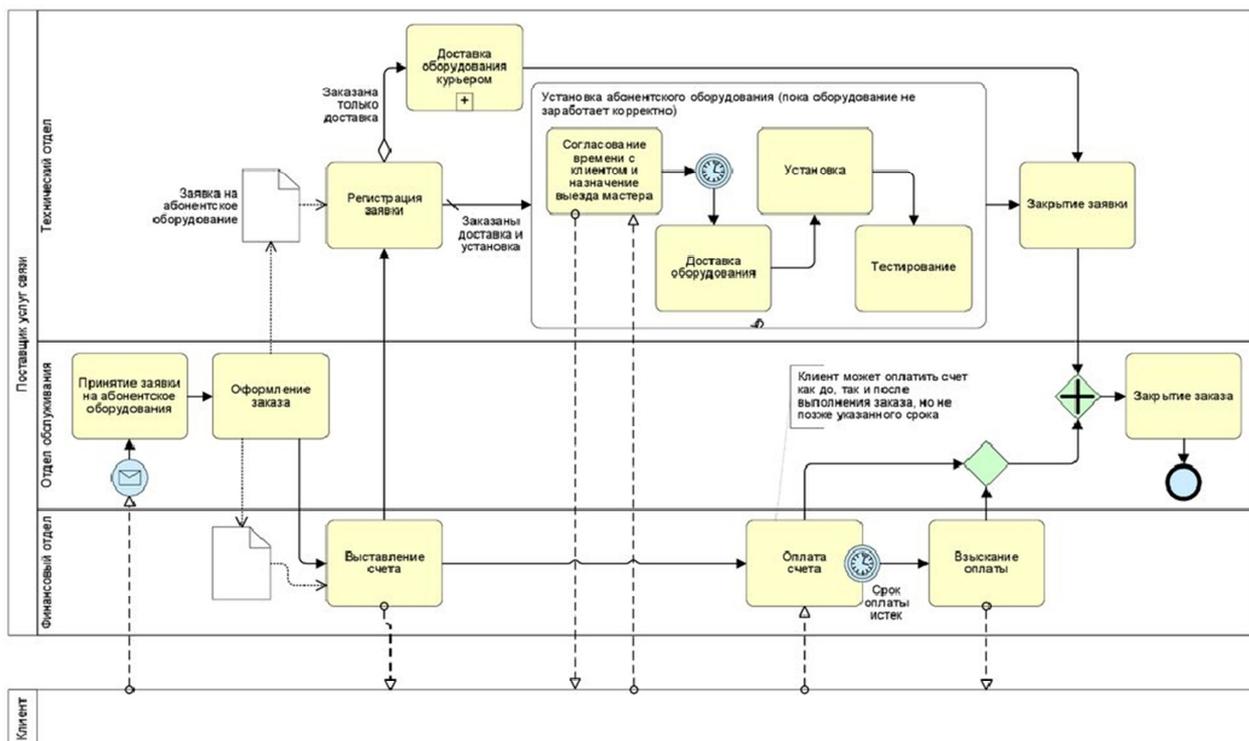
При помощи программы Tibco Business Studio разработать процесс верхнего уровня «Найм персонала» (рис. 2.23).

Задание 2.

Разработать подпроцессы «Поиск кандидатов на вакансию», «Оформление документов», «Обучение нового сотрудника».

Задание 3.

Разработать процесс установки абонентского оборудования



Задание 4.

Разработать модель бизнес-процесса по бронированию авиабилетов и гостиницы для туристов в деятельности турфирмы. При этом рассмотреть 4-х участников процесса: туриста, турфирму, авиакассу и гостиницу. При разработке использовать элементы «Свернутый подпроцесс», «Развернутый подпроцесс», «Циклический подпроцесс», «Откат», «Дорожка». Использование остальных элементов остается на усмотрение студента.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

1. Какие два типа участников используются в нотации BPMN?
2. Чем элемент Пул отличается от элемента Дорожка?
3. Что такое подпроцесс в BPMN?
4. Как изображается простой свернутый подпроцесс?
5. Какие типы подпроцессов существуют в BPMN?
6. Для чего используются подпроцессы в BPMN?
7. Как изображается циклический подпроцесс?
8. Как изображается множественный подпроцесс?
9. Как изображается произвольный подпроцесс?
10. Как изображается подпроцесс отката?
11. В чем преимущества декомпозиция процессов в BPMN?

ЛАБОРАТОРНАЯ РАБОТА № 8

Моделирование процесса пуско-наладочных работ в нотации BPMN

Цель: изучение основных элементов категорий «Связи» и «Артефакты» в нотации BPMN, разработка основного бизнес-процесса пуско-наладочных работ в нотации BPMN.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.6. Связи

На рис. 2.29. показаны связи разного вида, существующие в BPMN:

- ✓ **поток исполнения** (sequence flow) - это последовательность действий, обозначается как стрелка, и показывает, какое действие после какого необходимо совершить (рис. 2.29а). Является наиболее распространенным типом связи в BPMN;
- ✓ **поток сообщений** (message flow) - это пунктирные стрелки в бизнес-модели, показывающие сообщения, которыми обмениваются участники бизнес-процесса (рис. 2.29 б). Обмениваться сообщениями могут: конструкции pools друг с другом, задачи, подпроцессы и т.д.;. Сообщения являются способом общения между собой параллельно работающих сущностей, поэтому сущности могут обмениваться сообщениями, лишь находясь в разных pools. Например, если заказ переходит от клиента в обработку в отдел продаж, он сопровождается сообщением, которое содержит информацию об этом заказе.
- ✓ **ассоциации** (association) - это способ отобразить различные вспомогательные связи в модели бизнес-процессов; на рис. 2.29 в представлена ассоциация отката; на рис. 2.29г показана ассоциация исключения;

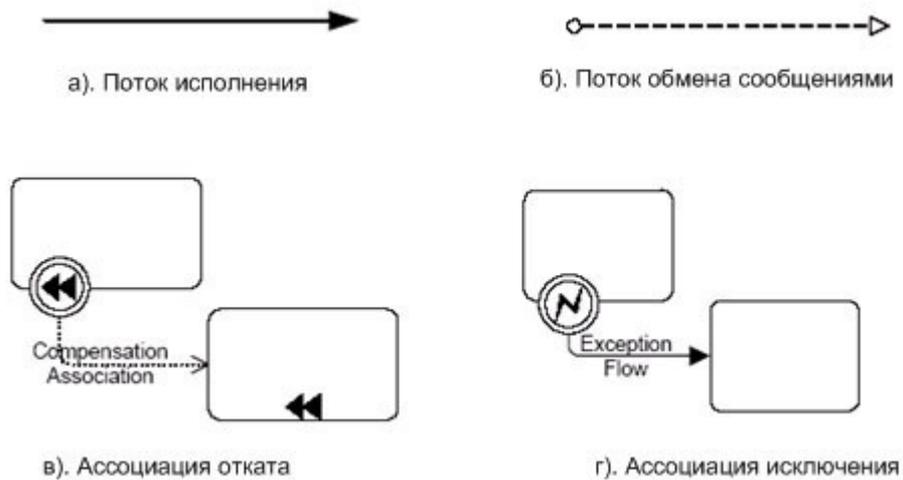


Рис. 2.29. Виды связей

2.7. Артефакты

Под артефактами в BPMN понимают объекты, не являющиеся действиями и не связанные с действиями напрямую. Это могут быть любые документы, данные, информация, которая не влияет напрямую на исполнение процесса.

Выделяют два вида артефактов:

- ✓ Группа объектов (Object Group);
- ✓ Текстовая аннотация (Text Annotation).
- ✓ Данные (Data Objects).

Группа объектов – это еще одна возможность объединить под общим символом несколько элементов, чтобы сэкономить место на диаграмме и повысить простоту ее восприятия. Здесь собираются различные активности под одним общим названием. Группу объектов также всегда можно рассмотреть детально. Группа выглядит как прямоугольник с закругленными углами, выполненный штриховой линией с точками (рис. 2.30 а).

Текстовая аннотация применяется для различных уточнений к диаграмме. Это могут быть комментарии, пояснения, другая информация, которая повысит читабельность диаграммы. Аннотации – это незакрытый прямоугольник, выполненный сплошной линией, от которого к объекту аннотации ведет линия, состоящая из точек (рис. 2.30 б)..

Данные показывают читателю, какие данные необходимы действиям для выполнения и какие данные действия производят. Данные отображаются так, как показано на рисунке 2.30 в.

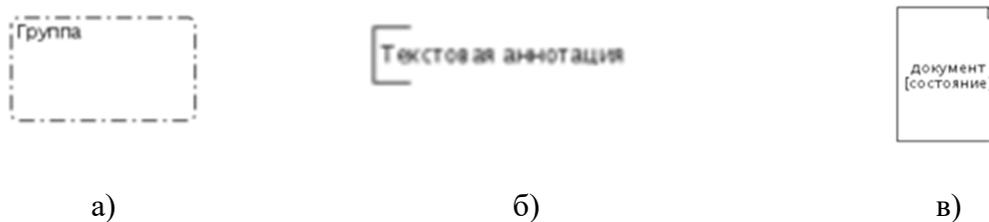


Рис. 2.30. Изображение артефактов в нотации BPMN

Артефакты позволяют разработчикам отображать дополнительную информацию в диаграмме. Это делает диаграмму более удобочитаемой и насыщенной информацией. Пример применения артефактов в процессе сдачи экзамена показан на рис. 2.31.

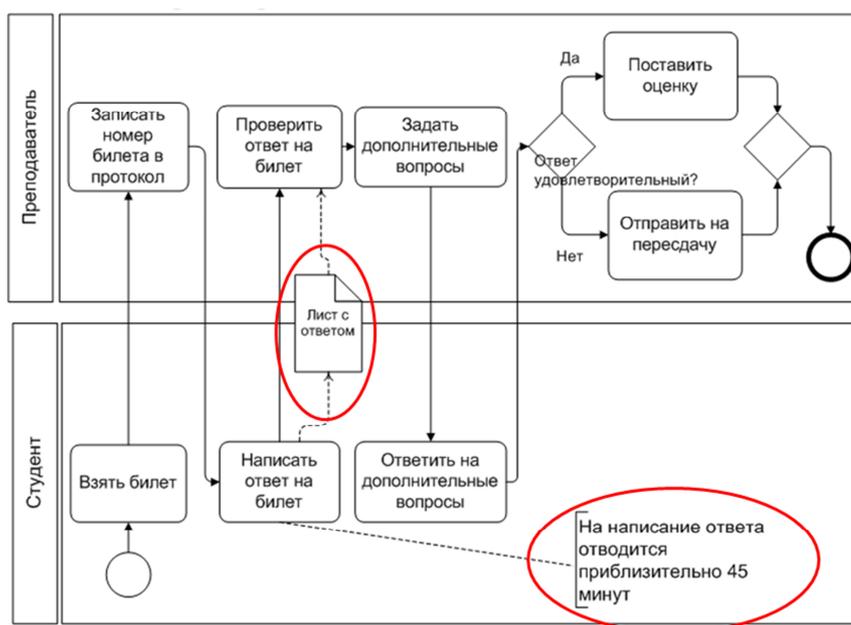


Рис. 2.31. Применение артефактов типа «Данные» и «Аннотация» в BPMN-модели процесса сдачи экзамена

2.8. Исполняемые и неисполняемые бизнес-процессы

В бизнес-моделировании процессы можно условно разделить на два вида — исполняемые, которые действительно будут работать при помощи специального обеспечения, например, Bizagi, Elma, ARIS Toolset и неисполняемые, т.е. бизнес-модели, необходимые только для изучения и демонстрации вариантов работы предприятия [9].

В принципе, между их построением нет особой разницы, здесь важен исключительно желаемый результат. Либо бизнес-модель будет применяться только для облегчения взаимопонимания между заказчиком (руководителем) и консультантом (исполнителем). Либо эта нотация будет впоследствии использоваться в какой-либо

программной среде для организации работы компании. В обычных руководствах такого разделения на две части не делается. Но в среде разработчиков принято считать, что имеет смысл условно так делить бизнес-процессы, так как при различном желаемом результате потребуется различная глубина проработки деталей и выбор возможных инструментов для работы.

Исполняемые бизнес-процессы обязательно должны быть выстроены в строгом соответствии всем правилам нотации BPMN, так как в противном случае программное обеспечение не сможет работать корректно с составленной бизнес-моделью. Исполняемые процессы нужны, например, на предприятиях, где принят процессный подход к деятельности. Программное обеспечение позволяет вести контроль всех процессов в режиме реального времени, и на основе получаемых на каждом из этапов данных, руководитель компании и подразделений всегда смогут понимать, на каком этапе находится работа по тому или иному процессу. Подобный метод позволяет значительно повысить эффективность управления.

Неисполняемые бизнес-процессы нужны исключительно для демонстрации какой-либо бизнес-модели. Это может быть диаграмма, отображающая реальное положение дел на предприятии, может быть наглядной иллюстрацией к предложенным изменениям при реинжиниринге. В этом случае, конечно, можно использовать любые удобные инструменты, в том числе, традиционный для многих IDEF0. А соблюдение правил языка моделирование необходимо исключительно для достижения взаимопонимания.

На начальном этапе работы с BPMN полезно научиться создавать неисполняемые бизнес-процессы. Это действительно очень удобная нотация для того, чтобы иллюстрировать свои идеи и предложения, демонстрировать «узкие места» в бизнесе, даже просто для себя разбираться в структуре работы той или иной компании очень удобно с использованием нотаций. Наглядная графика и строгие правила в этом очень помогают.

Исполняемый вариант требует глубоких знаний BPMN, а также внимательного отношения к каждой детали, так как вы, по сути, создаете программу (алгоритм) для компьютера, просто используете для этого не текстовый язык, а графические нотации. Это дело – для опытных специалистов.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1. Изучение и моделирование процесса пуско-наладочных работ компании «ЭлеСи-Про»

1. Изучите описание пункта «Подготовка к моделированию бизнес-процесса пуско-наладочных работ» (далее ПНР);
2. Изучите карту процесса;
3. Изучите текстовое описание процесса ПНР в компании «ЭлеСи-Про»
4. Разработайте BPMN-модель процесса пусконаладочных работ, показанную на рис. 2.32. Описание процесса приведено ниже.

«Подготовка к моделированию бизнес-процесса пуско-наладочных работ»

Для детального и полного понимания данного процесса нужно было выполнить следующие задачи:

1. Изучить основные документированные процедуры по процессу «Создание АСУ ТП ДП 2.07-03»;
2. Изучить основные документированные процедуры по процессу: «Проведение пуско-наладочных работ АСУ ТП РГ 023»;
3. Провести интервью с начальником отдела внедрения ПТК (программно-технических комплексов);
4. На основании полученной информации составить карту процесса ПНР.

Описание процесса пуско-наладочных работ в компании «Элеси-Про»

Процесс пуско-наладочных работ осуществляется при вводе в эксплуатацию или проведении технического обслуживания автоматизированных систем управления технологическими процессами (АСУ ТП) на объектах транспорта нефти и нефтепродуктов.

Данный процесс включает в себя следующие подпроцессы:

- ✓ шеф-монтажные работы;
- ✓ автономную наладку;
- ✓ комплексную наладку;
- ✓ сдачу в опытную эксплуатацию;
- ✓ сдачу в промышленную эксплуатацию.

Процесс начинается в соответствии с календарным планом-графиком выполнения работ (КПВР), после завершения строительно-монтажных работ монтажной организацией, и, в случае готовности объекта, по данному графику происходит подготовка к выезду и непосредственный выезд сотрудников отдела ОВ ПТК на объект для проведения в полной мере пуско-наладочных работ. По прибытию сотрудников на объект, в случае необходимости, возможно проведение шеф-монтажных работ. По их завершению происходит автономная и комплексная наладка АСУ ТП. После чего на данном этапе

прохождения процесса ПНР идет обучение сотрудников заказчика и происходит сдача системы в опытную эксплуатацию. Если при проверке не было обнаружено никаких дефектов или замечаний, то система сдается в промышленную эксплуатацию. При наличии положительного ответа заказчика на сдачу АСУ ТП, сотрудниками ОВ ПТК происходит передача ПО и конфигурации системы заказчику. Результатом проведения ПНР является введенная в промышленную эксплуатацию АСУ ТП. После подписания всех соответствующих документов и с разрешения начальника ОВ ПТК, сотрудники отдела возвращаются с объекта заказчика и оформляют финансовую отчетность, на чем и заканчивается процесс пуско-наладочных работ.

Практически каждый этап процесса сопровождается документооборотом. Документы и иные исходы процесса обозначены на рис.2.32.

После разработки основного процесса пуско-наладочных работ, представленного на рис. 2.32 необходимо провести детальное моделирование всех его подпроцессов.

Таблица 2.1. Карта процесса ПНР

1. Входы и выходы.	Вход: Копия письма Заказчика с вызовом на ПНР. Указания, полученные от ОУП. Отгруженная продукция. ПСД, НТД. Выход: Акты выполненных работ. Введенная в промышленную эксплуатацию АСУ ТП.
2. Начальная точка процесса.	Письмо-заявка от менеджера проекта. Доставка АСУ ТП Заказчику, завершение строительно-монтажных работ монтажной организацией.
3. Участники процессов и подпроцессов.	1) ОВ ПТК: ШМР, ВДС 2) ОТО
4. Владелец и ответственный процесса.	Ответственный за ведение процесса: начальник ОВ ПТК.
5. Поставщик и потребители процесса.	Поставщик: ОУП и заказчик ПНР. Потребитель: ОУП.
6. Требования к поставщикам процесса.	Соответствие КПВР и качеству работ.
7. Структура процесса.	<ol style="list-style-type: none"> 1. Начало процесса: Окончание монтажных работ. 2. Шеф-монтажные работы. 3. Приемка объекта к проведению ПНР. <ul style="list-style-type: none"> • Приведение в работоспособное состояние всю регулируемую и запорную арматуру, на которой смонтированы исполнительные механизмы систем автоматизации; • Введение в действие системы автоматического пожаротушения и сигнализации: <ul style="list-style-type: none"> – мониторинг смонтированного на объекте

	<p>оборудования АСУ ТП;</p> <ul style="list-style-type: none"> – мониторинг функциональных и принципиальных связей технических средств и приборов автоматизации, проверка соответствия назначений сигналов, клеммных полей щитов УСО, автоматики ЗРУ и другого оборудования, входящего в состав проекта; – проведение контроля выполнения работ по устранению замечаний организацией выполняющей монтаж. <ul style="list-style-type: none"> • Автономная наладка. • Комплексная наладка. <ul style="list-style-type: none"> – проверка выполнения функций автоматизированной системы во всех режимах функционирования, установленных в техническом задании и нормативной документации Заказчика; – проверка реакции системы на некорректную информацию и аварийные ситуации. • Сдача АСУ ТП в эксплуатацию: • <p>а) Приемка АСУ ТП в опытную эксплуатацию.</p> <ul style="list-style-type: none"> ○ Проведение испытаний на объекте. ○ Установка системы на тестовый прогон продолжительностью 72 часа с непрерывным контролем корректности функционирования. <ul style="list-style-type: none"> – соответствие объекта и смонтированного оборудования проекту; – соответствие выполненных строительно-монтажных работ и пуско-наладочных работ требованиям настоящего регламента и нормативных документов, состав которых приведен в разделе "Общие положения"; – подготовленность АСУ ТП к опытной эксплуатации, включая мероприятия по обеспечению на них условий труда в соответствии с требованиями техники безопасности и производственной санитарии, защите природной среды и действующего законодательства РФ. ○ Проведение первичного обучения персонала Заказчика (назначение АСУ ТП, ее основные функции и возможности, интерфейс пользователя). ○ Обучение персонала использованию средств настройки, конфигурирования и диагностики АСУ ТП. ○ <p>б) Приемка АСУ ТП в промышленную эксплуатацию.</p> <ul style="list-style-type: none"> – проводится после проведения опытной эксплуатации и последующего устранения замечаний, выявленных в процессе опытной эксплуатации. <ul style="list-style-type: none"> • Окончание процесса: передача Заказчику
--	---

	оборудования и программного обеспечение согласно комплекту поставки.
8. Времена и диаграммы.	<ul style="list-style-type: none"> • Согласно календарному плану • По вызову из ОУП
9. Записи процесса.	<ul style="list-style-type: none"> • Акт приемки монтажа. • Акт сдачи в ОЭ. Протоколы испытаний. • Акт сдачи в ПЭ.
10. Смежные процессы.	<ul style="list-style-type: none"> • Взаимодействие с Заказчиком. • Процесс отгрузки. • Строительно-монтажные работы.
11. Контроль процессов.	ОУП, начальник отдела ОВ ПТК
12. Документооборот.	<p>Вход: Данные от менеджера проекта.</p> <ul style="list-style-type: none"> • План-график проведения пуско-наладочных работ. • Календарный план выполнения работ. • ТЗ на АСУ ТП; • Согласованная с Заказчиком ПМИ; • Рабочая проектная документация; • Конструкторская документация на АСУ ТП; • ЭД на АСУ ТП, входящее в ее состав оборудование; • Технологические регламенты объектов автоматизации; • Проект производства работ (Пакет документов, согласованный с представителями Заказчика). • Рабочая проектная документация, переданная заказчиком, включая части проекта АСУ ТП: <ul style="list-style-type: none"> - математическое обеспечение (МО), - информационное обеспечение (ИО), - программное обеспечение (ПО), - организационное обеспечение (ОО). • Письменное подтверждение о готовности объекта от заказчика. • Письмо о направлении специалистов на объект. <p>Выход:</p> <ul style="list-style-type: none"> • Технический акт Ф01 РГ23 • Акт приема-передачи Ф02 РГ23 • Акт выполненных работ Ф03 РГ23 • Акт приемки в эксплуатацию систем автоматизации Ф04 РГ23 • Исполнительные документы об окончании монтажных работ. • Занесение несоответствий монтажа на соответствие проекту. • Оформление акта приемки смонтированных систем автоматизации. • Оформление актов обнаружения неисправности оборудования и далее актов приема-передачи материальных ценностей.

	<ul style="list-style-type: none"> • Ведение "Журнала проведения ПНР» (ход проведения работ, регистрация всех проблем, возникающих в процессе проведения ПНР). • Документальное оформление необходимых инструктажей у Заказчика (инструктажа на рабочем месте, инструктажей по охране труда, пожарной безопасности и прочих, принятых у конкретного Заказчика). • Документальное оформление проводимых работ на данном объекте в порядке, установленном заказчиком.
13. Валидация процесса.	Валидация всего комплекса проведенных работ.

Задание 2. Моделирование подпроцесса «Подготовка к выезду»

1. Разверните на основной диаграмме подпроцесс «Подготовка к выезду»;
2. Разработайте модель подпроцесса «Подготовка к выезду» (рис. 2.33).
3. Создайте текстовое описание подпроцесса.

Задание 3. Моделирование подпроцесса «Шеф-монтажные работы»

1. Разверните на основной диаграмме подпроцесс «Шеф-монтажные работы»;
2. Разработайте модель подпроцесса «Шеф монтажные работы» (рис. 2.34).
3. Создайте текстовое описание подпроцесса.

Задание 4. Моделирование подпроцесса «Автономная наладка»

1. Разверните на основной диаграмме подпроцесс «Автономная наладка»;
2. Разработайте модель подпроцесса «Автономная наладка» (рис. 2.35).
3. Создайте текстовое описание подпроцесса.

Задание 5. Моделирование подпроцесса «Комплексная наладка»

1. Разверните на основной диаграмме подпроцесс «Комплексная наладка»;
2. Разработайте модель подпроцесса «Комплексная наладка» (рис. 2.36).
3. Создайте текстовое описание подпроцесса.

Задание 6. Моделирование подпроцесса «Сдача АСУ ТП в опытную эксплуатацию»

1. Разверните на основной диаграмме подпроцесс «Сдача АСУ ТП в опытную эксплуатацию».
2. Разработайте модель подпроцесса «Сдача АСУ ТП в опытную эксплуатацию» (рис. 2.37).
3. Создайте текстовое описание подпроцесса.

Задание 7. Моделирование подпроцесса «Сдача АСУ ТП в промышленную эксплуатацию»

1. Разверните на основной диаграмме подпроцесс «Сдача АСУ ТП в промышленную эксплуатацию»;
2. Разработайте модель подпроцесса «Сдача АСУ ТП в промышленную эксплуатацию» (рис. 2.38).
3. Создайте текстовое описание подпроцесса.

Задание 8. Сформируйте итоговый отчет по лабораторной работе, где должны быть все разработанные модели и их текстовые описания. Оформите титульный лист к лабораторным работам по образцу. Образец титульного листа оформления отчета представлен в приложении А к настоящим методическим рекомендациям.

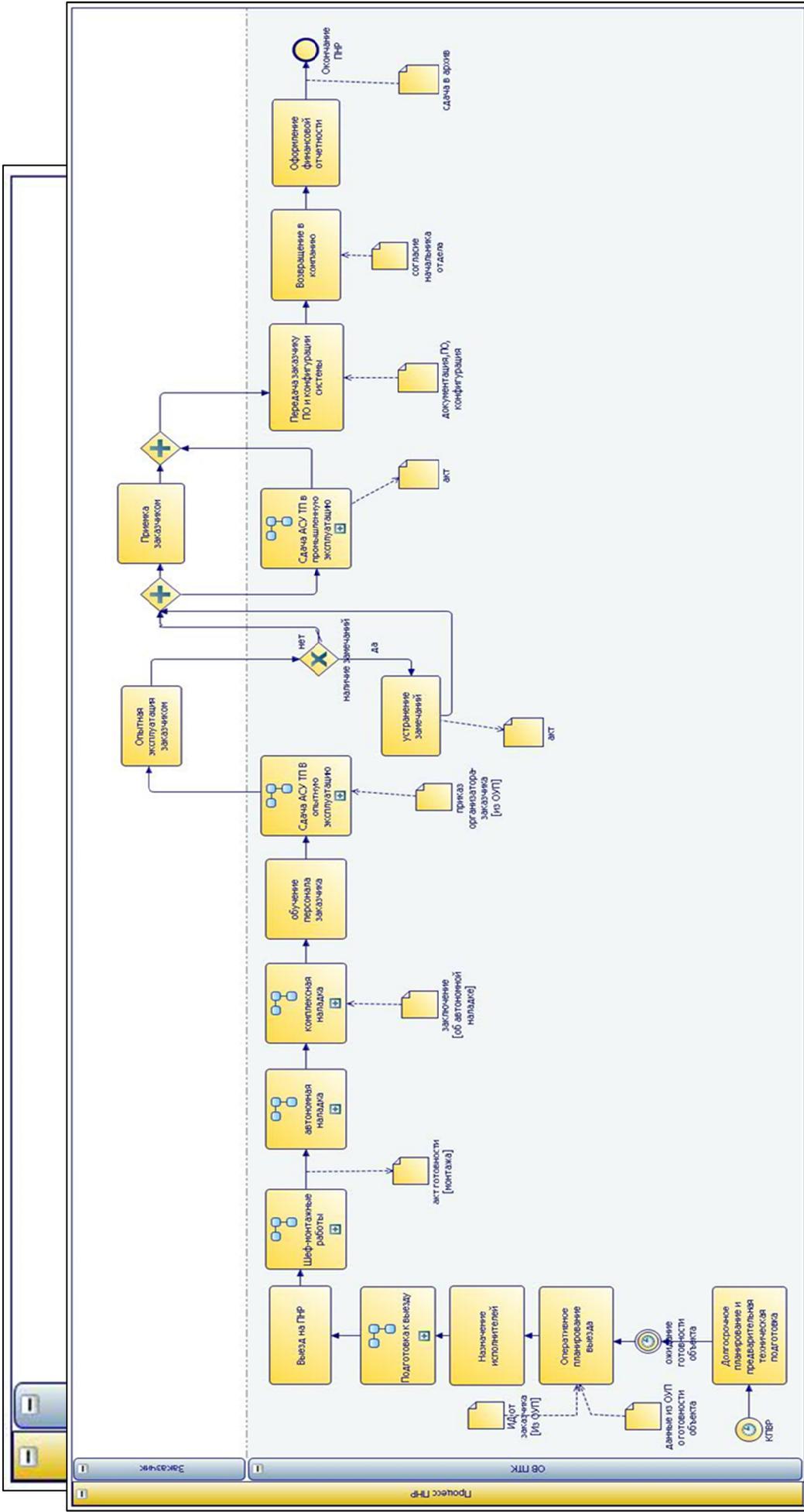


Рис. 2.32. Описание процесса пуско-наладочных работ

Рис. 2.33. Описание процесса «Подготовка к выезду»

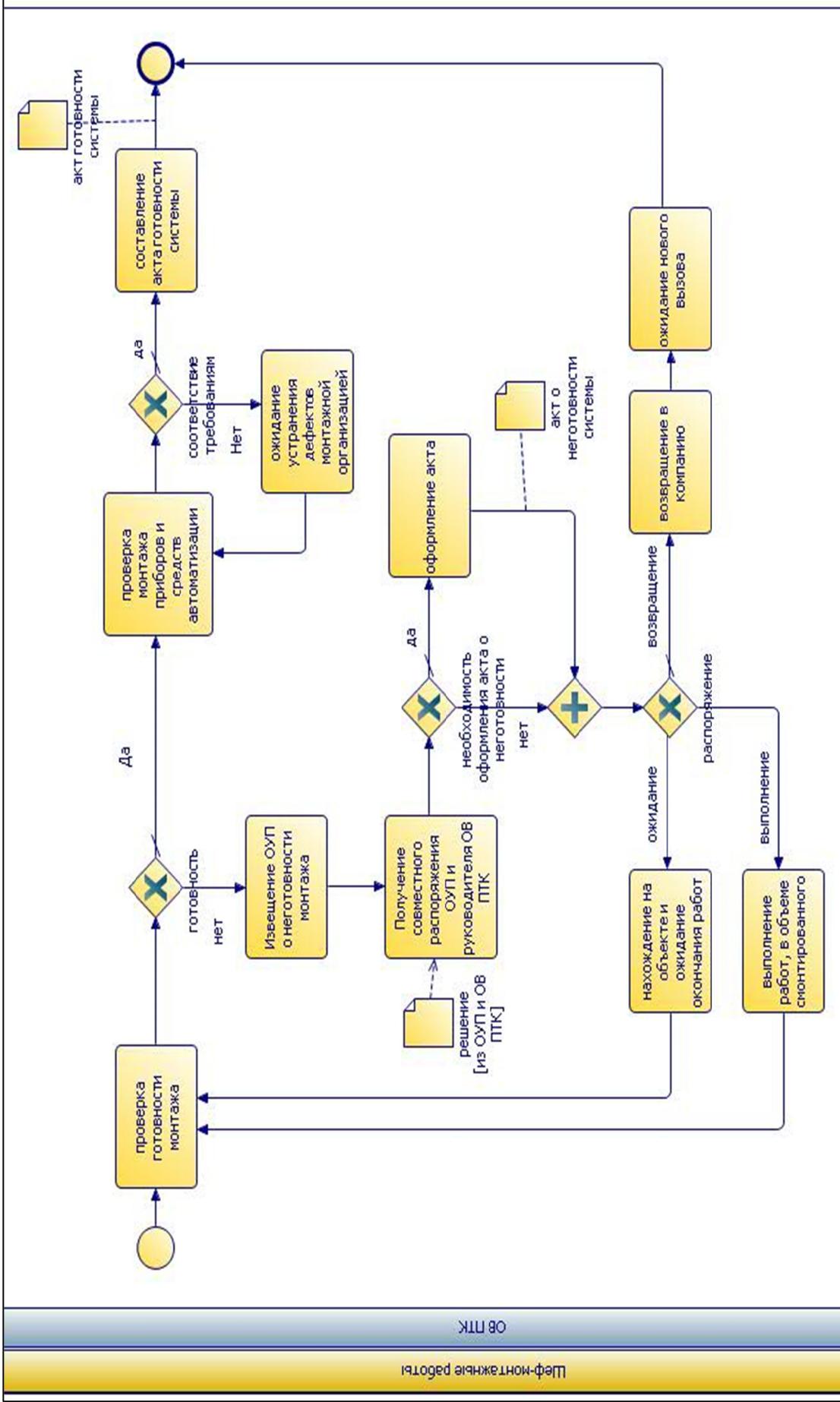


Рис. 2.34. Подпроцесс «Шеф-монтажные работы»

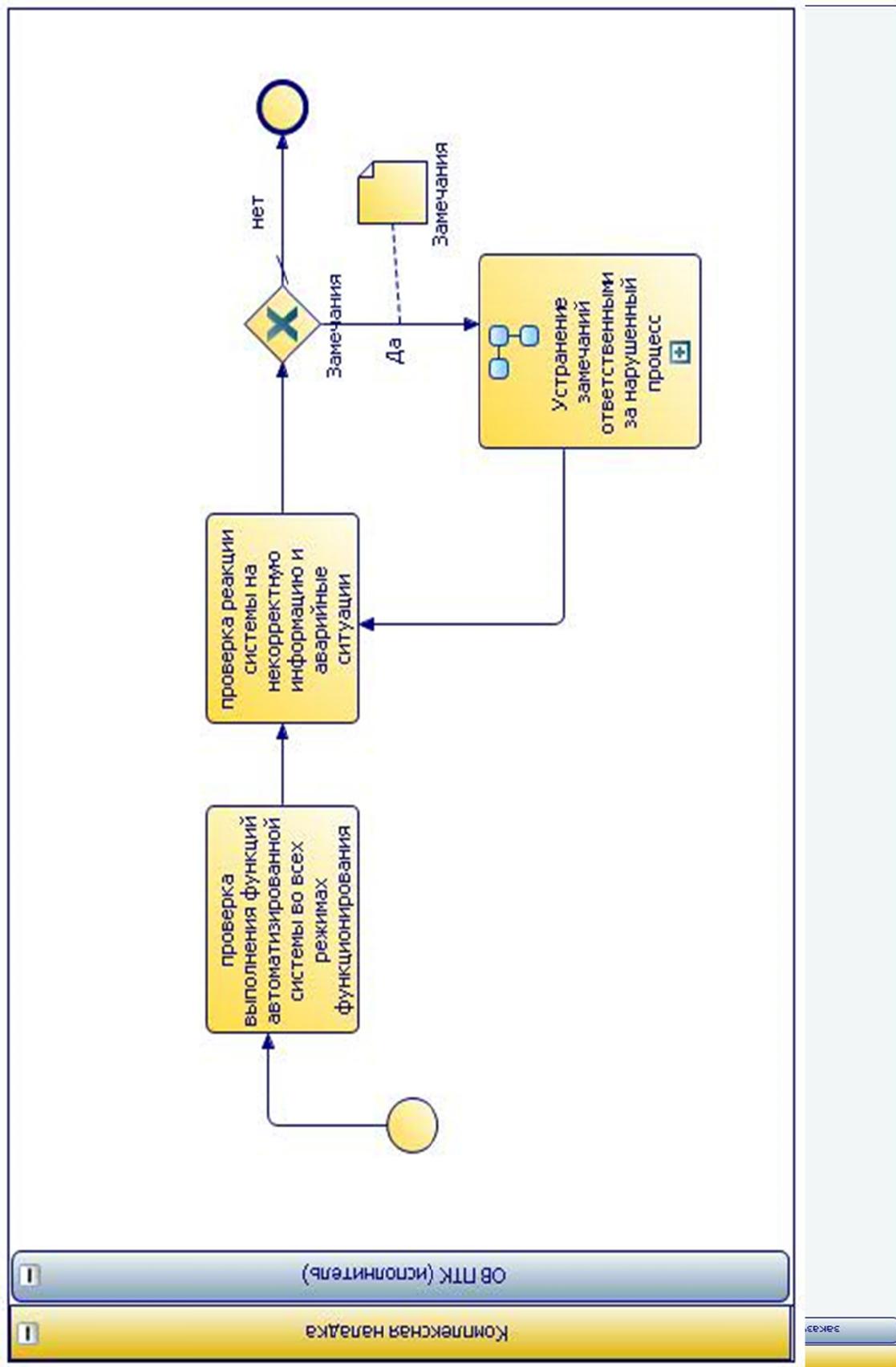


Рис. 2.35. Подпроцесс «Автономная наладка»

Рис. 2.36. Подпроцесс «Комплексная наладка»

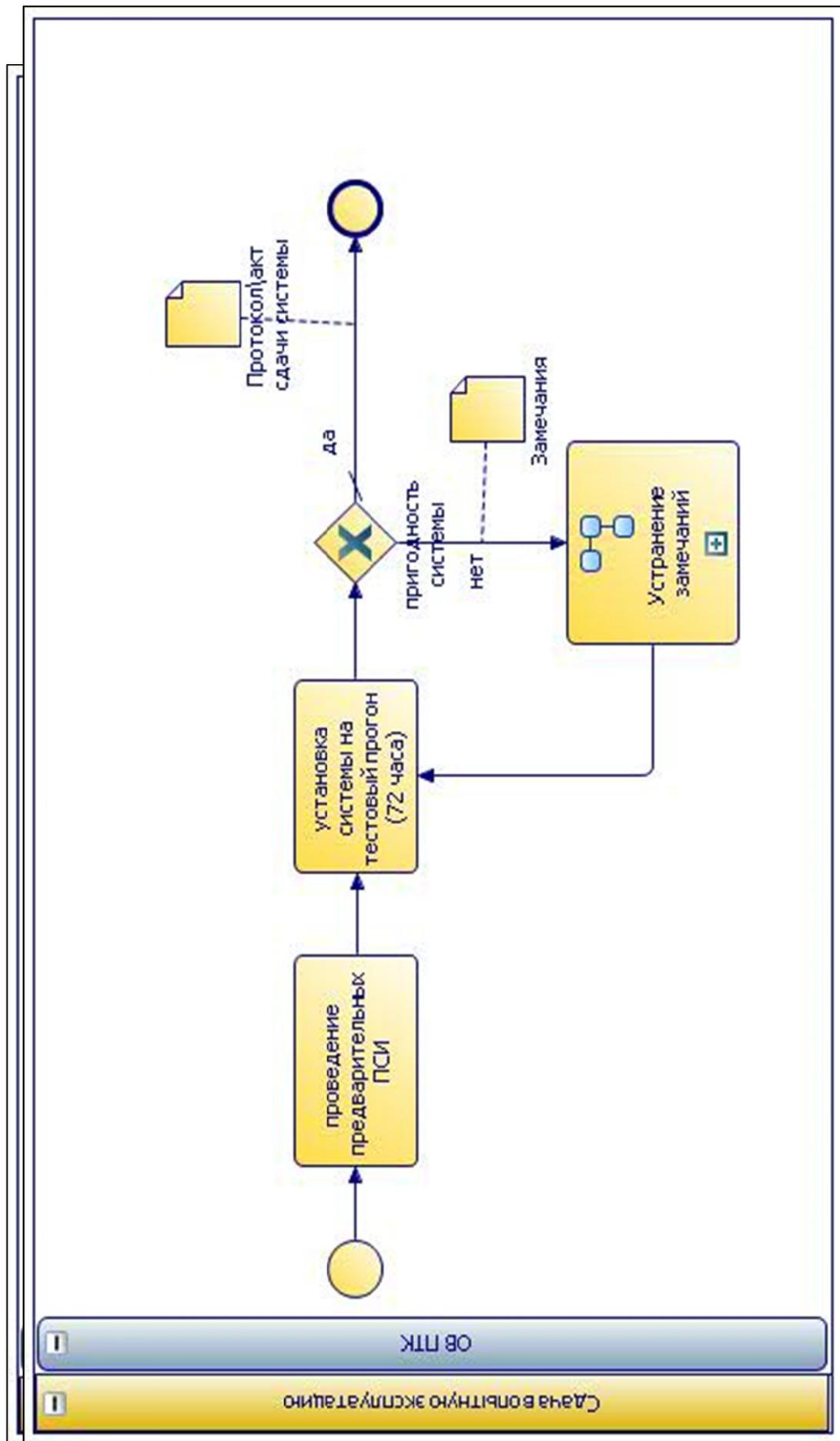


Рис. 2.37. Подпроцесс «Сдача в опытную эксплуатацию»

Рис. 2.38. Подпроцесс «Сдача в промышленную эксплуатацию»

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

1. Какие виды связей используются в нотации BPMN-моделях?
2. Как изображается связь «Ассоциация» в BPMN-моделях?
3. Что такое артефакт и какие типы их используются в нотации BPMN?
4. Для чего применяются артефакты в BPMN-моделях?
5. Что такое исполняемый бизнес-процесс?
6. Что такое неисполняемый бизнес-процесс?
7. Какие существуют системы управления бизнес-процессами?

ЛАБОРАТОРНАЯ РАБОТА № 9

Моделирование бизнес-процессов в нотации BPMN

Цель: закрепление знаний, отработка навыков по разработке моделей бизнес-процессов в нотации BPMN.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Нотации BPMN можно и даже нужно использовать при работе с малым и средним бизнесом. Возможно, что вы не будете реализовать бизнес-модель на уровне программного обеспечения, так как это всегда — дополнительные затраты, и в условиях малого бизнеса нет необходимости в подобных инструментах контроля и анализа работы [9].

Но, тем не менее, на уровне неисполняемых бизнес-процессов зачастую наиболее активно используют именно BPMN. Дело в том, что при всей сложности вхождения (т.е. изучения и умения работать с нотациями), уровень понимания BPMN — низкий, т.е. для чтения нотаций не требуется вообще никаких особых знаний и навыков. Графические нотации понимаются интуитивно. Эта нотация создавалась специально для того, чтобы найти общий язык между аналитиком и обычными бизнесменами (управленцами).

Таким образом, использование BPMN позволяет экономить время разработчика и время заказчика (руководителя). А также позволяет добиваться максимально высокого уровня взаимопонимания. Нотации не позволяют “двойного прочтения”, а потому очень помогают в работе.

2.8. Минусы нотации BPMN

Несмотря на огромное количество плюсов, о которых было сказано выше, для выбора любого инструмента важно также понимать и возможные минусы. Не стал исключением и BPMN. Из недостатков можно отметить:

- ✓ система имеет значительное количество понятий и терминов, их нужно знать и применять грамотно.

- ✓ высокий уровень вхождения. Как и любой инструмент с широкими возможностями требует большего времени на изучение, по сравнению с другими нотациями (IDEF0, IDEF3).

- ✓ необходимо знание бизнес-анализа. В BPMN модели — это не просто картинки или схемы, которые вы можете нарисовать в любом графическом редакторе. Здесь очень важна грамотная структура и четкая последовательность.

2.9. Рекомендации по разработке BPMN-диаграмм

В своей статье автор Кинзябулатов Рамиль советы по разработке BPMN-моделей, основанные на знаниях особенностей бизнес-моделирования и личном практическом опыте [9].

1. Необходимо запланировать начало и конец процесса. С этого начинается моделирование любого процесса. Так обозначаются рамки, в которых предстоит работать.

2. Для начала лучше всего описать линейную последовательность действий: шаг за шагом движение от начала к финальному результату. Далее при необходимости добавляются ветвления. В таком порядке работать намного проще, чем ставить две или более ветвей одновременно и путаться в стрелках, что откуда и куда идет.

3. Определить ответственных лиц. Установить исполнителей и ответственных.

4. Добавить данные, сноски, комментарии.

Возможно создавать диаграммы и в другой последовательности, главное определить четкую логику процесса моделирования и избегать путаницы.

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. На основе практических знаний, полученных при выполнении предыдущей работы, разработайте собственную BPMN-модель;

2. Тема модели определяется студентом и согласовывается с преподавателем;

3. Требования к модели:

- а) модель должна содержать элементы действия;

- б) модель должна содержать не менее 3-х подпроцессов;

- в) модель должна содержать участников процесса;

- г) модель должна содержать все рассмотренные типы элементов: шлюзы, артефакты, связи, события;
- 4. Разработайте модели каждого из подпроцессов основного процесса
- 5. По результатам работы составьте отчет. Отчет должен содержать:
 - а) титульный лист;
 - б) все BPMN диаграммы, оформленные и подписанные как рисунки;
 - в) текстовые описания основного процесса и подпроцессов;
 - г) карту основного процесса;
 - д) описание документов, используемых в процессах.

ГЛАВА III. УНИФИЦИРОВАННЫЙ ЯЗЫК МОДЕЛИРОВАНИЯ UML

В начале 1997 г. в результате сотрудничества консорциума компаний UML-Partners, в который вошли такие компании, как Rational Software, Microsoft, IBM, Hewlett-Packard, Oracle, DEC, Unisys, IntelliCorp, Platinum Technology, увидела свет первая версия нового языка объектно-ориентированного моделирования и анализа — Unified Modeling Language (UML — унифицированный язык моделирования). Он объединил объектно-ориентированные методики, наилучшим образом зарекомендовавшие себя на практике. В течение 1997 г. UML стал стандартом де-факто языков объектного моделирования. В сентябре 1997 г. разработанный язык был представлен Object Management Group (OMG) — консорциуму из более чем 600 компаний — поставщиков объектных технологий, отвечающему за принятие стандартов в этой области, для принятия UML в качестве отраслевого стандарта расширяемого языка визуального моделирования. Введение в UML целесообразно начать с известной картинки уже достаточно долго живущей в Интернете (рис. 3.1). Она в полной мере демонстрирует типичный процесс создания продукта, или "решения" [10]. Здесь четко видны все проблемы программной инженерии, в частности проблемы с коммуникацией и пониманием, вызванные отсутствием четкой спецификации создаваемого продукта.

Создатели UML определяют его следующим образом: «Unified Modeling Language (UML) — язык для определения, представления, проектирования и документирования программных систем, бизнес-систем и прочих систем непрограммного обеспечения. UML представляет собрание лучших технических методов, которые успешно доказали свою применимость при моделировании больших и сложных систем. Целью создателей UML было управление сложностью систем, поскольку они увеличиваются в возможностях и своих масштабах».

Унифицированный язык моделирования появился в основном благодаря усилиям Гради Буча (Grady Booch), Джеймса Рамбо (James Rumbaugh) и Айвара Якобсона (Ivar Jacobson). В настоящее время консорциум OMG принял этот язык как стандартный язык моделирования, который предоставляет разработчикам четкую нотацию, позволяющую отображать модели общепринятыми и понятными каждому члену проекта графическими элементами. За последние годы развития язык существенно усовершенствовался и на сегодняшний день используется версия вторая версия языка UML, одобренная OMG.

Бизнес-аналитики получают в UML средство описания и анализа жизнедеятельности бизнес-систем. UML позволяет визуализировать процессы и объекты, которые включены в систему, и описать взаимосвязи между ними. Метод анализа, предлагаемый UML,

позволяет описать стороны деятельности предприятия, наиболее важные для анализа эффективности. Это позволяет перестроить жизнедеятельность предприятия (произвести реинжиниринг бизнеса) таким образом, чтобы достигнуть наибольшей эффективности при функционировании системы. В данной сфере UML развил и дополнил методологию Айвара Якобсона (так называемых *прецедентов* использования системы), зарекомендовавшую себя в мире как достаточно успешную при анализе и реинжиниринге бизнес-систем.

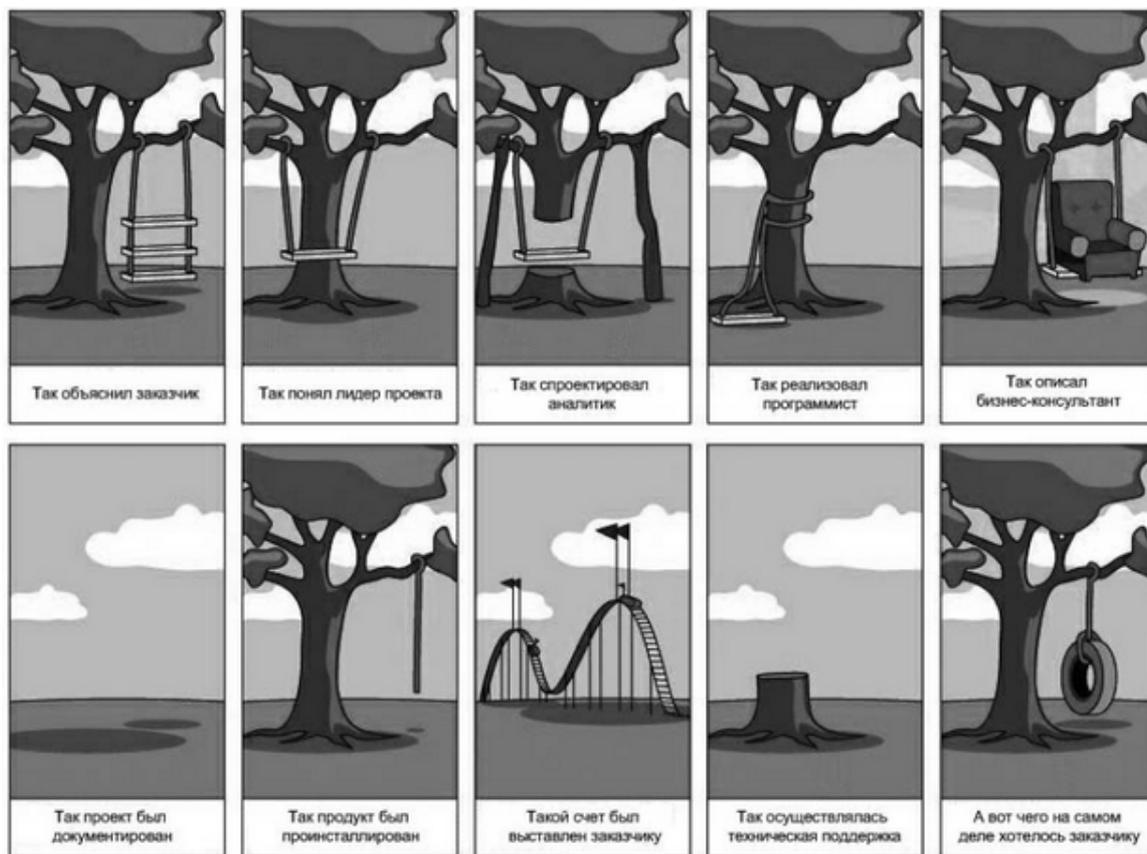


Рис. 3.1. Демонстрация разных взглядов заказчика и исполнителя на проблему

ЛАБОРАТОРНАЯ РАБОТА № 10

Диаграммы вариантов использования в UML

Цель работы: изучение семантики и получение навыков разработки диаграмм вариантов использования при помощи UML.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Визуальное моделирование в UML можно представить как некоторый процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной системы к логической, а затем и к физической модели соответствующей программной системы. Для достижения этих целей вначале строится модель в форме, так называемой **диаграммы вариантов использования (use case diagram)**, которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки [11].

Разработка диаграммы вариантов использования преследует цели:

- ✓ определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы.
- ✓ сформулировать общие требования к функциональному поведению проектируемой системы.
- ✓ разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей.
- ✓ подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть данной диаграммы состоит в следующем [12]: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых вариантов использования. При этом **актером (actor)** или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, **вариант использования (use case)** служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой. Базовыми элементами диаграммы являются **вариант использования и актер.**

Вариант использования (use case) - внешняя спецификация последовательности действий, которые система или другая сущность могут выполнять в процессе взаимодействия с актерами. Каждый вариант использования определяет последовательность действий, которые должны быть выполнены проектируемой системой при взаимодействии ее с соответствующим актером. Диаграмма вариантов может

дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов. Такой пояснительный текст получил название примечания или сценария.

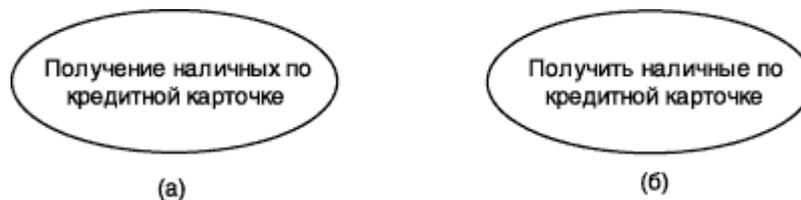


Рис. 3.2. Графическое обозначение варианта использования

Отдельный вариант использования обозначается на диаграмме эллипсом, внутри которого содержится его краткое название (рис. 3.2 а) или имя в форме глагола (рис. 3.2 б) с пояснительными словами.

Цель варианта использования заключается в том, чтобы определить законченный аспект или фрагмент поведения некоторой сущности без раскрытия внутренней структуры этой сущности. В качестве такой сущности может выступать исходная система или любой другой элемент модели, который обладает собственным поведением, подобно подсистеме или классу в модели системы.

Каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемую сущность или систему по запросу пользователя (актера), т.е. определяет способ применения этой сущности. Сервис, который инициализируется по запросу пользователя, представляет собой законченную последовательность действий. Это означает, что после того как система закончит обработку запроса пользователя, она должна возвратиться в исходное состояние, в котором готова к выполнению следующих запросов.

Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы. Для удобства множество вариантов использования может рассматриваться как отдельный пакет.

Примерами вариантов использования могут являться следующие действия: проверка состояния текущего счета клиента, оформление заказа на покупку товара,

получение дополнительной информации о кредитоспособности клиента, отображение графической формы на экране монитора и другие действия.

Актер (actor) - согласованное множество ролей, которые играют внешние сущности по отношению к вариантам использования при взаимодействии с ними. При этом актеры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой. Каждый актер может рассматриваться как некая отдельная роль относительно конкретного варианта использования. Стандартным графическим обозначением актера на диаграммах является фигурка "человечка", под которой записывается конкретное имя актера (рис. 3.3).



Рис. 3.3. Графическое обозначение актера

В некоторых случаях актер может обозначаться в виде прямоугольника класса с ключевым словом <<actor>> и обычными составляющими элементами класса. Имена актеров должны записываться заглавными буквами и следовать рекомендациям использования имен для типов и классов модели. При этом символ отдельного актера связывает соответствующее описание актера с конкретным именем. Имена абстрактных актеров, как и других абстрактных элементов языка UML, рекомендуется обозначать курсивом.

***Примечание:** имя актера должно быть достаточно информативным с точки зрения семантики. Вполне подходят для этой цели наименования должностей в компании (например, продавец, кассир, менеджер, президент). Не рекомендуется давать актерам имена собственные (например, "А. Пушкин") или моделей конкретных устройств (например, "компьютер Pentium 4"), даже если это с очевидностью следует из контекста проекта. Дело в том, что одно и то же лицо может выступать в нескольких ролях и, соответственно, обращаться к различным сервисам системы. Например, посетитель банка может являться как потенциальным клиентом, и тогда он востребует один из его сервисов, а может быть и налоговым инспектором или следователем прокуратуры. Сервис для последнего может быть совершенно исключительным по своему характеру [12].*

Примерами актеров могут быть: клиент банка, банковский служащий, продавец магазина, менеджер отдела продаж, пассажир авиарейса, водитель автомобиля, администратор гостиницы, сотовый телефон и другие сущности, имеющие отношение к концептуальной модели соответствующей предметной области.

Так как в общем случае актер всегда находится вне системы, его внутренняя структура никак не определяется. Для актера имеет значение только его внешнее представление, т.е. то, как он воспринимается со стороны системы. Актеры взаимодействуют с системой посредством передачи и приема сообщений от вариантов использования. Сообщение представляет собой запрос актером сервиса от системы и получение этого сервиса. Это взаимодействие может быть выражено посредством ассоциаций между отдельными актерами и вариантами использования или классами. Кроме этого, с актерами могут быть связаны интерфейсы, которые определяют, каким образом другие элементы модели взаимодействуют с этими актерами [13].

Два и более актера могут иметь общие свойства, т. е. взаимодействовать с одним и тем же множеством вариантов использования одинаковым образом. Такая общность свойств и поведения представляется в виде рассматриваемого ниже отношения обобщения с другим, возможно, абстрактным актером, который моделирует соответствующую общность ролей.

Интерфейс (interface) служит для спецификации параметров модели, которые видимы извне без указания их внутренней структуры. В языке UML интерфейс является классификатором и характеризует только ограниченную часть поведения моделируемой сущности. Применительно к диаграммам вариантов использования, интерфейсы определяют совокупность операций, которые обеспечивают необходимый набор сервисов или функциональности для актеров. Интерфейсы не могут содержать ни атрибутов, ни состояний, ни направленных ассоциаций. Они содержат только операции без указания особенностей их реализации. Формально интерфейс эквивалентен абстрактному классу без атрибутов и методов с наличием только абстрактных операций [13].

На диаграмме вариантов использования интерфейс изображается в виде маленького круга, рядом с которым записывается его имя (рис. 3.4 а). В качестве имени может быть существительное, которое характеризует соответствующую информацию или сервис (например, "датчик", "сирена", "видеокамера"), но чаще строка текста (например, "запрос к базе данных", "форма ввода", "устройство подачи звукового сигнала"). Если имя записывается на английском, то оно должно начинаться с заглавной буквы I, например, *ISecureInformation*, *ISensor* (рис. 3.4 б).

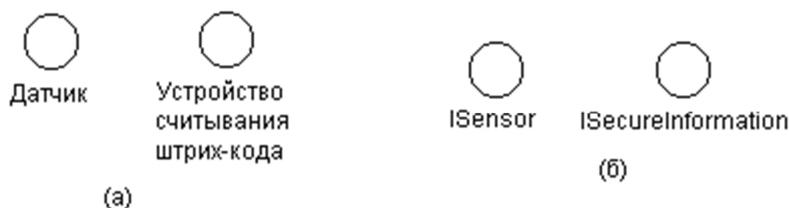


Рис. 3.4. Графическое изображение интерфейсов на диаграммах вариантов использования

Графический символ отдельного интерфейса может соединяться на диаграмме **сплошной линией** с тем вариантом использования, который его поддерживает. Сплошная линия в этом случае указывает на тот факт, что **связанный с интерфейсом вариант использования должен реализовывать все операции, необходимые для данного интерфейса, а возможно и больше** (рис. 3.5 а). Кроме этого, интерфейсы могут соединяться с вариантами использования **пунктирной линией со стрелкой** (рис. 3.5 б), означающей, что **вариант использования предназначен для спецификации только того сервиса, который необходим для реализации данного интерфейса.**

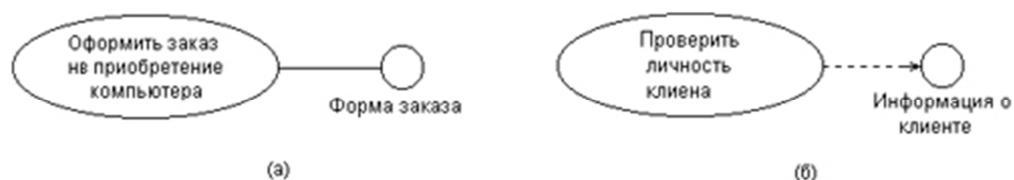


Рис. 3.5. Графическое изображение взаимосвязей интерфейсов с вариантами использования

Важность интерфейсов заключается в том, что они определяют стыковочные узлы в проектируемой системе, что совершенно необходимо для организации коллективной работы над проектом. Более того, спецификация интерфейсов способствует "безболезненной" модификации уже существующей системы при переходе на новые технологические решения. В этом случае изменению подвергается только реализация операций, но никак не функциональность самой системы. А это обеспечивает совместимость последующих версий программ с первоначальными при спиральной технологии разработки программных систем.

Примечания (notes) в языке UML предназначены для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта. В качестве такой информации могут быть комментарии разработчика (например, дата и версия разработки диаграммы или ее отдельных компонентов), ограничения (например, на значения отдельных связей или экземпляры сущностей) и помеченные значения. Применительно к диаграммам вариантов использования примечание может носить самую общую информацию, относящуюся к общему контексту системы.

Графически примечания обозначаются прямоугольником с "загнутым" верхним правым углом (рис. 3.6). Внутри прямоугольника содержится текст примечания. Примечание может относиться к любому элементу диаграммы, в этом случае их соединяет

пунктирная линия. Если примечание относится к нескольким элементам, то от него проводятся, соответственно, несколько линий. Разумеется, примечания могут присутствовать не только на диаграмме вариантов использования, но и на других канонических диаграммах.



Рис. 3.6. Примеры примечаний в языке UML

Отношения на диаграмме вариантов использования

Между компонентами диаграммы вариантов использования могут существовать различные отношения, которые описывают взаимодействие экземпляров одних актеров и вариантов использования с экземплярами других актеров и вариантов. Один актер может взаимодействовать с несколькими вариантами использования. В этом случае этот актер обращается к нескольким сервисам данной системы. В свою очередь один вариант использования может взаимодействовать с несколькими актерами, предоставляя для всех них свой сервис. Следует заметить, что два варианта использования, определенные для одной и той же сущности, не могут взаимодействовать друг с другом, поскольку каждый из них самостоятельно описывает законченный вариант использования этой сущности. Более того, варианты использования всегда предусматривают некоторые сигналы или сообщения, когда взаимодействуют с актерами за пределами системы. В то же время могут быть определены другие способы для взаимодействия с элементами внутри системы [14].

Отношение (*relationship*) — семантическая связь между отдельными элементами модели. В языке UML имеется несколько стандартных видов отношений между актерами и вариантами использования [12]:

- ✓ отношение ассоциации (*association relationship*);
- ✓ отношение расширения (*extend relationship*);
- ✓ отношение обобщения (*generalization relationship*);
- ✓ отношение включения (*include relationship*);

При этом общие свойства вариантов использования могут быть представлены тремя различными способами, а именно с помощью отношений расширения, обобщения и включения.

Отношение ассоциации – одно из фундаментальных понятий в языке UML и в той или иной степени используется при построении всех графических моделей систем в форме канонических диаграмм. Применительно к диаграммам вариантов использования ассоциация служит для обозначения специфической роли актера при его взаимодействии с отдельным вариантом использования. Другими словами, ассоциация специфицирует семантические особенности взаимодействия актеров и вариантов использования в графической модели системы. На диаграмме вариантов использования, так же, как и на других диаграммах, отношение ассоциации обозначается сплошной линией между актером и вариантом использования. Эта линия может иметь некоторые дополнительные обозначения, например, имя и кратность (рис. 3.7) [12].

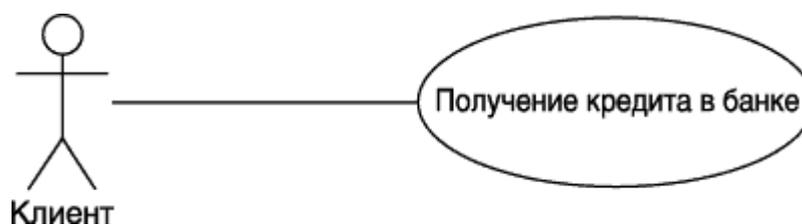


Рис. 3.7. Пример графического представления отношения ассоциации между актером и вариантом использования

В контексте диаграммы вариантов использования отношение ассоциации между актером и вариантом использования может указывать на то, что актер инициирует соответствующий вариант использования. **Такого актера называют главным.** В других случаях подобная ассоциация может указывать на актера, которому предоставляется справочная информация о результатах функционирования моделируемой системы. **Таких актеров часто называют второстепенными** [15].

Отношение расширения (extend) определяет взаимосвязь базового варианта использования с другим вариантом использования, функциональное поведение которого задействуется базовым не всегда, а только при выполнении дополнительных условий. В языке UML отношение расширения является зависимостью, направленной к базовому варианту использования и соединенной с ним в так называемой точке расширения. Отношение расширения между вариантами использования обозначается как отношение зависимости в форме пунктирной линии со стрелкой, направленной от того варианта использования, который является расширением для базового варианта использования.

Данная линия со стрелкой должна быть помечена стереотипом <<extend>> ("расширяет"), как показано на рис. 3.8. [15].



Рис. 3.8. Пример графического изображения отношения расширения между вариантами использования

Отношение расширения отмечает тот факт, что один из вариантов использования может присоединять к своему поведению некоторое дополнительное поведение, определенное для другого варианта использования. Данное отношение включает в себя некоторое условие и ссылки на точки расширения в базовом варианте использования. Чтобы расширение имело место, должно быть выполнено определенное условие данного отношения. Ссылки на точки расширения определяют те места в базовом варианте использования, в которые должно быть помещено соответствующее расширение при выполнении условия.

Один из вариантов использования может быть расширением для нескольких базовых вариантов, а также иметь в качестве собственных расширений несколько других вариантов. Базовый вариант использования может дополнительно никак не зависеть от своих расширений.

Семантика отношения расширения определяется следующим образом. Если экземпляр варианта использования выполняет некоторую последовательность действий, которая определяет его поведение, и при этом имеется точка расширения на экземпляр другого варианта использования, которая является первой из всех точек расширения у исходного варианта, то проверяется условие данного отношения. Если условие выполняется, исходная последовательность действий расширяется посредством включения действий экземпляра другого варианта использования. Следует заметить, что условие отношения расширения проверяется лишь один раз — при первой ссылке на точку расширения, и если оно выполняется, то все расширяющие варианты использования вставляются в базовый вариант.

В представленном выше примере (рис. 3.9) при оформлении заказа на приобретение товара только в некоторых случаях может потребоваться предоставление клиенту каталога всех товаров. При этом условием расширения является запрос от клиента на получение каталога товаров. Очевидно, что после получения каталога клиенту необходимо некоторое время на его изучение, в течение которого оформление заказа приостанавливается. После ознакомления с каталогом клиент решает либо в пользу

выбора отдельного товара, либо отказа от покупки вообще. Сервис или вариант использования "Оформить заказ на приобретение товара" может отреагировать на выбор клиента уже после того, как клиент получит для ознакомления каталог товаров.

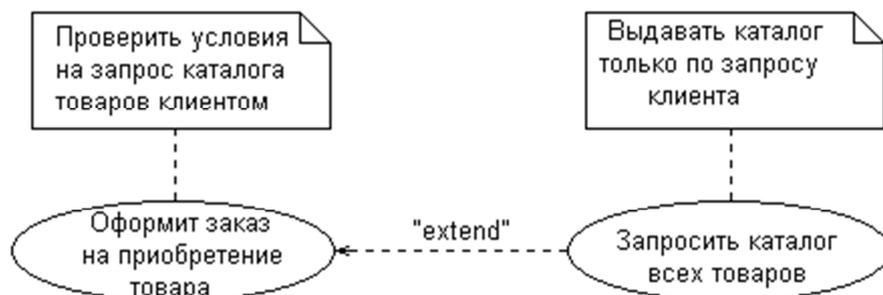


Рис. 3.9. Графическое изображение отношения расширения с примечаниями условий выполнения вариантов использования

Точка расширения может быть как отдельной точкой в последовательности действий, так и множеством отдельных точек. Важно представлять себе, что если отношение расширения имеет некоторую последовательность точек расширения, только первая из них может определять множество отдельных точек. Все остальные должны определять в точности одну такую точку. Какая из точек должна быть первой точкой расширения, т.е. определяться единственным расширением. Такие ссылки на расположение точек расширения могут быть представлены различными способами, например, с помощью текста примечания на естественном языке (рис. 3.9) [12].

Отношение обобщения служит для указания того факта, что некоторый вариант использования А может быть обобщен до варианта использования В. В этом случае вариант А будет являться специализацией варианта В. При этом В называется предком или родителем по отношению А, а вариант А — потомком по отношению к варианту использования В. Следует подчеркнуть, что потомок наследует все свойства и поведение своего родителя, а также может быть дополнен новыми свойствами и особенностями поведения. Графически данное отношение обозначается сплошной линией со стрелкой в форме незакрашенного треугольника, которая указывает на родительский вариант использования (рис. 3.10). Эта линия со стрелкой имеет специальное название — стрелка "обобщение" [12].



Рис. 3.10. Пример графического изображения отношения обобщения между вариантами использования

Отношение обобщения между вариантами использования применяется в том случае, когда необходимо отметить, что дочерние варианты использования обладают всеми

атрибутами и особенностями поведения родительских вариантов. При этом дочерние варианты использования участвуют во всех отношениях родительских вариантов. В свою очередь, дочерние варианты могут наделяться новыми свойствами поведения, которые отсутствуют у родительских вариантов использования, а также уточнять или модифицировать наследуемые от них свойства поведения.

Применительно к данному отношению, один вариант использования может иметь несколько родительских вариантов. В этом случае реализуется множественное наследование свойств и поведения отношения предков: С другой стороны, один вариант использования может быть предком для нескольких дочерних вариантов, что соответствует таксономическому характеру отношения обобщения [12].

Между отдельными актерами также может существовать отношение обобщения. Данное отношение является направленным и указывает на факт специализации одних актеров относительно других. Например, отношение обобщения от актера А к актеру В отмечает тот факт, что каждый экземпляр актера А является одновременно экземпляром актера В и обладает всеми его свойствами. В этом случае актер В является родителем по отношению к актеру А, а актер А, соответственно, потомком актера В. При этом актер А обладает способностью играть такое же множество ролей, что и актер В. Графически данное отношение также обозначается стрелкой обобщения, т. е. сплошной линией со стрелкой в форме незакрашенного треугольника, которая указывает на родительского актера (рис. 3.11).

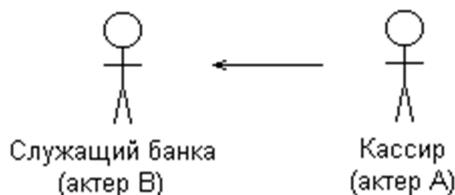


Рис. 3.11. Пример графического изображения отношения обобщения между актерами

Отношение включения (*include*) в языке UML — это разновидность отношения зависимости между базовым вариантом использования и его специальным случаем. При этом отношением зависимости (*dependency*) является такое отношение между двумя элементами модели, при котором изменение одного элемента (независимого) приводит к изменению другого элемента (зависимого). Отношение включения устанавливается только между двумя вариантами использования и указывает на то, что заданное поведение для одного варианта использования включается в качестве составного фрагмента в последовательность поведения другого варианта использования. Данное отношение

является направленным бинарным отношением в том смысле, что пара экземпляров вариантов использования всегда упорядочена в отношении включения [12].

Так, например, отношение включения, направленное от варианта использования **"Предоставление кредита в банке"** к варианту использования **"Проверка платежеспособности клиента"**, указывает на то, что каждый экземпляр первого варианта использования всегда включает в себя функциональное поведение или выполнение второго варианта использования. В этом смысле поведение второго варианта использования является частью поведения первого варианта использования на данной диаграмме. Графически данное отношение обозначается как отношение зависимости в форме пунктирной линии со стрелкой, направленной от базового варианта использования к включаемому варианту использования. При этом данная линия помечается стереотипом `<<include>>`, как показано на рис. 3.12 [15].

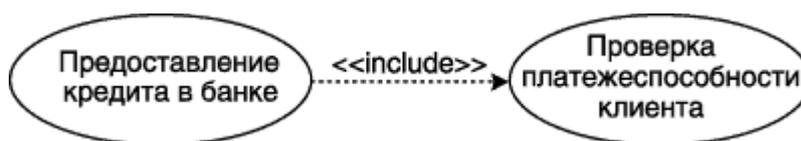


Рис. 3.12. Пример графического изображения отношения включения между вариантами использования

Семантика этого отношения определяется следующим образом. Процесс выполнения базового варианта использования включает в себя как собственное подмножество последовательность действий, которая определена для включаемого варианта использования. При этом выполнение включаемой последовательности действий происходит всегда при инициировании базового варианта использования [12].

Один вариант использования может входить в несколько других вариантов, а также содержать в себе другие варианты. Включаемый вариант использования является независимым от базового варианта в том смысле, что он предоставляет последнему инкапсулированное поведение, детали реализации которого скрыты от последнего и могут быть легко перераспределены между несколькими включаемыми вариантами использования. Более того, базовый вариант зависит только от результатов выполнения включаемого в него варианта использования, но не от структуры включаемых в него вариантов.

ПРАКТИЧЕСКАЯ ЧАСТЬ

- 1) Запустить программу MS Visio;

- 2) Для того, чтобы загрузить нужную панель инструментов для разработки диаграмм при создании документа в поле «Загрузить шаблоны из Интернета» укажите «UML»
- 3) В найденных пунктах выберите «UML. Варианты использования».

Задание 1. Разработать диаграмму вариантов использования «Снятие наличных в банкомате»

1. Создайте новый проект, выбрав тип диаграммы «Диаграмма вариантов использования». Для иллюстрации особенностей спецификации функциональных требований на диаграмме вариантов использования рассмотрим модель обычного банкомата.

Процесс функционирования банкомата хорошо знаком любому владельцу дебетовой или кредитной карточки и не требует дополнительного описания. Рассматриваемая система имеет *двух актеров*, один из которых является *клиентом банкомата*, а другой - *Банкомат*, который осуществляет выполнение соответствующих транзакций. Каждый из этих актеров взаимодействует с банкоматом, хотя главный актер *Клиент*, поскольку именно он инициирует функциональность банкомата [14].

Основные функциональные требования к банкомату заключаются в предоставлении клиенту возможности снятия наличных по карточке и получении справки о состоянии счета. Именно эти функциональные требования специфицируются отдельными вариантами использования, которые служат ключевыми элементами соответствующей концептуальной модели. Поскольку для выполнения этих вариантов использования необходимо аутентифицировать кредитную карточку, они оба обращаются к дополнительному сервису "Проверка ПИН-кода кредитной карточки". Как следует из существа выдвигаемых к банкомату функциональных требований, этот сервис может выступать в качестве отдельного варианта использования разрабатываемой диаграммы и соединяться с первыми двумя вариантами отношением включения. Соответствующая диаграмма вариантов использования может включать в себя только указанных двух актеров и три варианта использования (рис. 3.13).



Рис. 3.13. Диаграмма вариантов использования для модели банкомата

3) Построить диаграмму вариантов использования, показанную на рис. 3.13.

На следующем этапе разработки модели вариантов использования для рассматриваемой системы банкомата следует дополнить данную диаграмму текстовым сценарием. Этот сценарий будет дополнять диаграмму, раскрывая содержание и логическую последовательность отдельных действий, которые выполняются системой и актерами в процессе снятия наличных по кредитной карточке. В этом случае сценарий удобно представить в виде трех таблиц, каждая из которых описывает отдельный раздел шаблона.

В главном разделе сценария (табл. 3.1.) указывается имя рассматриваемого варианта использования, имена взаимосвязанных с ним актеров, цель выполнения варианта, условный тип и ссылки на другие варианты использования.

Таблица 3.1. Главный раздел сценария выполнения варианта использования "Снятие наличных по кредитной карточке"

Вариант использования	Снятие наличных по кредитной карточке
Актеры	Клиент, Банк
Цель	Получение требуемой суммы наличными
Краткое описание	Клиент запрашивает требуемую сумму. Банкомат обеспечивает доступ к счету клиента. Банкомат выдает клиенту наличные.
Тип	Базовый
Ссылки на другие варианты	Включает в себя ВИ:

использования	<ul style="list-style-type: none"> • Проверка ПИН-кода кредитной карточки • Идентифицировать кредитную карточку
----------------------	---

В следующем разделе сценария (табл. 3.2) описывается последовательность действий, приводящая к успешному выполнению рассматриваемого варианта использования. При этом инициатором действий должен выступать актер Клиент. Для удобства последующих ссылок каждое действие помечается порядковым номером в последовательности.

Таблица 3.2. Раздел Типичный ход событий сценария выполнения варианта использования "Снятие наличных по кредитной карточке"

Действия актеров	Отклик системы
1. Клиент вставляет кредитную карточку в устройство чтения банкомата Исключение №1: Кредитная карточка недействительна	2. Банкомат проверяет кредитную карточку 3. Банкомат предлагает ввести ПИН-код
4. Клиент вводит персональный PIN-код Исключение №2: Клиент вводит неверный ПИН-код	5. Банкомат проверяет ПИН-код 6. Банкомат отображает опции меню
7. Клиент выбирает снятие наличных со своего счета	8. Система делает запрос в Банк и выясняет текущее состояние счета клиента 9. Банкомат предлагает ввести требуемую сумму
10. Клиент вводит требуемую сумму 11. Банк проверяет введенную сумму Исключение №3: Требуемая сумма превышает сумму на счете клиента	12. Банкомат изменяет состояние счета клиента, выдает наличные и чек
13. Клиент получает наличные и чек	14. Банкомат предлагает клиенту забрать кредитную карточку
15. Клиент получает свою кредитную карточку	16. Банкомат отображает сообщение о готовности к работе

В третьем разделе сценария (табл.3.3) описывается последовательность действий, выполняемых при возникновении исключительных ситуаций или исключений.

Таблица 3.3. Раздел Исключения сценария выполнения варианта использования "Снятие наличных по кредитной карточке"

Исключение №1. Кредитная карточка недействительна или неверно вставлена	
Действия актера	Отклик системы
	3. Банкомат отображает информацию о неверно вставленной

	кредитной карточке 14. Банкомат возвращает клиенту его кредитную карточку
15. Клиент получает свою кредитную карточку	
Исключение №2. Клиент вводит неверный ПИН-код	
	6. Банкомат отображает информацию о неверном ПИН-коде
4. Клиент вводит новый ПИН-код	
Исключение №3. Требуемая сумма превышает сумму на счете клиента	
	12. Банкомат отображает информацию о превышении кредита
10. Клиент вводит новую требуемую сумму	

Можно дополнить данный сценарий, аналогичным образом описав не только варианты использования "Получение справки о состоянии счета" и "Проверка Пин-кода кредитной карточки", но и рассмотрев другие исключения, например, отказ клиента от получения наличных после проверки ПИН-кода и т.п. При этом полнота сценариев и модели вариантов использования будут определяться теми функциональными требованиями, которые сформулированы в рамках конкретного проекта разработки соответствующего банкомата.

Отдельные небольшие по своему объему сценарии могут быть размещены на диаграмме в форме примечаний.

Примечание (note) предназначено для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта.

В качестве такой информации могут быть комментарии разработчика (например, дата и версия разработки диаграммы или ее отдельных компонентов), ограничения (например, на значения отдельных связей или экземпляры сущностей) и помеченные значения. Применительно к диаграммам вариантов использования примечание может иметь уточняющую информацию, относящуюся к контексту тех или иных вариантов использования.

Задание 2. Построить диаграмму вариантов использования «Взаимодействие оператор мобильной связи и пользователя»

1. Рассмотрим модель функционирования мобильного телефона. Достоинством этого проекта является то, что он не требует специального описания предметной области, поскольку предполагает интуитивное знакомство читателей с особенностями функционирования телефона.

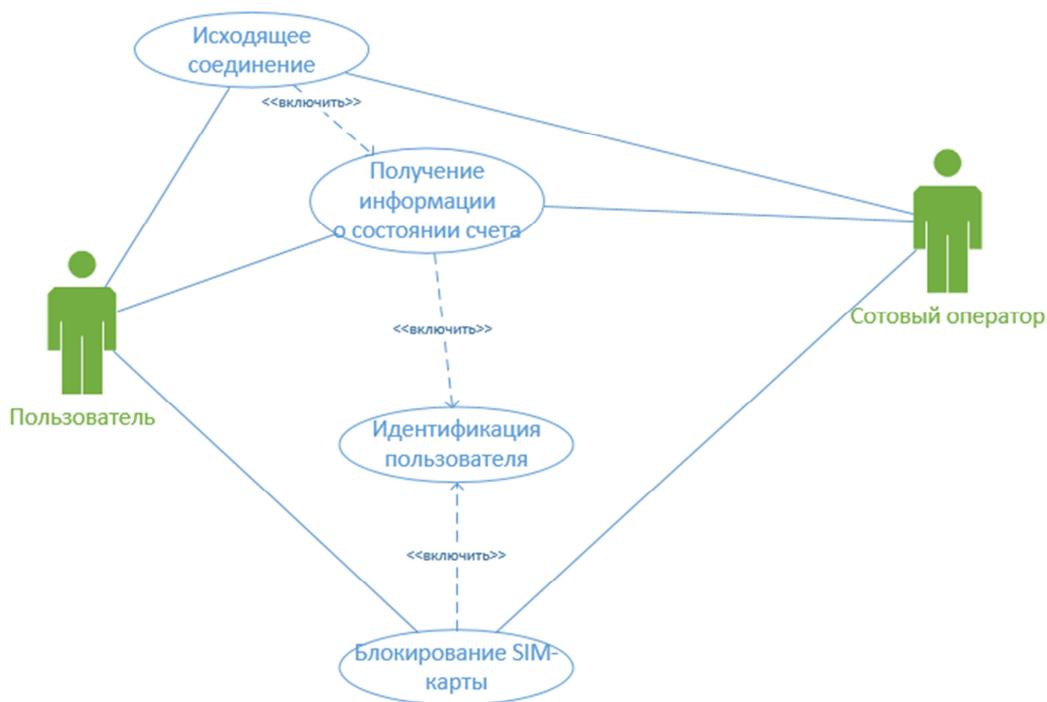


Рис. 3.14. Диаграмма вариантов использования для взаимодействия с сотовым оператором

Таблица 3.4. Главный раздел сценария выполнения варианта использования "Звонок мобильному оператору"

Действия актеров	Отклик системы
1. Клиент осуществляет набор номера.	2. Система осуществляет идентификацию пользователя. 3. Система делает запрос на проверку состояния счета
4. Сотовый оператор проверяет наличие денег на счету и возможность вызова Исключение №1: Недостаточно средств на счету Исключение №2: Некорректно введен номер	5. Система проверяет корректность введенного номера. 6. Система передает управление сотовому оператору для соединения с другим пользователем.

Действия актеров	Отклик системы
Исключение №1: Недостаточно средств на счету	5. Система выдает сообщение о недостатке

	средств и прерывает звонок
Исключение №2: Некорректно введен номер	
	б. Система выдает сообщение о некорректности номера и прерывает звонок

Вариант использования	Блокирование SIM-карты
Актеры	Пользователь, Сотовый оператор
Цель	Заблокировать SIM-карту в случае утраты телефона или необходимости в ней
Краткое описание	Пользователь делает запрос на блокировку, система осуществляет идентификацию пользователя и добавляет его в очередь call-центра сотового оператора для звонка и уточнения информации по блокировке, после звонка оператора он делает отметку-подтверждение и SIM-карта блокируется.
Тип	Базовый
Ссылки на другие варианты использования	Включает в себя ВИ: <ul style="list-style-type: none"> • Идентификация пользователя

Действия актеров	Отклик системы
1. Пользователь осуществляет запрос на блокировку	2. Система осуществляет идентификацию пользователя. 3. Система передает данные о пользователе в call-центр сотового оператора
4. Сотовый оператор связывается с пользователем, чтобы уточнить о причинах блокировки 5. Клиент разъясняет сотовому оператору причину блокировки. 6. Сотовый оператор блокирует SIM-карту	

2. В программе MS Visio постройте диаграмму, показанную на рис. 3.14.
3. На основе текстового сценария в табл. 3.4. составьте текстовое описание взаимодействия системы «Пользователь» и «Сотовый оператор».

Задание 3. Разработка вариантов использования системы «Сайт по доставке пиццы».

1. В программе MS Visio постройте диаграмму, показанную на рис. 3.15.
2. На основании диаграммы 3.15. создайте текстовый сценарий к работе сайта по доставке пиццы», где опишите главный сценарий и все сценарии-исключения.

Задание 4. Разработать собственную диаграмму вариантов использования на любую тему.

1. Выберите любой процесс по своему выбору и разработайте диаграмму вариантов использования;
2. Дополните диаграмму текстовыми сценариями (главным и исключаяющими);

Задание 5. Подготовьте отчет по лабораторной работе.

Составьте единый текстовый отчет в формате doc, который должен содержать все диаграммы, описания и сценарии, выполненные в данной лабораторной работе.

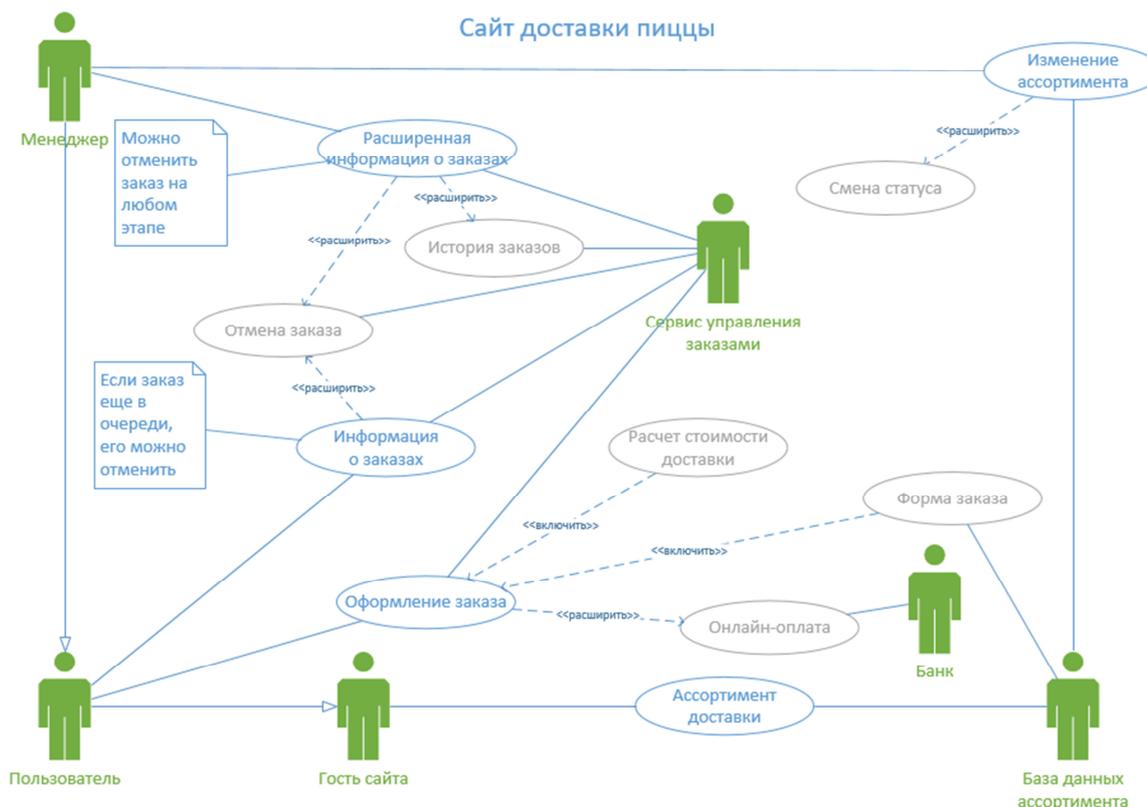


Рис. 3.15. Диаграмма вариантов использования «Сайт по доставке пиццы»

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

- 1) Какие виды отношений могут существовать между актерами и вариантами использования?
- 2) Каких актеров называют главными, а каких второстепенными?
- 3) В чем состоит суть диаграммы вариантов использования?
- 4) Объяснить графическую нотацию и смысл актера на диаграмме вариантов использования;
- 5) Объяснить графическую нотацию и смысл элемента «Вариант использования» на диаграмме вариантов использования;

- 6) В каком случае на диаграмме вариантов использования применяется отношение включения?
- 7) В каком случае на диаграмме вариантов использования применяется отношение обобщения?
- 8) В каком случае на диаграмме вариантов использования применяется отношение ассоциации?
- 9) В каком случае на диаграмме вариантов использования применяется отношение расширения?
- 10) Для чего служит элемент «интерфейс» на диаграмме вариантов использования и как он изображается графически?

ЛАБОРАТОРНАЯ РАБОТА № 11

Диаграммы классов в UML

Цель работы: изучение синтаксиса и семантики, получение навыков разработки диаграмм классов при помощи UML.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Диаграмма классов занимает одно из главных мест в объектно-ориентированном подходе. Фактически любая методология включает в себя некоторую разновидность диаграммы классов.

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами. В UML термин функциональность (feature) применяется в качестве основного термина, описывающего и свойства, и операции класса. [11]. Диаграммы классов могут применяться и при прямом проектировании, то есть в процессе разработки новой системы, и при *обратном проектировании* - описании существующих и используемых систем. Информация с диаграммы классов напрямую отображается в исходный код приложения - в большинстве существующих инструментов UML-моделирования возможна *кодогенерация* для определенного языка программирования (обычно Java или C++). Таким образом, диаграмма классов - конечный результат проектирования и отправная точка процесса разработки [16].

Существует два вида диаграмм классов:

✓ **Статический вид диаграммы** рассматривает логические взаимосвязи классов между собой;

✓ **Аналитический вид диаграммы** рассматривает общий вид и взаимосвязи классов, входящих в систему.

Существуют разные точки зрения на построение диаграмм классов в зависимости от целей их применения:

✓ **Концептуальная точка зрения** — диаграмма классов описывает модель предметной области, в ней присутствуют только классы прикладных объектов;

✓ **Точка зрения спецификации** — диаграмма классов применяется при проектировании информационных систем;

✓ **Точка зрения реализации** — диаграмма классов содержит классы, используемые непосредственно в программном коде (при использовании объектно-ориентированных языков программирования) [17].

На рис. 3.16 изображена типичная модель класса, понятная каждому, кто имел дело с обработкой заказов клиентов [11]. Прямоугольники на диаграмме представляют классы и разделены на три части: имя класса (жирный шрифт), его атрибуты и его операции.

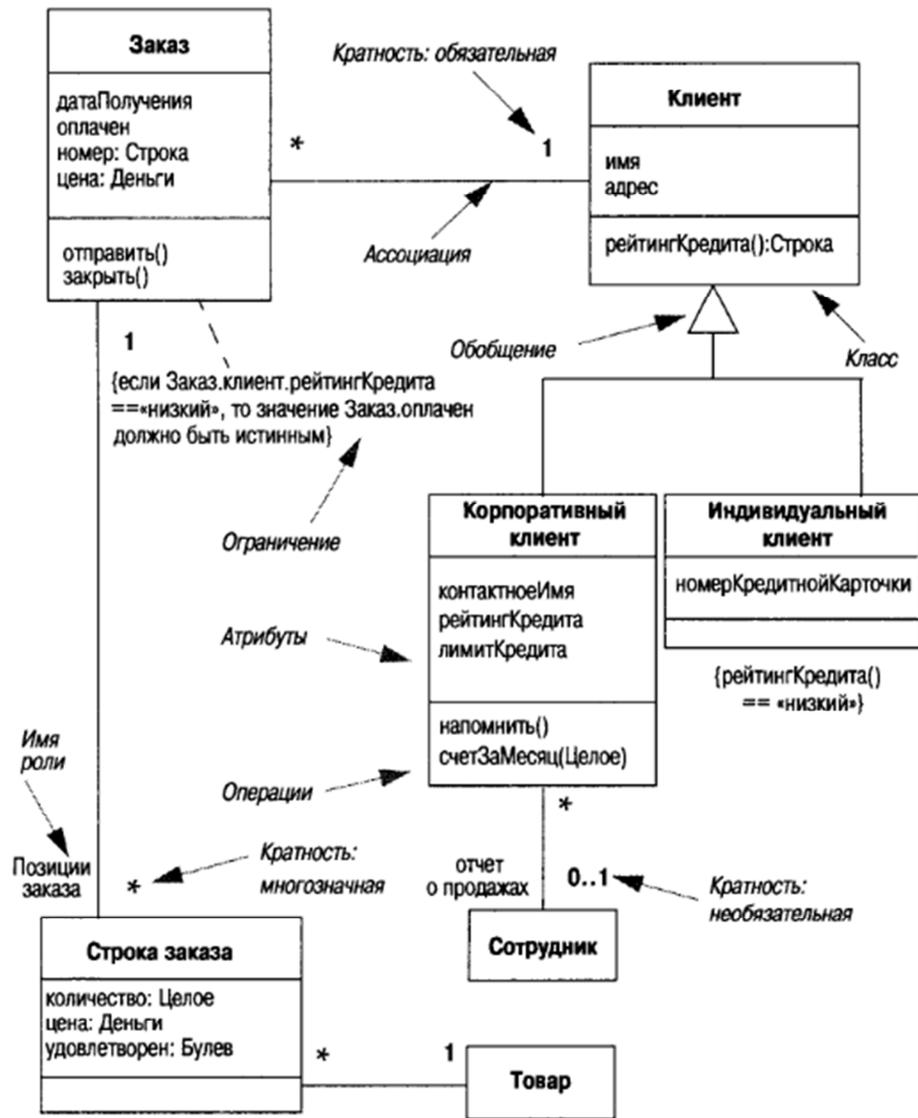


Рис. 3.16. Пример диаграммы классов

Класс на диаграмме показывается в виде прямоугольника, разделенного на 3 области (рис. 3.17). В верхней содержится название класса, в средней – описание атрибутов (свойств), в нижней – названия операций – услуг, предоставляемых объектами этого класса.

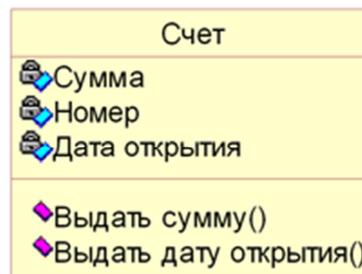


Рис. 3.17. Изображение класса в нотации UML

Так на рис.3.17. изображен прямоугольник класса «Счет», который имеет атрибуты сумма, номер и дата открытия. Этому классу доступны операции «Выдать сумму», «Выдать дату открытия». Атрибуты класса определяют состав и структуру данных, хранимых в объектах этого класса. Каждый атрибут имеет имя и тип, определяющий, какие данные он представляет. При реализации объекта в программном коде для атрибутов будет выделена память, необходимая для хранения всех атрибутов, и каждый атрибут будет иметь конкретное значение в любой момент времени работы программы. Объектов одного класса в программе может быть сколько угодно много, все они имеют одинаковый набор атрибутов, описанный в классе, но значения атрибутов у каждого объекта свои и могут изменяться в ходе выполнения программы [18].



Рис.3.18. Наследование атрибутов и операций

Отношения

Классы редко существуют автономно. Как правило, они взаимодействуют между собой разными способами. Существуют три вида отношений, особенно важных для объектно-ориентированного моделирования (рис. 3.19) [14].

- ✓ **зависимости**, которые описывают существующие между классами отношения использования (включая отношения уточнения, трассировки и связывания);
- ✓ **обобщения**, связывающие обобщенные классы со специализированными;
- ✓ **ассоциации**, представляющие структурные отношения между объектами.

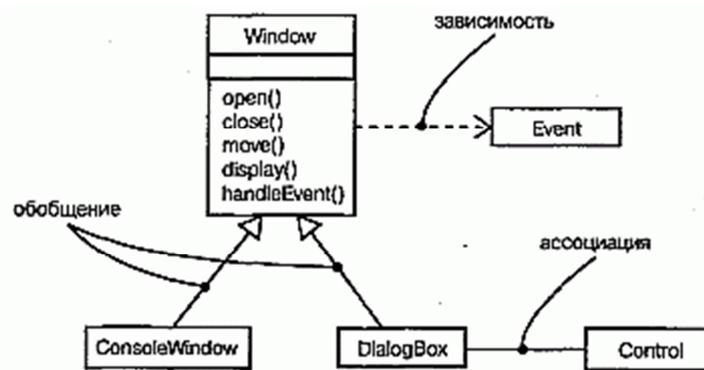


Рис. 3.19. Отношения

Зависимости

Зависимость (Dependency) называют отношение использования, согласно которому изменение в спецификации одного элемента (например, класса Event) может повлиять на другой элемент, его использующий (в данном случае - класс Window), причем обратное не обязательно. Графически зависимость изображается пунктирной линией со стрелкой, направленной от данного элемента на тот, от которого он зависит. Используйте зависимости, когда хотите показать, что один элемент использует другой.

Чаще всего зависимости применяются при работе с классами, чтобы отразить в сигнатуре операции тот факт, что один класс использует другой в качестве аргумента (см. рис. 3.20).

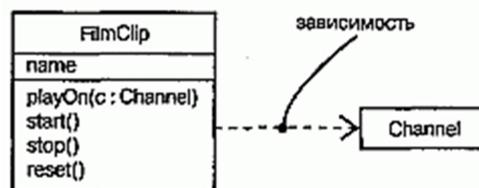


Рис. 3.20. Зависимости

Обобщения

Обобщение (Generalization) - это отношение между общей сущностью (суперклассом, или родителем) и ее конкретным воплощением (субклассом, или потомком). Обобщения иногда называют отношениями типа "является", имея в виду, что одна сущность (например, класс BayWindow) является частным выражением другой, более общей (скажем, класса Window). Обобщение означает, что объекты класса-потомка могут использоваться всюду, где встречаются объекты класса-родителя, но не наоборот. Другими словами, потомок может быть подставлен вместо родителя. При этом он наследует свойства родителя, в частности его атрибуты и операции. Часто, хотя и не всегда, у потомков есть и свои собственные атрибуты и операции, помимо тех, что существуют у родителя. Операция потомка с той же сигатурой, что и у родителя,

замещает операцию родителя; это свойство называют *полиморфизмом* (Polymorphism). Графически отношение обобщения изображается в виде линии с большой незакрашенной стрелкой, направленной на родителя, как показано на рис. 3.21. Применяйте обобщения, когда хотите показать отношения типа "родитель/потомок".

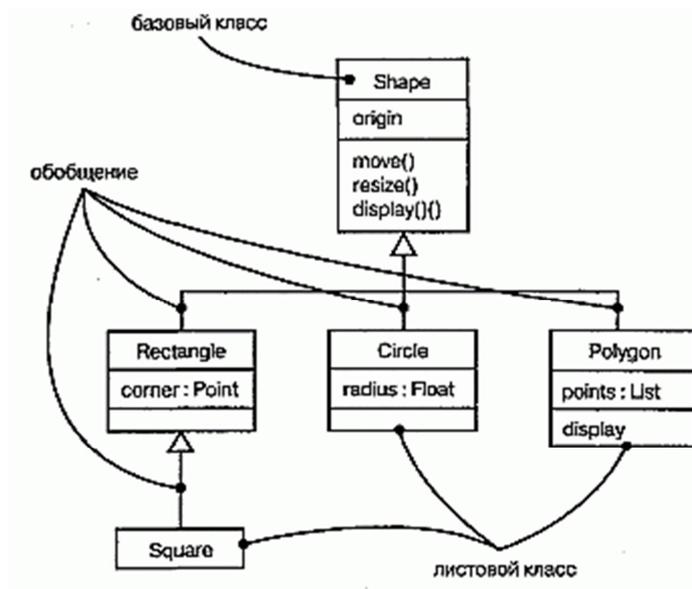


Рис. 3.21. Обобщение

Класс может иметь одного или нескольких родителей или не иметь их вовсе. Класс, у которого нет родителей, но есть потомки, называется **базовым** (base) или **корневым** (root), а тот, у которого нет потомков, - листовым (leaf). О классе, у которого есть только один родитель, говорят, что он использует одиночное наследование (Single inheritance); если родителей несколько, речь идет о множественном наследовании (Multiple inheritance).

Ассоциации

Ассоциацией (Association) называется структурное отношение, показывающее, что объекты одного типа неким образом связаны с объектами другого типа. Если между двумя классами определена ассоциация, то можно перемещаться от объектов одного класса к объектам другого. Вполне допустимы случаи, когда оба конца ассоциации относятся к одному и тому же классу. Это означает, что с объектом некоторого класса позволительно связать другие объекты из того же класса. Ассоциация, связывающая два класса, называется бинарной. Можно, хотя это редко бывает необходимым, создавать ассоциации, связывающие сразу несколько классов; они называются n-арными. Графически ассоциация изображается в виде линии, соединяющей класс сам с собой или с другими классами.



Рис. 3.24. Кратность

На рис.3.25 ассоциация связывает один объект класса «Набор товаров» с одним или более объектами класса «товар». Связаны между собой могут быть и объекты одного класса, поэтому ассоциация может связывать класс с самим собой. Ассоциация с названием «Включает» показывает, что набор может включать несколько различных товаров. В данном случае направленная ассоциация позволяет найти все виды товаров, входящие в набор, но не дает ответа на вопрос, входит ли товар данного вида в какой-либо набор.

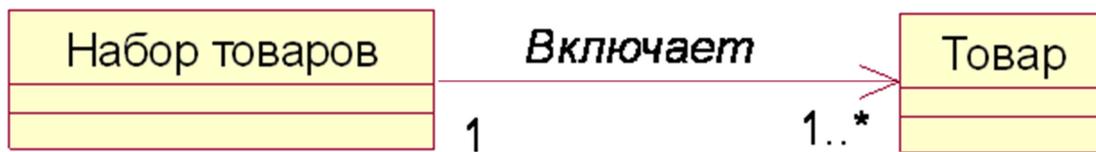


Рис. 3.25. Применение ассоциаций в UML

Агрегирование. Простая ассоциация между двумя классами отражает структурное отношение между равноправными сущностями, когда оба класса находятся на одном концептуальном уровне и ни один не является более важным, чем другой. Но иногда приходится моделировать отношение типа "часть/целое", в котором один из классов имеет более высокий ранг (целое) и состоит из нескольких меньших по рангу (частей). Отношение такого типа называют агрегированием; оно причислено к отношениям типа "имеет" (с учетом того, что объект-целое имеет несколько объектов-частей). Агрегирование является частным случаем ассоциации и изображается в виде простой ассоциации с незакрашенным ромбом со стороны "целого", как показано на рис. 3.26.

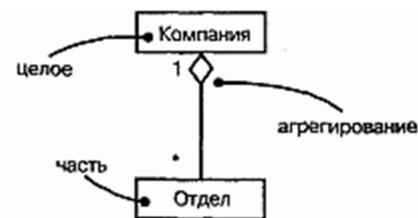


Рис. 3.26. Агрегирование

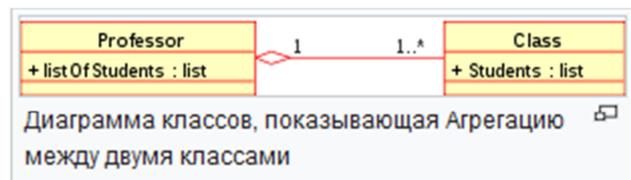


Рис. 3.27. Пример агрегации

Как уже говорилось ранее, UML позволяет строить модели с различным уровнем детализации. На рис.3.28. показана детализация модели, представленной на рис.3.18.

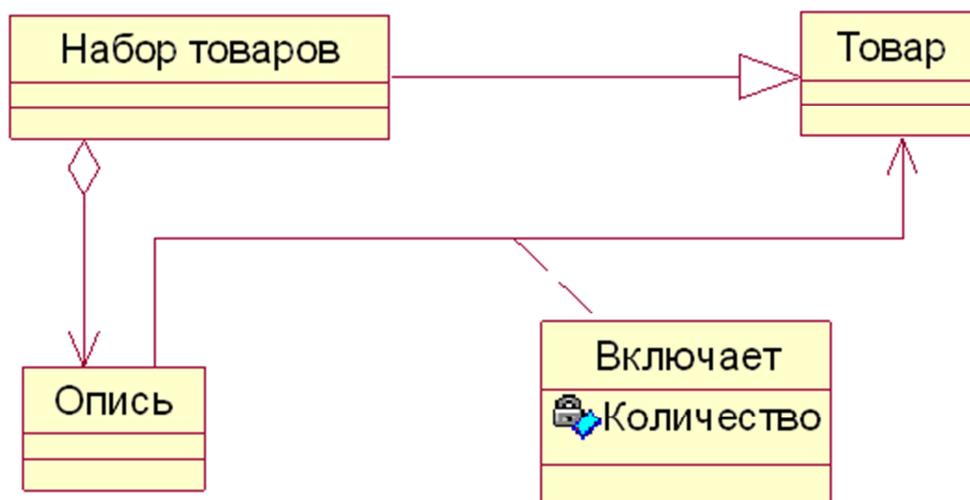


Рис. 3.28. Детализация модели набора товаров

Обобщение показывает, что набор товаров – это тоже товар, который может быть предметом заказа, продажи, поставки и т.д. Набор включает опись, в которой указывается, какие товары входят в набор, а класс-ассоциация «включает» определяет количество каждого вида товаров в наборе.

Стереотипы классов

При создании диаграмм классов часто пользуются понятием «стереотип». Далее по тексту мы будем говорить именно о стереотипах классов. **Стереотип** класса – это элемент расширения словаря UML, который обозначает отличительные особенности в использовании класса. Стереотип имеет название, которое задается в виде текстовой строки. При изображении класса на диаграмме стереотип показывается в верхней части класса в двойных угловых скобках. Есть четыре стандартных стереотипа классов, для которых предусмотрены специальные графические изображения (см. рис. 3.29).

Стереотип используется для обозначения классов-сущностей (классов данных), стереотип описывает пограничные классы, которые являются посредниками между программной системой и внешними по отношению к ней сущностями – актерами, обозначаемыми стереотипом $\langle \rangle$. Наконец, стереотип описывает классы и объекты, которые управляют взаимодействиями. Применение стереотипов позволяет, в частности, изменить вид диаграмм классов.

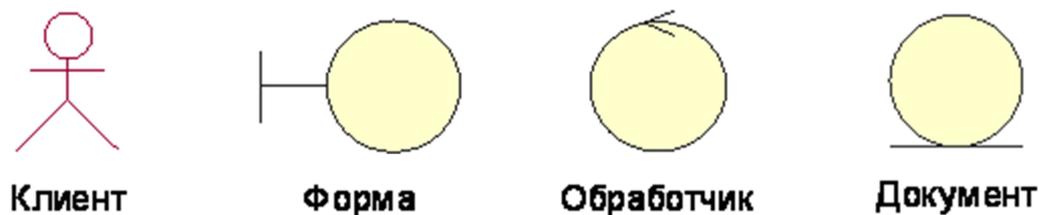


Рис.3.29. Стереотипы для классов

Применение диаграмм классов

Диаграммы классов создаются при логическом моделировании программных систем и служат для следующих целей:

- ✓ Для моделирования данных. Анализ предметной области позволяет выявить основные характерные для нее сущности и связи между ними. Это удобно моделируется с помощью диаграмм классов. Эти диаграммы являются основой для построения концептуальной схемы базы данных.
- ✓ Для представления архитектуры ПС. Можно выделить архитектурно значимые классы и показать их на диаграммах, описывающих архитектуру ПС.
- ✓ Для моделирования навигации экранов. На таких диаграммах показываются пограничные классы и их логическая взаимосвязь. Информационные поля моделируются как атрибуты классов, а управляющие кнопки – как операции и отношения.
- ✓ Для моделирования логики программных компонент (будет описано в последующих статьях).
- ✓ Для моделирования логики обработки данных.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1. Моделирование простой зависимости

Самым распространенным видом отношения зависимости является соединение между классами, когда один класс использует другой в качестве параметра операции.

Для моделирования такого отношения изобразите зависимость, направленную от класса с операцией к классу, используемому в качестве ее параметра.

Например, на рис. 3.30 показано несколько классов, взятых из системы, управляющей распределением студентов и преподавателей на университетских курсах. Зависимость направлена от класса РасписаниеЗанятий к классу Курс, поскольку последний используется в операциях add и remove класса РасписаниеЗанятий.

1. Средствами MS Visio постройте диаграмму классов с простой зависимостью, показанную на рис. 3.30.

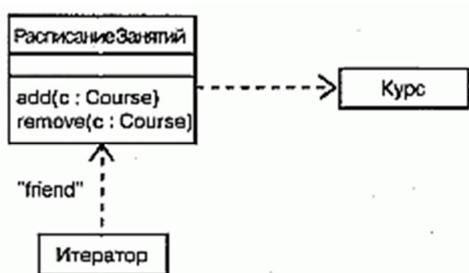


Рис. 3.30. Отношения зависимости

Задание 2. Моделирование одиночного наследования

Моделируя словарь системы, вам часто придется работать с классами, похожими на другие по структуре и поведению. В принципе их можно моделировать как различные, независимые друг от друга абстракции. Но лучше выделить одинаковые свойства и сформировать на их основе общие классы, которым наследуют специализированные.

Моделирование отношений наследования осуществляется в таком порядке:

1. Найдите атрибуты, операции и обязанности, общие для двух или более классов из данной совокупности
2. Вынесите эти элементы в некоторый общий класс (если надо, создайте новый, но следите, чтобы уровней не оказалось слишком много).
3. Отметьте в модели, что более специализированные классы наследуют более общим, включив отношение обобщения, направленное от каждого потомка к его родителю.

На рис. 3.31 видно несколько классов, взятых из приложения по организации работы трейдеров. Здесь показано отношение обобщения, которое от четырех классов - РасчетныйСчет, Акция, Облигация и Собственность - направлено к более общему классу ЦенныеБумаги. Он является родителем, а остальные - его потомками. Каждый специализированный класс - это частный случай класса ЦенныеБумаги. Обратите внимание, что в классе ЦенныеБумаги есть две операции - presentValue (текущаяСтоимость) и history (история). Это значит, что все его потомки наследуют данные операции, а заодно и все остальные атрибуты и операции родителя, которые могут не изображаться на рисунке.

Иерархия "обобщение/специализация" не обязательно ограничивается двумя уровнями. Как видно из рисунка, вполне допустимо существование более двух уровней наследования. АкцияСМалымКапиталом и АкцияСБольшимКапиталом - потомки класса Акция, который, в свою очередь, является потомком класса ЦенныеБумаги. Последний является базовым классом, поскольку не имеет родителей. Классы же АкцияСМалымКапиталом и АкцияСБольшимКапиталом - листовые, поскольку не имеют

потомков. Наконец, класс Акция имеет как родителей, так и потомков, а следовательно, не является ни листовым, ни базовым.

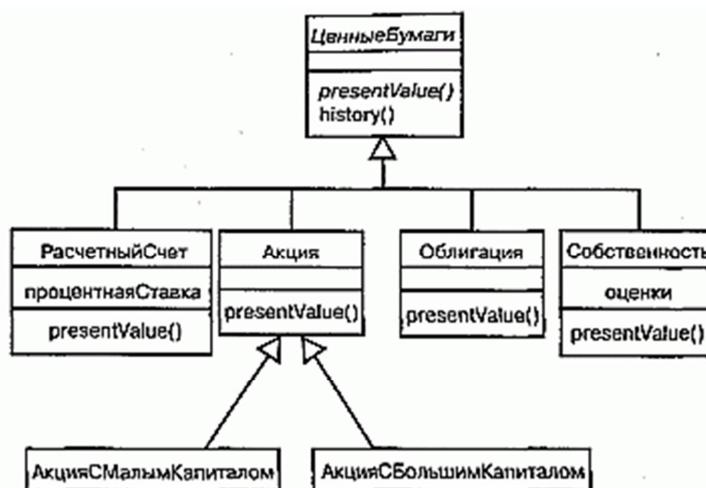


Рис. 3.31. Отношения наследования

2) Разработайте диаграмму с отношениями наследования, показанную на рис. 3.31.

Задание 3. Моделирование структурных отношений

Отношения зависимости и обобщения применяются при моделировании классов, которые находятся на разных уровнях абстракции или имеют различную значимость. Что касается отношения зависимости, один класс зависит от другого, но тот может ничего не "знать" о наличии первого. Когда речь идет об отношении обобщения, класс-потомок наследует своему родителю, но сам родитель о нем не осведомлен. Другими словами, отношения зависимости и обобщения являются односторонними.

Ассоциации предполагают участие равноправных классов. Если между двумя классами установлена ассоциация, то каждый из них каким-то образом зависит от другого, и навигацию можно осуществлять в обоих направлениях. В то время как зависимость - это отношение использования, а обобщение - отношение "является", ассоциации определяют структурный путь, обуславливающий взаимодействие объектов данных классов. По умолчанию ассоциации являются двунаправленными, но вы можете оставить только одно направление.

Моделирование структурных отношений производится следующим образом:

1. Определите ассоциацию для каждой пары классов, между объектами которых надо будет осуществлять навигацию. Это взгляд на ассоциации с точки зрения данных.
2. Если объекты одного класса должны будут взаимодействовать с объектами другого иначе, чем в качестве параметров операции, следует определить между этими классами ассоциацию. Это взгляд на ассоциации с точки зрения поведения.

3. Для каждой из определенных ассоциаций задайте кратность (особенно если она не равна *, то есть значению по умолчанию) и имена ролей (особенно если это помогает объяснить модель).

4. Если один из классов ассоциации структурно или организационно представляет собой целое в отношении классов на другом конце ассоциации, выглядящих как его части, пометьте такую ассоциацию как агрегирование.

На рис. 3.32 изображены классы, взятые из вузовской информационной системы. В нижней части диаграммы находятся классы Студент, Курс и Преподаватель. Между классами Студент и Курс существует ассоциация, показывающая, что студенты посещают курсы. Каждый студент может посещать любое число курсов, и на каждый курс может приходиться любое количество студентов.



Рис. 3.32. Структурные отношения

Аналогичным образом между классами Курс и Преподаватель определена ассоциация, показывающая, что преподаватель читает курс. Для каждого курса должен быть хотя бы один преподаватель, и каждый преподаватель может вести любое количество курсов (в том числе и ни одного) [14].

Отношения между классом Вуз и классами Студент и Факультет слегка отличаются друг от друга, хотя оба являются отношениями агрегирования. В вузе может быть любое количество студентов (включая ноль), и каждый студент может обучаться в одном или нескольких вузах; вуз может состоять из одного или нескольких факультетов, но каждый факультет принадлежит одному и только одному вузу. В принципе можно было бы обойтись без отношений агрегирования и использовать простые ассоциации, определяя при этом Вуз как целое, а Студента и Факультет - как его части. Тем самым вы ясно показываете, какой из классов организационно стоит выше остальных. Так, вузы в какой-то степени определяются своими студентами и факультетами. В то же время студенты и факультеты вообще не могут существовать вне связи со своим вузом, и в какой-то мере он формирует их облик.

Видно также, что между классами Факультет и Преподаватель установлены две ассоциации. Одна из них показывает, что каждый преподаватель работает на одном или нескольких факультетах, и на каждом факультете должен быть по меньшей мере один преподаватель. Здесь мы имеем дело с агрегированием, так как организационно факультеты находятся на более высоком уровне вузовской структуры, чем преподаватели. Другая ассоциация показывает, что каждым факультетом управляет только один преподаватель - декан. Согласно данной модели, преподаватель может быть деканом только одного факультета, причем некоторые преподаватели не являются деканами.

3) Разработайте диаграмму структурных отношений, показанную на рис. 3.32.

Задание 4. Описание классов и атрибутов

1) Описать диаграмму классов, представленную на рис. 3.33. Описать какие классы содержит диаграмма, атрибут и операции каждого класса. А также все связи между классами, которые имеются на данной диаграмме.

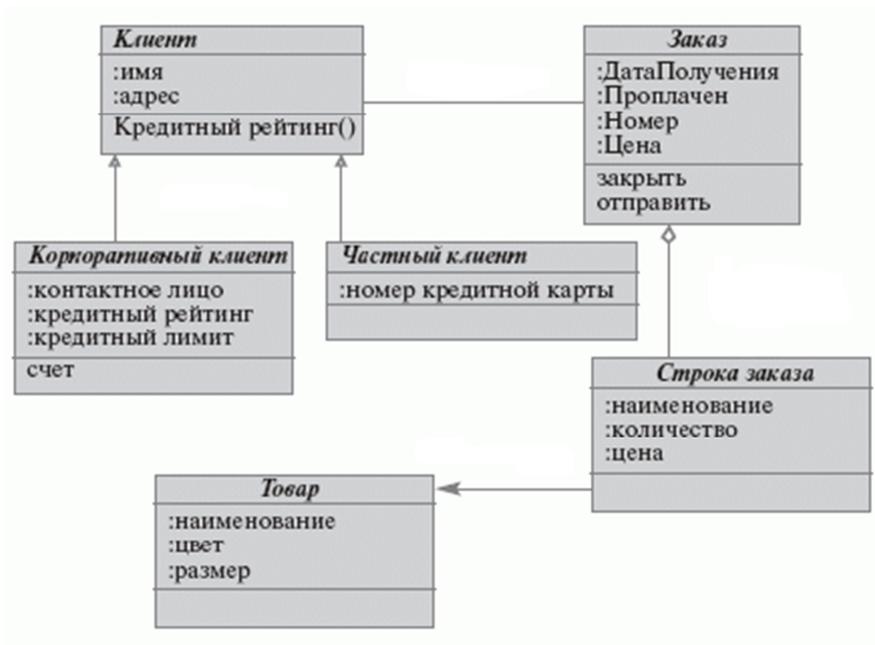


Рис. 3.33. Диаграмма классов

2) Постройте диаграмму, показанную на рис. 3.33, средствами пакета MS Visio.

Задание 5.

1) Постройте диаграмму, показанную на рис. 3.34, средствами пакета MS Visio.

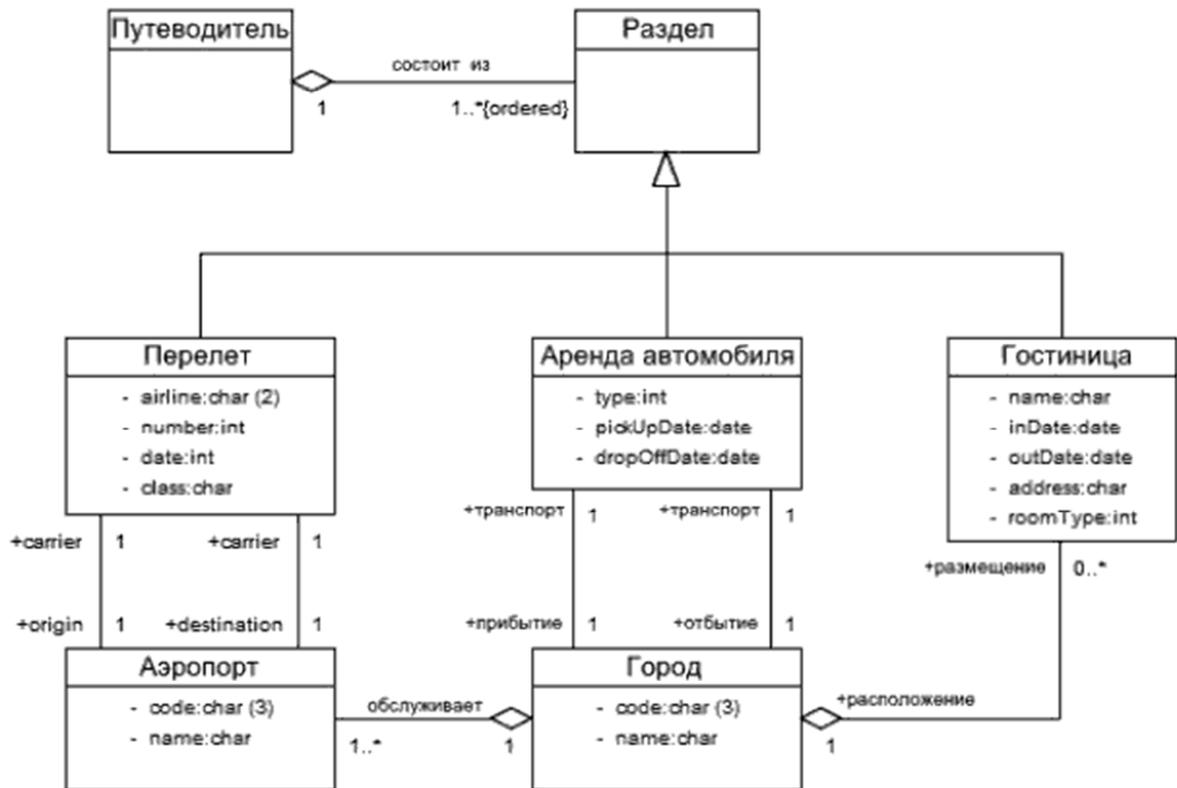


Рис. 3.34. Диаграмма классов «Путешествие» [19]

Задание 6.

Разработайте собственный пример диаграммы классов на основе теоретического материала и примеров, рассмотренных в заданиях 1-5.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

- 1) Что описывает диаграмма классов?
- 2) Для чего используются диаграммы классов?
- 3) Какие точки зрения на построение диаграмм классов существуют?
- 4) Какие отношения между классами существуют в UML?
- 5) Как обозначаются и для чего применяются отношения зависимости?
- 6) Как обозначаются и для чего применяются отношения обобщения?
- 7) Как обозначаются и для чего применяются отношения ассоциации?
- 8) Какие дополнения применимы к ассоциациям?
- 9) Как обозначаются и для чего применяется агрегирование?
- 10) Что понимают под стереотипами класса?

ЛАБОРАТОРНАЯ РАБОТА № 12

Диаграммы деятельности в UML

Цель работы: изучение синтаксиса, семантики и получение навыков разработки диаграмм деятельности при помощи UML.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Как правило, для моделирования бизнес-процессов, технологических процессов, последовательных и параллельных вычислений используется диаграмма деятельности. **Диаграмма деятельности (Activity diagram)** — диаграмма, на которой показано разложение некоторой *деятельности* на её составные части [20].

Каждая диаграмма деятельности должна иметь единственное начальное и конечное состояния. При этом каждая деятельность начинается в начальном состоянии и заканчивается в конечном состоянии. Саму диаграмму деятельности принято располагать таким образом, чтобы действия следовали сверху вниз или слева направо. В этом случае начальное состояние будет изображаться в верхней или левой части диаграммы, а конечное - в ее нижней или правой части. В интересах удобства визуального представления на диаграмме деятельности допускается изображать несколько конечных состояний. В этом случае все их принято считать эквивалентными друг другу. Если из состояния действия выходит единственный переход, то его можно никак не помечать. Если же таких переходов несколько, то при моделировании последовательной деятельности запускается только один из них. При этом для всех выходящих из некоторого состояния деятельности переходов должно выполняться требование истинности только одного из них. Подобный случай встречается тогда, когда последовательно выполняемая деятельность должна разделиться на альтернативные ветви в зависимости от значения промежуточного результата. Такая ситуация получила название ветвления, а для ее обозначения применяется специальный символ решения [20].

Графически ветвление на диаграмме деятельности обозначается символом решения (decision), изображаемого в форме *небольшого ромба*, внутри которого нет никакого текста (рис. 3.35 вверху). В этот ромб может входить только одна стрелка от того состояния действия, после выполнения которого поток управления должен быть продолжен по одной из взаимно исключающих ветвей. Принято входящую стрелку присоединять к верхней или левой вершине символа решения. Выходящих стрелок может

быть две или более, но для каждой из них явно указывается соответствующее сторожевое условие в форме булевского выражения.

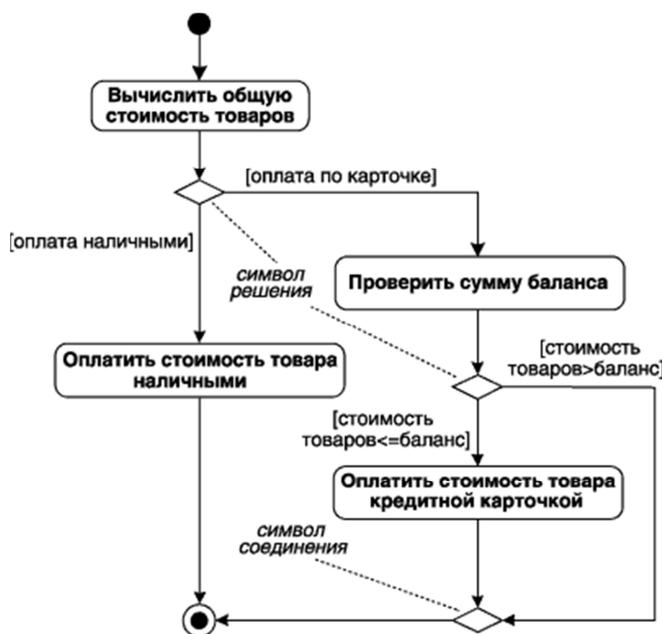


Рис. 3.35. Различные варианты ветвлений на диаграмме деятельности

Для *графического объединения альтернативных ветвей* на диаграмме деятельности рекомендуется также использовать *аналогичный символ в форме ромба*, который в этом случае называют соединением (*merge*). Наличие этого символа, внутри которого также не записывается никакого текста, упрощает визуальный контроль логики выполнения процедурных действий на диаграмме деятельности (рис. 7.2. внизу). Входящих стрелок у символа соединения может быть несколько, они исходят от состояний действия, принадлежащих к одной из взаимно исключающих ветвей. Выходить из ромба соединения может только одна стрелка, при этом ни входящие, ни выходящая стрелки не должны содержать сторожевых условий. Исключением является ситуация, когда с целью сокращения диаграммы объединяют символ решения с символом соединения. Нарушение этих правил делает диаграмму деятельности несостоятельной (*ill formed*).

Диаграмма деятельности на рис. 3.35 моделирует ситуацию, возникающую в супермаркетах при оплате товаров. Как правило, заплатить за покупки можно либо наличными, либо по кредитной карточке. Если покупателем выбран вариант оплаты по кредитной карточке, то проверяется сумма баланса предъявленной к оплате кредитной карточки. При этом оплата происходит только в том случае, если общая стоимость приобретаемых товаров не превышает суммы баланса этой карточки. В противном случае оплаты не происходит, и товар остается у продавца [20].

Обычно распараллеливание вычислений существенно повышает общее быстродействие программных систем, поэтому необходимы графические примитивы для представления параллельных процессов. В диаграммах деятельности с этой целью используется специальный символ для *разделения и слияния параллельных вычислений* или потоков управления. Это прямая черточка, аналогичная обозначению параллельных переходов для диаграмм состояний. На диаграммах деятельности такая черточка изображается отрезком горизонтальной, реже - вертикальной, линии, толщина которой несколько шире линий простых переходов диаграммы деятельности. При этом *разделение (fork)* имеет один входящий переход и несколько выходящих (рис. 3.35 а), которые изображаются отрезками вертикальных, реже - горизонтальных, линий. *Слияние (join)*, наоборот, имеет несколько входящих переходов и один выходящий (рис. 3.35 б). Параллельные переходы на диаграмме деятельности можно изображать в удлиненной форме, а входящие и выходящие переходы вертикальными стрелками.

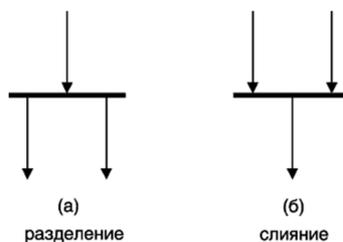


Рис. 3.35. Графическое изображение разделения и слияния параллельных потоков управления на диаграмме деятельности

Рассмотренных переходов оказывается достаточно для моделирования различных по сложности ситуаций. Для иллюстрации особенности изображения ветвления и параллельных деятельностей можно рассмотреть пример регистрации пассажиров в аэропорту (рис. 3.36) [20].

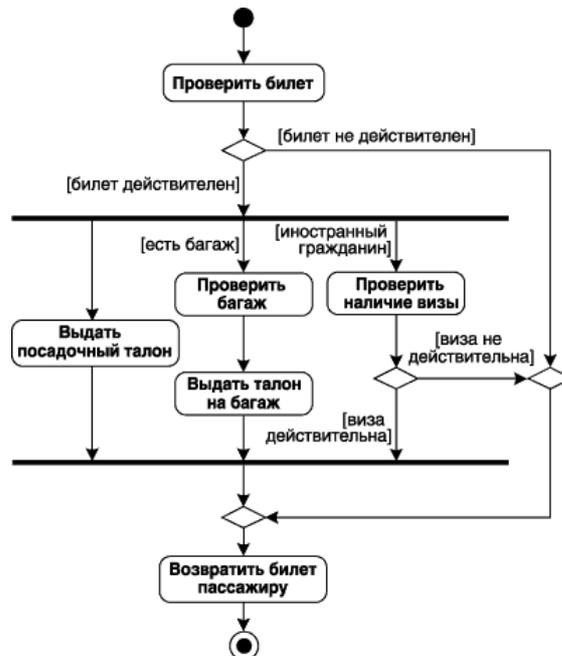


Рис. 3.36. Диаграмма деятельности для примера регистрации пассажиров в аэропорту

Первоначально выполняется деятельность по проверке билета. В случае если билет не действителен, он возвращается пассажиру, при этом никаких дополнительных действий не выполняется. Если же билет действителен, то пассажиру выдается посадочный талон. В дополнение к этому проверяется гражданство и наличие багажа у пассажира. Если есть багаж, то его проверка может быть выполнена параллельно, по результатам которой пассажиру выдается талон на багаж. Если пассажир является иностранным гражданином, то дополнительно проверяется наличие у него визы. Если виза действительна, то проверка завершается успешно, и пассажир с возвращенным ему билетом может проследовать на посадку.

Если же виза окажется не действительной, то для этого пассажира посадка на данный рейс оказывается невозможной. В этом случае ему не выдается посадочный талон и талон на багаж, в случае его наличия, поскольку происходит прекращение всех выполняемых сотрудниками аэропорта действий [20].

Дорожки

Как правило, применительно к бизнес-процессам желательно выполнение каждого действия ассоциировать с конкретным подразделением компании. В этом случае подразделение будет нести ответственность за реализацию определенных действий, а сам бизнес-процесс представляется в виде переходов действий из одного подразделения к другому. Для моделирования этих особенностей в языке UML предложена специальная конструкция, получившая название дорожки.

Дорожка (swimlane) - графическая область диаграммы деятельности, содержащая элементы модели, ответственность за выполнение которых принадлежит отдельным подсистемам [20].

В данном случае имеется в виду визуальная аналогия с плавательными дорожками в бассейне, если смотреть на соответствующую диаграмму деятельности сверху. При этом все состояния на диаграмме деятельности делятся на группы, разграниченные вертикальными линиями. Две соседних линии и образуют дорожку, а группа состояний между этими линиями выполняется организационным подразделением (отделом, группой, отделением, филиалом) или сотрудником компании (рис. 3.37). В последнем случае принято указывать должность сотрудника, ответственного за выполнение определенных действий.

Названия подразделений или должностей явно указываются в верхней части дорожки. Пересекать линию дорожки могут только переходы, которые в этом случае обозначают выход или вход потока управления в соответствующее подразделение компании. Порядок следования дорожек не несет какой-либо семантической информации и определяется соображениями удобства.



Рис. 3.37. Вариант диаграммы деятельности с дорожками

В качестве примера рассмотрим фрагмент диаграммы деятельности торговой компании, обслуживающей клиентов в форме заказов. Подразделениями компании обычно являются отдел приема и оформления заказов, отдел продаж и склад. Этим подразделениям будут соответствовать три дорожки на диаграмме деятельности, каждая из которых специфицирует зону ответственности подразделения. В этом случае диаграмма деятельности включает в себе не только информацию о последовательности выполнения рабочих действий, но и о том, какое подразделение торговой компании должно выполнять, то или иное действие (рис. 3.38). Из указанной диаграммы деятельности видно, что после принятия заказа от клиента отделом приема и оформления заказов осуществляется распараллеливание деятельности на два потока (переход-разделение). Первый из них остается в этом же отделе и связан с получением оплаты от клиента за заказанный товар. Второй инициирует выполнение действия по регистрации

заказа в отделе продаж (модель товара, размеры, цвет, год выпуска и пр.). Однако выдача товара со склада начинается только после того, как будет получена от клиента оплата за товар (переход-слияние). Затем выполняется подготовка товара к отправке и его отправка клиенту в отделе продаж. После завершения этих действий заказ закрывается в отделе приема и оформления заказов [11].

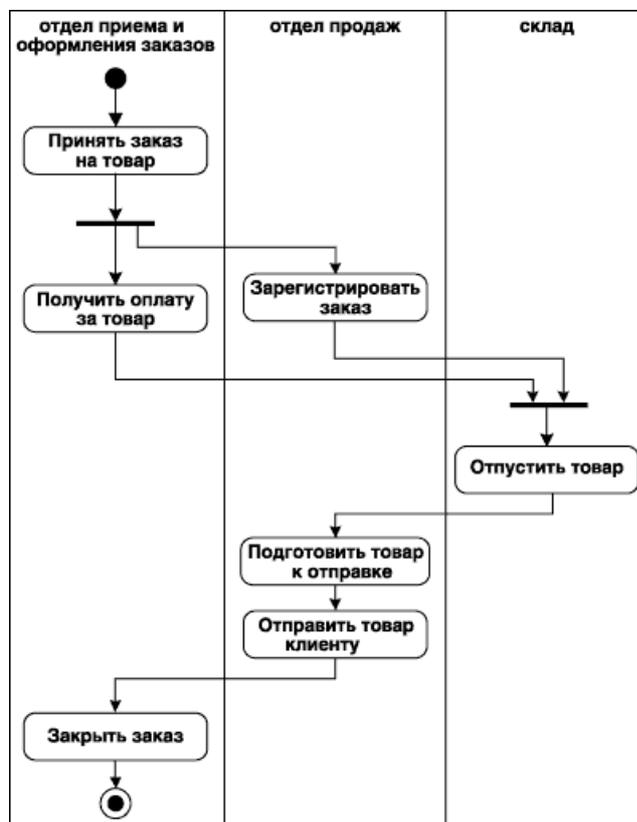


Рис. 3.38. Фрагмент диаграммы деятельности для торговой компании

Объекты на диаграмме деятельности

Действия на диаграмме деятельности могут производиться над теми или иными объектами. Эти объекты либо инициируют выполнение действий, либо определяют результат этих действий. При этом действия специфицируют вызовы, которые передаются от одного объекта графа деятельности другому. Поскольку в таком ракурсе объекты играют определенную роль в понимании процесса деятельности, иногда возникает необходимость явно указать их на диаграмме деятельности [20].

Базовым графическим представлением объекта в нотации языка UML является **прямоугольник класса**, с тем отличием, что имя объекта подчеркивается. На диаграммах деятельности после имени может указываться характеристика состояния объекта в прямых скобках. Такие прямоугольники объектов присоединяются к состояниям действия отношением зависимости пунктирной линией со стрелкой. Соответствующая зависимость

определяет состояние конкретного объекта после выполнения предшествующего действия.

На диаграмме деятельности с дорожками расположение объекта может иметь дополнительный смысл. А именно, если объект расположен на границе двух дорожек, то это может означать, что переход к следующему состоянию действия в соседней дорожке ассоциирован с нахождением документа в некотором состоянии. Если же объект расположен внутри дорожки, то и состояние этого объекта целиком определяется действиями данной дорожки [20].

Применительно к диаграммам деятельности объекты, как правило, являются экземплярами классов сущностей или бизнес - сущностей. Стоит также заметить, что на диаграмме деятельности один и тот же объект может быть изображен несколько раз, при этом для исключения несогласованности диаграммы необходимо указывать для них различные характеристики состояния.

В предыдущем примере с торговой компанией центральным объектом процесса продажи является заказ или вернее состояние его выполнения. Вначале до обращения клиента заказ как объект отсутствует и возникает лишь после контакта с клиентом. В результате фиксируется полученный заказ, после чего он регистрируется в отделе продаж. Затем он передается на склад, где после получения оплаты за товар оформляется окончательно. Наконец, после того, как товар отправлен клиенту, эта информация вносится в заказ, и он считается выполненным. Эта информация может быть представлена графически в виде модифицированного варианта диаграммы деятельности торговой компании (рис. 3.39).

Достоинством диаграммы деятельности является возможность визуализировать отдельные аспекты поведения рассматриваемой системы или реализации отдельных операций классов в виде процедурной последовательности действий. Таким образом, полная модель системы может содержать одну или несколько диаграмм деятельности, каждая из которых описывает последовательность реализации либо наиболее важных вариантов использования (типичный ход событий и все исключения), либо нетривиальных операций классов [20].

В заключение следует заметить, что диаграмма деятельности, так же как и другие виды канонических диаграмм, не содержат средств выбора оптимальных вариантов конфигурации собственно диаграмм. При разработке сложных проектов проблема выбора оптимальных решений применительно к диаграммам деятельности становится весьма актуальной. Рациональное расходование средств, затраченных на разработку и

эксплуатацию системы, повышение ее производительности и надежности зачастую определяют конечный результат всего проекта [20].

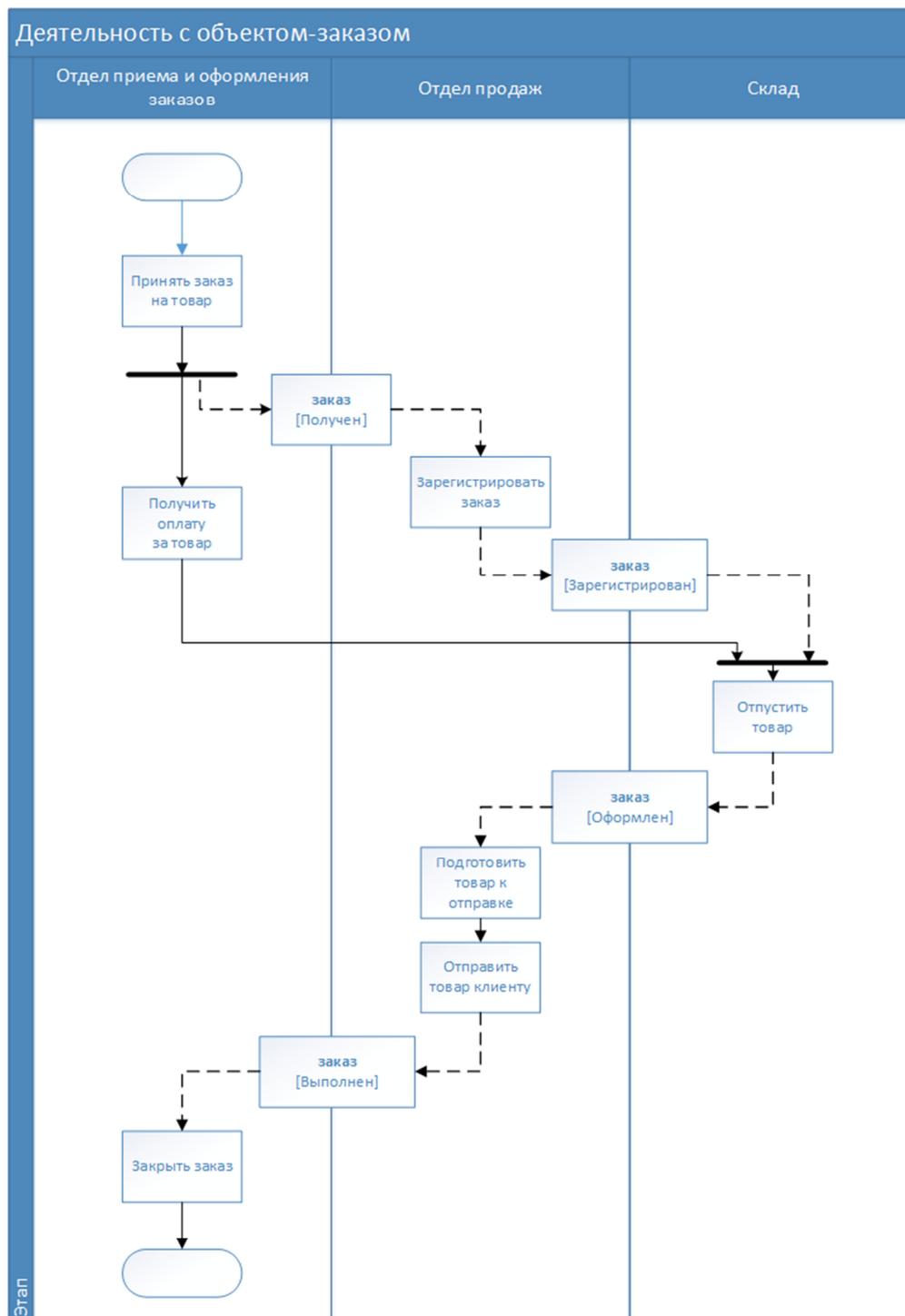


Рис. 3.39. Фрагмент диаграммы деятельности торговой компании с объектом-заказом

- 1) Запустить программу MS Visio;
- 2) Для того, чтобы загрузить нужную панель инструментов для разработки диаграмм при создании документа в поле «Загрузить шаблоны из Интернета» укажите «UML»
- 3) В найденных пунктах выберите «UML. Деятельность».

Задание 1. Разработать диаграммы деятельности

На основе представленного материала разработать диаграммы деятельности, показанные на рис. 7.4, 7.6 и 7.7

Задание 2. Разработать диаграмму деятельности «Приготовление блюда»

Используя правила графической нотации языка UML, опишите деятельность по приготовлению вашего любимого блюда.

Задание 3. Разработать диаграмму деятельности «Работа информационной системы отправки отчетности в ФНС»

Всем известно, что в настоящее время существует крайне удобный сервис отправки налоговой отчетности в Федеральную налоговую службу (ФНС) посредством сети Интернет. Он существенно сократил время ожидания сдачи документов, уменьшил очереди и разгрузил работников службы. Ниже приведено тактовое описание работы информационной системы по отправке отчетности в ФНС. На основании этого описания постройте диаграмму деятельности UML.

Тестовое описание работы информационной системы отправки отчетности

Налоговая декларация формируется и загружается на сайт, затем отправляется в налоговый орган. При этом фиксируется дата отправки и приходит подтверждения даты отправки документов. Если документ получен налоговым органом, то приходит налогоплательщику отправляется извещение о получении электронного документа. Если документ не получен налоговым органом, то отправляется сообщение об ошибке и операция отправки повторяется. После того, как документы получены начинается проверка налоговой декларации. Если декларация заполнена правильно, то формируется квитанция о приеме и отправляется налогоплательщику. Документы из декларации заносятся в хранилище электронных документов налогового органа. Налогоплательщику отправляется уведомление об успешном вводе данных декларации. Процесс завершается.

Если декларация была заполнена неправильно, рассматриваются два условия: а) корректировка возможна б) корректировка невозможна. В случае если корректировка возможна, происходит запрос корректировки, отправляется квитанция о получении корректировки и происходит повторный процесс проверки декларации.

В случае если корректировка невозможна формируется уведомление об отказе и отправляется налогоплательщику. Процесс завершается.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

- 1) Как расшифровывается аббревиатура UML?
- 2) Как графически изобразить начало и конец процесса на диаграмме деятельности?
- 3) Каков смысл использования дорожек на диаграммах деятельности UML?
- 4) Сколько начальных и конечных состояний может иметь диаграмма деятельности в UML?
- 5) Каким образом изобразить ветвление (выбор альтернативы) на диаграмме деятельности?
- 6) Что является базовым графическим представлением объекта в нотации языка UML?
- 7) Каким образом, использование нотации UML могло бы помочь в решении проблемы, показанной на рис. 7.1?
- 8) В чем состоит недостаток диаграмм деятельности?
- 9) Чем отличается изображение объекта от прямоугольника класса на диаграммах деятельности?

ЛАБОРАТОРНАЯ РАБОТА № 13

Диаграммы последовательностей в UML

Цель работы: изучение синтаксиса и семантики, получение навыков разработки диаграмм последовательностей при помощи UML.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Диаграмма последовательности (sequence diagram) - диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления [21].

Особенности взаимодействия элементов моделируемой системы могут быть представлены на диаграммах кооперации и последовательности. Диаграммы кооперации используются для спецификации динамики поведения систем, хотя время в явном виде в них отсутствует. Однако временной аспект поведения может иметь существенное значение при моделировании синхронных процессов, описывающих взаимодействие объектов. Именно для этой цели в языке UML используются диаграммы последовательности [21].

На диаграмме последовательности неявно присутствует ось времени, что позволяет визуализировать временные отношения между передаваемыми сообщениями. С помощью

диаграммы последовательности можно представить взаимодействие элементов модели как своеобразный временной график "жизни" всей совокупности объектов, связанных между собой для реализации варианта использования программной системы, достижения бизнес-цели или выполнения какой-либо задачи [13].

Изображение объектов на диаграмме последовательности

На диаграмме последовательности, как и на диаграммах рассмотренных ранее, изображаются объекты, которые непосредственно участвуют во взаимодействии, при этом никакие статические связи с другими объектами не визуализируются. Для диаграммы последовательности ключевым моментом является именно динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы *два измерения*. Одно - слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии. Второе измерение диаграммы последовательности - вертикальная временная ось, направленная сверху вниз [21].

Каждый объект графически изображается в форме прямоугольника и располагается в верхней части своей линии жизни (рис. 3.40). Внутри прямоугольника записываются собственное имя объекта со строчной буквы и имя класса, разделенные двоеточием. При этом вся запись подчеркивается, что является признаком объекта, который, как указывалось ранее, представляет собой экземпляр класса [13].

Если на диаграмме последовательности отсутствует собственное имя объекта, то при этом должно быть указано имя класса. Такой объект считается *анонимным*. Может отсутствовать и имя класса, но при этом должно быть указано собственное имя объекта. Такой объект считается *сиротой*. Роль классов в именах объектов на диаграммах последовательности, как правило, не указывается [21].

Крайним слева на диаграмме изображается объект - инициатор моделируемого процесса взаимодействия (объект а на рис. 3.40). Правее - другой объект, который непосредственно взаимодействует с первым. Таким образом, порядок расположения объектов на диаграмме последовательности определяется исключительно соображениями удобства визуализации их взаимодействия друг с другом [21].

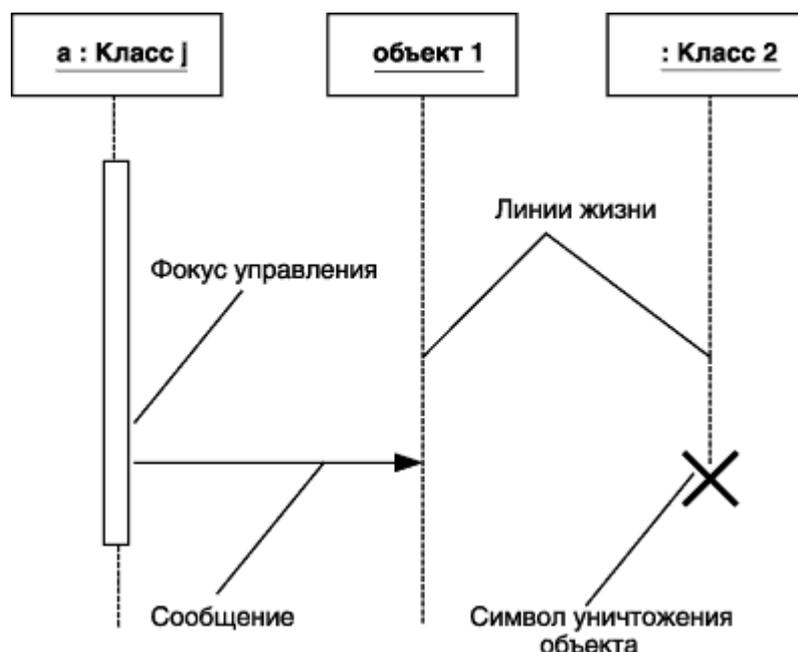


Рис. 3.40. Графические элементы диаграммы последовательности

Начальному моменту времени соответствует самая **верхняя часть** диаграммы. При этом процесс взаимодействия объектов реализуется посредством сообщений, которые посылаются одними объектами другим. Сообщения изображаются в виде горизонтальных стрелок с именем сообщения и образуют определенный порядок относительно времени своей инициализации. Другими словами, сообщения, расположенные на диаграмме последовательности выше, передаются раньше тех, которые расположены ниже. При этом масштаб на оси времени не указывается, поскольку диаграмма последовательности моделирует лишь временную упорядоченность взаимодействий типа "раньше-позже" [13].

Линия жизни объекта (object lifeline) - вертикальная линия на диаграмме последовательности, которая представляет существование объекта в течение определенного периода времени [21].

Линия жизни объекта изображается пунктирной вертикальной линией, ассоциированной с единственным объектом на диаграмме последовательности. Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях. Если объект существует в системе постоянно, то и его линия жизни должна продолжаться по всей рабочей области диаграммы последовательности от самой верхней ее части до самой нижней (объект 1 и анонимный объект Класса 2 на рис. 3.40) [21].

Отдельные объекты, закончив выполнение своих операций, могут быть уничтожены, чтобы освободить занимаемые ими ресурсы. Для таких объектов линия жизни обрывается в момент его уничтожения. Для обозначения момента уничтожения объекта в языке UML применяется специальный символ в форме латинской буквы "X". На рис. 3.41 этот символ

используется для уничтожения анонимного объекта, образованного от Класса 3. Ниже этого символа пунктирная линия не изображается, поскольку соответствующего объекта в системе уже нет, и этот объект должен быть исключен из всех последующих взаимодействий [21].

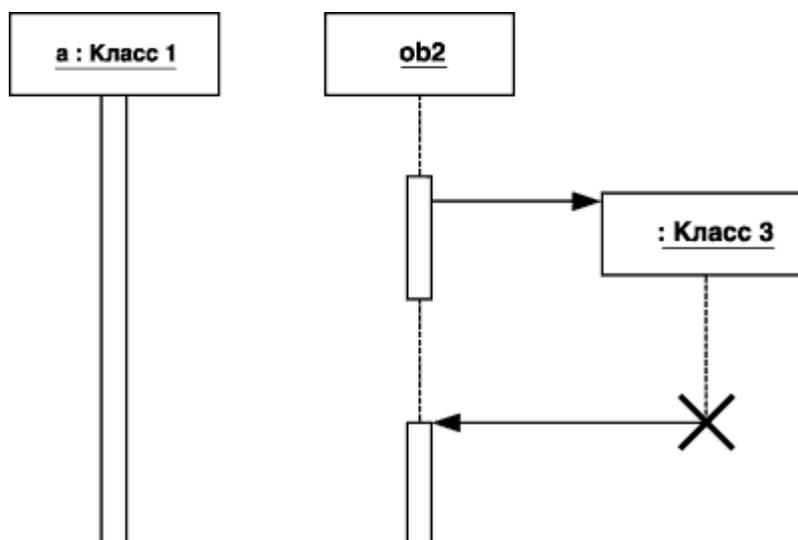


Рис. 3.41. Графическое изображение линий жизни и фокусов управления объектов

Не обязательно создавать все объекты в начальный момент времени. Отдельные объекты в системе могут создаваться по мере необходимости, существенно экономя ресурсы системы и повышая ее производительность. В этом случае прямоугольник такого объекта изображается не в верхней части диаграммы последовательности, а в той, которая соответствует моменту создания объекта (анонимный объект, образованный от Класса 3 на рис. 3.41). При этом прямоугольник объекта вертикально располагается в том месте диаграммы, которое по оси времени совпадает с моментом его возникновения в системе. Объект создается со своей линией жизни *a*, возможно, и с фокусом управления [21].

В процессе функционирования объектно-ориентированных систем одни объекты могут находиться в активном состоянии, непосредственно выполняя определенные действия, или в состоянии пассивного ожидания сообщений от других объектов. Фокус управления - символ, применяемый для того, чтобы явно выделить подобную активность объектов на диаграммах последовательности [21].

Фокус управления (focus of control) - специальный символ на диаграмме последовательности, указывающий период времени, в течение которого объект выполняет некоторое действие, находясь в активном состоянии [21].

Фокус управления изображается в форме вытянутого узкого прямоугольника (объект *a* на рис. 3.40), верхняя сторона которого обозначает начало получения фокуса управления объектом (начало активности), а ее нижняя сторона - окончание фокуса управления (окончание активности). Этот прямоугольник располагается ниже обозначения

соответствующего объекта и может заменять его линию жизни (объект а на рис. 3.41), если на всем ее протяжении он активен [21].

Периоды активности объекта могут чередоваться с периодами его пассивности или ожидания. В этом случае у такого объекта фокусы управления изменяют свое изображение на линию жизни и наоборот (объект сирота ob2 на рис.3.41). Важно понимать, что получить фокус управления может только объект, у которого в этот момент имеется линия жизни. Если же объект был уничтожен, то вновь возникнуть в системе он уже не может. Вместо него может быть создан лишь экземпляр этого же класса, который, строго говоря, будет другим объектом [21].

В отдельных случаях инициатором взаимодействия в системе может быть актер или внешний пользователь. При этом актер изображается на диаграмме последовательности самым первым объектом слева со своим фокусом управления (рис. 3.42). Наиболее часто актер и его фокус управления будут существовать в системе постоянно, отмечая характерную для пользователя активность в иницировании взаимодействий с системой. Актер может иметь собственное имя либо оставаться анонимным [21].

В отдельных случаях объект может посылать сообщения самому себе, иницируя так называемые **рефлексивные** сообщения. Для этой цели служит специальное изображение (сообщение у объекта а на рис. 3.42). Такие сообщения изображаются в форме сообщения, начало и конец которого соприкасаются с линией жизни или фокусом управления одного и того же объекта. Подобные ситуации возникают, например, при обработке нажатий на клавиши клавиатуры при вводе текста в редактируемый документ, при наборе цифр номера телефона абонента [21].

Если в результате рефлексивного сообщения создается новый подпроцесс или нить управления, то говорят о рекурсивном или вложенном фокусе управления. На диаграмме последовательности рекурсия обозначается небольшим прямоугольником, присоединенным к правой стороне фокуса управления того объекта, для которого изображается данное рекурсивное взаимодействие (анонимный объект **Класса 2** на рис. 3.42) [21].

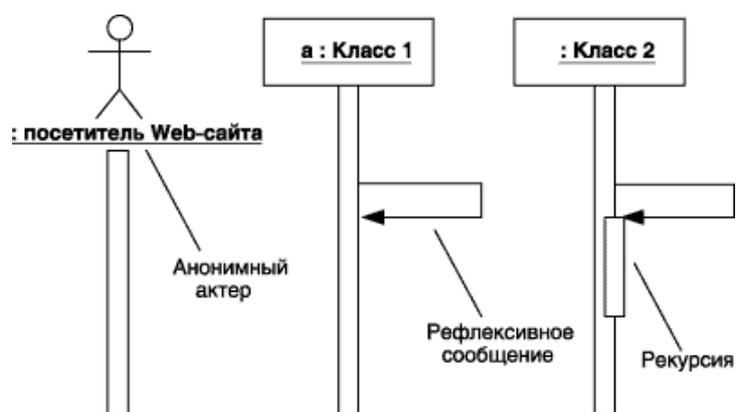


Рис. 3.42. Графическое изображение актера, рефлексивного сообщения и рекурсии на диаграмме последовательности

Сообщения на диаграмме последовательности

Как правило, сообщения в UML изображаются в виде стрелок с указателем. Стрелки сообщений изображаются аналогично рассмотренным ранее диаграммам, но применительно к диаграммам последовательности сообщения имеют дополнительные семантические особенности. При этом на диаграмме последовательности все сообщения упорядочены по времени своей передачи в моделируемой системе, хотя номера у них могут не указываться [21].

На диаграммах последовательности могут присутствовать три разновидности сообщений, каждое из которых имеет свое графическое изображение (рис. 3.43).

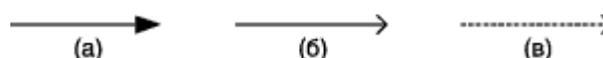


Рис. 3.43. Графическое изображение различных видов сообщений между объектами на диаграмме последовательности

Первая разновидность сообщения (рис. 3.43, а) наиболее распространена и используется для вызова процедур, выполнения операций или обозначения отдельных вложенных потоков управления. Начало этой стрелки, как правило, соприкасается с фокусом управления того объекта-клиента, который инициирует это сообщение. Конец стрелки соприкасается с линией жизни того объекта, который принимает это сообщение и выполняет в ответ определенные действия. При этом принимающий объект может получить фокус управления, становясь в этом случае активным. Передающий объект может потерять фокус управления или остаться активным [21].

Вторая разновидность сообщения (рис. 3.43 б) используется для обозначения простого асинхронного сообщения, которое передается в произвольный момент времени.

Передача такого сообщения обычно не сопровождается получением фокуса управления объектом-получателем [21].

Третья разновидность сообщения (рис. 3.43 в) используется для возврата из вызова процедуры. Примером может служить простое сообщение о завершении вычислений без предоставления результата расчетов объекту-клиенту. В процедурных потоках управления эта стрелка может быть опущена, поскольку ее наличие неявно предполагается в конце активизации объекта. В то же время считается, что каждый вызов процедуры имеет свою пару - возврат вызова. Для непроцедурных потоков управления, включая параллельные и асинхронные сообщения, стрелка возврата должна указываться явным образом [21].

Обычно сообщения изображаются горизонтальными стрелками, соединяющими линии жизни или фокусы управления двух объектов на диаграмме последовательности. При этом неявно предполагается, что время передачи сообщения достаточно мало по сравнению с процессами выполнения действий объектами. Считается также, что за время передачи сообщения с соответствующими объектами не может произойти никаких событий. Другими словами, состояния объектов не изменяются. Если же это предположение не может быть признано справедливым, то стрелка сообщения изображается под наклоном, так чтобы конец стрелки располагался ниже ее начала [21].

Каждое сообщение на диаграмме последовательности ассоциируется с определенной операцией, которая должна быть выполнена принявшим его объектом. При этом операция может иметь аргументы или параметры, значения которых влияют на получение различных результатов. Соответствующие параметры операции будет иметь и вызывающее это действие сообщение. Более того, значения параметров отдельных сообщений могут содержать условные выражения, образуя ветвление или альтернативные пути основного потока управления.

Ветвление потока управления

Одна из главных особенностей диаграммы последовательности - возможность визуализировать простое ветвление процесса. Для изображения ветвления используются две или более стрелки, выходящие из одной точки фокуса управления объекта (объект ob1 на рис. 3.44). При этом рядом с каждой из них должно быть явно указано соответствующее условие ветви в форме булевского выражения [21].

Количество ветвей может быть произвольным, однако наличие ветвлений может существенно усложнить интерпретацию диаграммы последовательности. Предложение-условие должно быть явно указано для каждой ветви и записывается в форме обычного текста, псевдокода или выражения языка программирования. Это выражение всегда должно возвращать некоторое булевское выражение. Запись этих условий должна

исключать одновременную передачу альтернативных сообщений по двум и более ветвям. В противном случае на диаграмме последовательности может возникнуть конфликт ветвления [21].

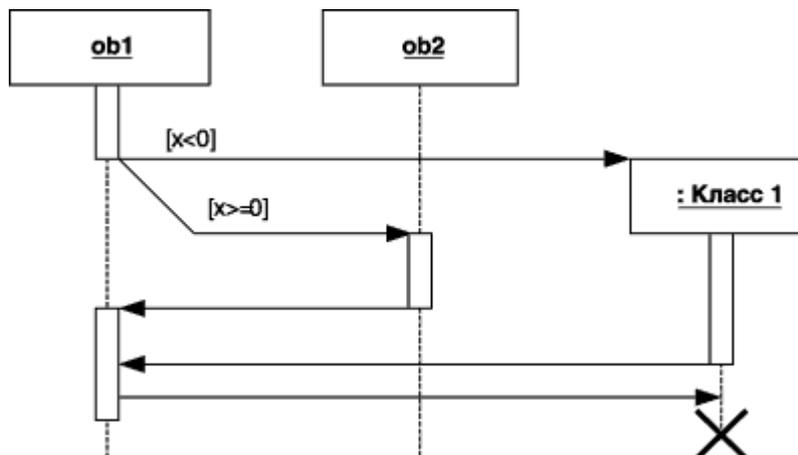


Рис. 3.44. Графическое изображение бинарного ветвления потока управления на диаграмме последовательности

С помощью ветвления можно изобразить и более сложную логику взаимодействия объектов между собой (объект ob1 на рис. 3.45). Если условий более двух, то для каждого из них необходимо предусмотреть ситуацию единственного выполнения. Описанный ниже пример относится к моделированию взаимодействия программной системы обслуживания клиентов в банке. В этом примере диаграммы последовательности объект ob1 вызывает выполнение действий у одного из трех других объектов [21].

Условием ветвления может служить сумма снимаемых клиентом средств со своего текущего счета. Если эта сумма превышает 1500\$, то могут потребоваться дополнительные действия, связанные с созданием и последующим разрушением объекта Класса 1. Если же сумма превышает 100\$, но не превышает 1500\$, то вызывается операция или процедура объекта ob3. И, наконец, если сумма не превышает 100\$, то вызывается операция или процедура объекта ob2. При этом объекты ob1, ob2 и ob3 постоянно существуют в системе. Последний объект создается от Класса 1 только в том случае, если справедливо первое из альтернативных условий. В противном случае он может быть никогда не создан [21].

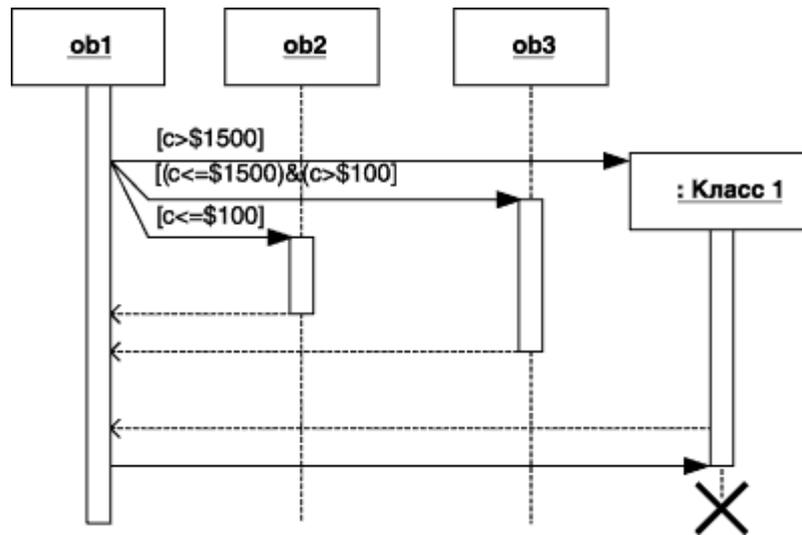


Рис. 3.45. Графическое изображение тернарного ветвления потока управления на диаграмме последовательности

Объект ob1 имеет постоянный фокус управления, а все остальные объекты - получают фокус управления только для выполнения ими соответствующих операций.

Рекомендации по построению диаграмм последовательности

Построение диаграммы последовательности целесообразно начинать с выделения из всей совокупности классов только тех, объекты которых участвуют в моделируемом взаимодействии. После этого все объекты наносятся на диаграмму, с соблюдением порядка инициализации сообщений. Здесь необходимо установить, какие объекты будут существовать постоянно, а какие временно - только на период выполнения ими требуемых действий [21].

Когда объекты визуализированы, можно приступить к спецификации сообщений. При этом необходимо учитывать те операции, которые имеют классы соответствующих объектов в модели системы. При необходимости уточнения этих операций следует использовать их стереотипы. Для уничтожения объектов, которые создаются на время выполнения своих действий, нужно предусмотреть явное сообщение. Наиболее простые случаи ветвления процесса взаимодействия можно изобразить на одной диаграмме с использованием соответствующих графических примитивов. В более сложных случаях для моделирования каждой ветви управления может потребоваться отдельная диаграмма последовательности. Следует помнить, что каждый альтернативный поток управления затрудняет понимание построенной модели [21].

Общим правилом является визуализация особенностей реализации каждого варианта использования на отдельной диаграмме последовательности. В этой ситуации отдельные диаграммы должны рассматриваться совместно как одна модель взаимодействия.

Необходимость синхронизации сложных потоков управления, как правило, требуют введение в модель дополнительных ограничений [21].

Диаграммы последовательности могут быть использованы для моделирования последовательности действий совершенно обыденной жизни. Например, покупка книг у поставщика библиотекой, показанная на рис. 3.46.

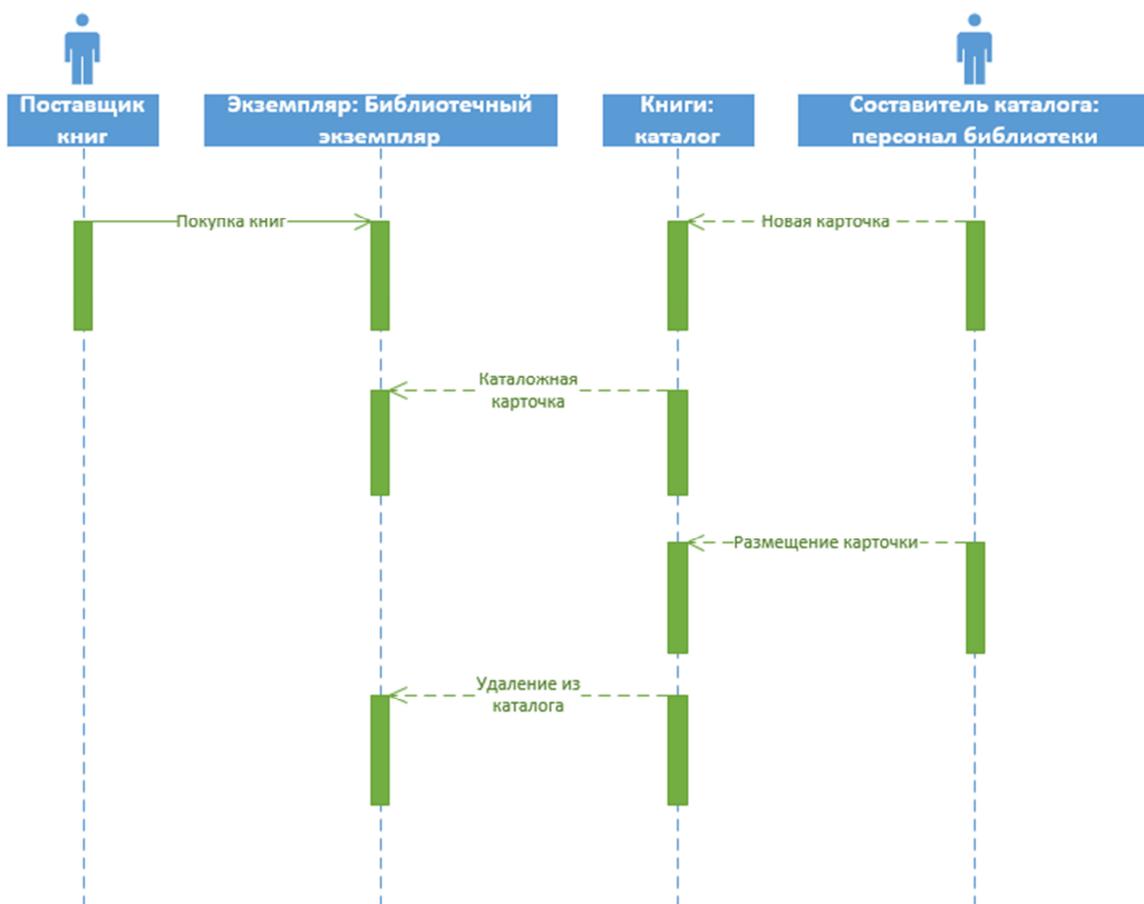


Рис. 3.46. Последовательность действий по покупке книг библиотекой

В узкоспециализированных сферах деятельности диаграммы последовательностей также находят применение. Особо часто, диаграммы, построенные в нотации UML используют для описания свойств программного обеспечения.

На рис. 3.47. приводится пример диаграммы последовательностей, на которой изображен порядок действий при прохождении плановых медосмотров в военкомате. При получении вызова от военкомата призывник собирает справки и сдает анализы согласно памятки в тексте вызова, после чего является в назначенный день на медкомиссию. Регистратор медкомиссии делает отметку о посещении и выдает призывнику обходной лист с пометками сколько и каких врачей нужно обойти.

Призывник приходит к нужным врачам, проходит осмотр. Затем врачи пишут заключение в обходной лист и возвращают его призывнику. После приема у всех врачей

призывник передает обходной лист обратно регистратору. Регистратор фиксирует у себя все пометки из обходного листа, ставит отметку о прохождении медкомиссии и передает данные в администрацию военкомата для дальнейшей работы.

Обратите внимание, что на рис. 3.47 диаграмма последовательности имеет цикл. Так оформляется многократное повторение одного и того же действия. Цикл задается от 1 до X. Если известно число итераций, то вместо X указывается конкретное число. Действие, которое будет повторяться многократно помещается в прямоугольник.

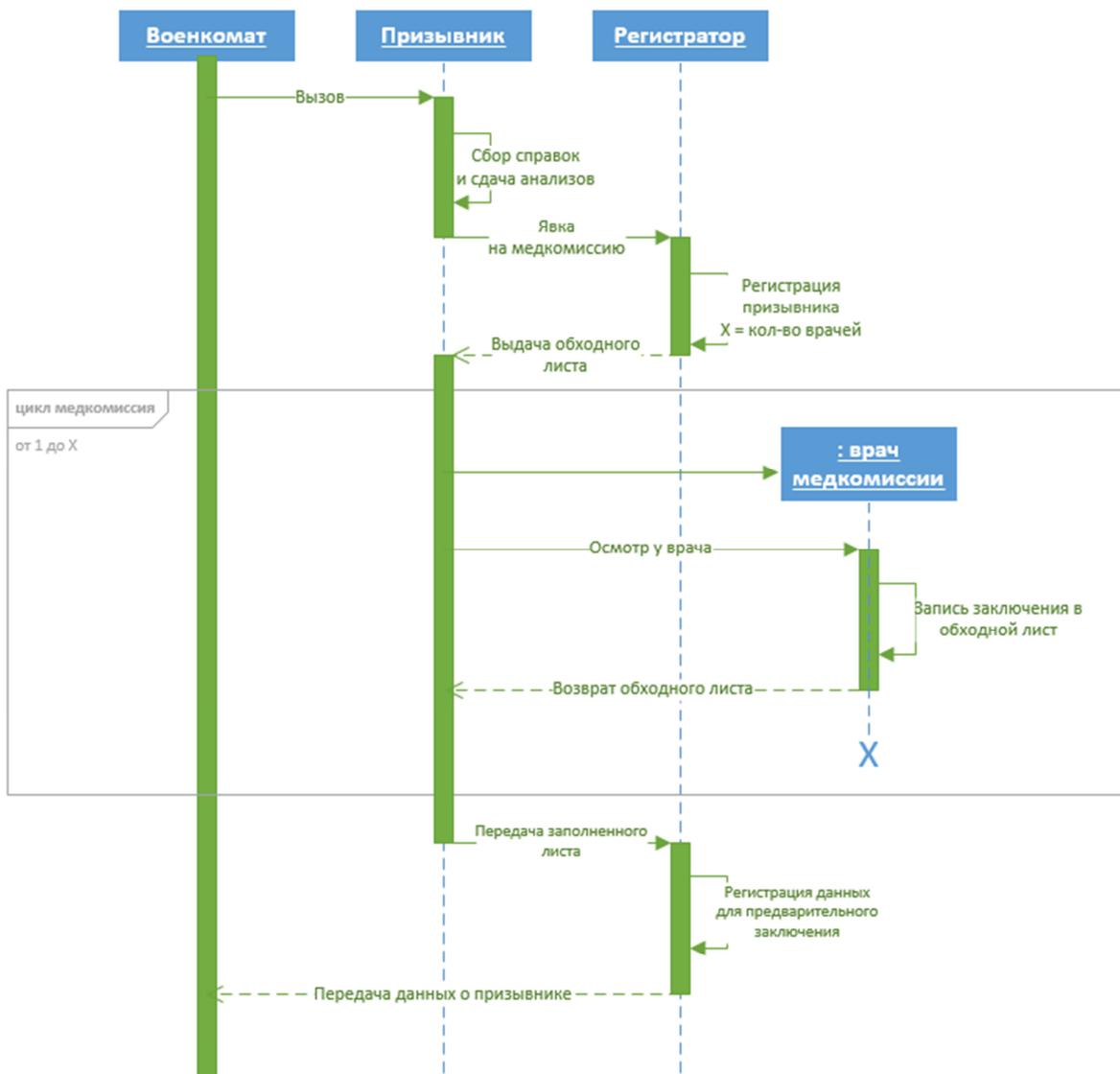


Рис. 3.47. Последовательность действий при прохождении медкомиссии в военкомате

ПРАКТИЧЕСКАЯ ЧАСТЬ

- 1) Запустить программу MS Visio;

- 2) Для того, чтобы загрузить нужную панель инструментов для разработки диаграмм при создании документа в поле «Загрузить шаблоны из Интернета» укажите «UML»
- 3) В найденных пунктах выберите «UML. Деятельность».

Задание 1. Создать диаграмму последовательности «Закупка книг библиотекой»

- 1) Создайте диаграмму последовательностей, изображенную на рис. 3.46.
- 2) Дайте диаграмме 3.46 текстовое описание, по примеру диаграммы на рис. 3.47.

Задание 2. Создать диаграмму последовательности «Прохождении комиссии в военкомате»

- 1) Создайте диаграмму последовательностей, изображенную на рис. 3.47

Задание 3. Создать диаграмму последовательности «Сдача экзаменационной сессии»

- 1) Самостоятельно разработайте и постройте диаграмму последовательности действий по сдаче экзаменационной сессии;
- 2) Дополните созданную диаграмму текстовым описанием последовательности действий.

Задание 4. Разработайте собственную диаграмму последовательности

- 1) На основе примеров диаграмм последовательностей, разработанных в заданиях 1-4, создайте свою диаграмму последовательностей действий из любой области.
- 2) Дополните созданную диаграмму текстовым описанием последовательности действий.

Задание 5. Подготовьте отчет

Подготовьте отчет по лабораторной работе, который должен содержать:

- а) титульный лист
- б) все разработанные диаграммы и текстовые описания к ним.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

- 1) Сколько измерений может иметь диаграмма последовательности?
- 2) Как изображаются объекты на диаграммах последовательности?
- 3) Что называют линией жизни объекта?
- 4) Каков порядок передачи сообщений на диаграмме последовательностей?
- 5) Как изобразить на диаграмме последовательностей прекращение деятельности какого-либо объекта?

- 6) Какой объект на диаграмме последовательностей называют сиротой, а какой анонимом?
- 7) С какой целью строят диаграммы последовательностей?
- 8) Что называют фокусом управления?
- 9) Что называют рефлексивным сообщением?
- 10) В каком случае на диаграмме последовательностей может возникнуть конфликт ветвления?

ЛАБОРАТОРНАЯ РАБОТА № 14

Разработка смешанных UML-диаграмм

Цель работы: получение навыка разработки смешанных UML-диаграмм, закрепление навыков разработки UML-диаграмм разного типа.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Построение модели информационной системы на ранних этапах ее разработки является предпосылкой успешного построения проектируемой системы.

Описание проектных решений на языке UML представляет собой совокупность диаграмм. На верхних уровнях могут строиться ролевые диаграммы, разработанные на базе диаграмм прецедентов и отображающие взаимодействие объектов системы, а также выполняемые ими функции (рис. 3.48).

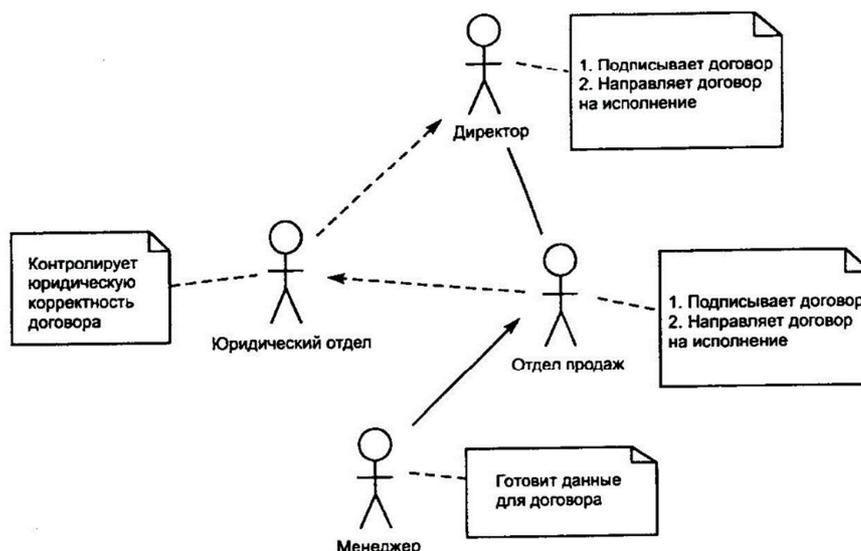


Рис. 3.48. Вид ролевой диаграммы

При необходимости отображения взаимосвязи процессов могут строиться модели их взаимодействия (рис. 3.49).

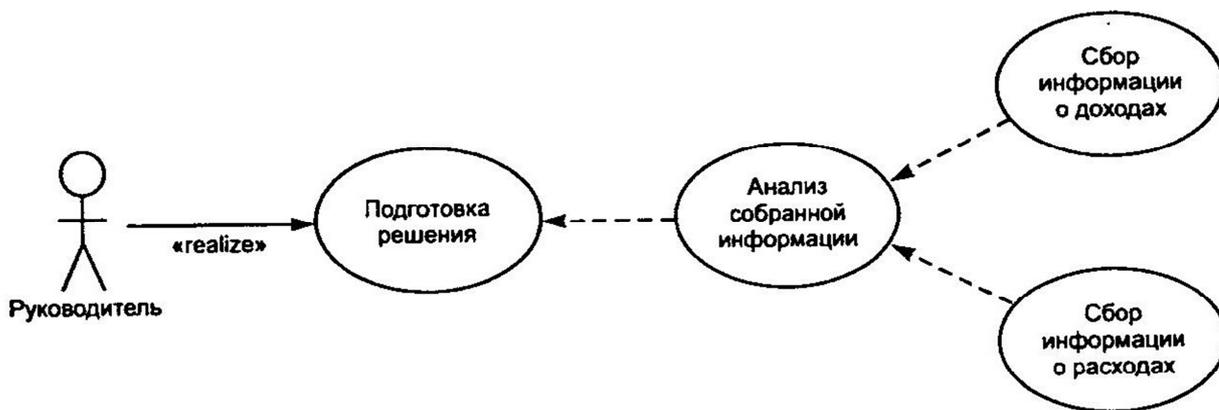


Рис. 3.49. Вид модели взаимодействия процессов

Развитие нотации поведенческих диаграмм до современного расстояния позволяет успешно использовать смешанные модели, добавляя компоненты из других видов диаграмм и благодаря этому повышая наглядность. Так, например, к диаграммам активности могут быть добавлены элементы диаграмм прецедентов для обозначения ролевых объектов. При этом смешанные диаграммы с использованием языка UML строятся не в произвольной взаимосвязи, а в соответствии с определенными правилами взаимосвязи (рис. 3.50).



Рис. 3.50. Схемы взаимосвязи UML диаграмм

Определяющими (начальными) обычно являются диаграммы прецедентов, а декомпозиция может выполняться с использованием других видов диаграмм.

Диаграммы UML обеспечивают формирование ясного представления проектируемых структур и позволяют вырабатывать правильные решения в процессе анализа бизнес-процессов и создания информационных систем. Разнообразие диаграмм,

входящих в состав языка UML, позволяет рассмотреть бизнес-процессы в разных аспектах и провести моделирование с учетом всех требуемых факторов.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1.

На основе примеров, выполненных в лабораторных работах №№ 10-14 разработайте совокупность UML-диаграмм, детально описывающих работу какого-либо приложения или информационной системы (ИС). Приложение или ИС для разработки и описания выбираются самостоятельно. Это может быть как реально существующее программное обеспечение, так и собственная идея студента.

Работа должна содержать следующие элементы:

1. Диаграмму вариантов использования. Данная диаграмма должна содержать список функций, которое будет выполнять ваше приложение, а также актёров, которые будут взаимодействовать с системой.
2. Для двух любых вариантов использования разработайте главные и исключающие сценарии использования. Опишите какой актер является главным, а какие второстепенными.
3. Диаграмму классов, которая должна содержать объекты системы и отношения между ними. При описании однородных классов указать их атрибуты и операции. Опишите какой класс является базовым, а какие листовыми.
4. Диаграммы деятельности, содержащие детализацию действий по реализации двух любых функций из тех, что представлены на диаграмме вариантов использования.
5. Текстовое описание диаграмм деятельности.
6. Диаграмму последовательностей, содержащую последовательность действий по взаимодействию с пользователем и другими ресурсами при работе приложения или ИС.
7. Текстовое описание работы и функционала вашего приложения, его целевую аудиторию.

Задание 2.

Разработайте смешанную UML-диаграмму, описывающую любую предметную область по вашему выбору.

Задание 3.

Составьте отчет по лабораторной работе, который должен содержать

- а) титульный лист;
- б) обоснование выбора данной темы, актуальность;
- в) все диаграммы и описания, указанные в задании 2
- г) смешанную UML-диаграмму.

ПРИМЕЧАНИЕ: Все выбранные темы согласовываются с преподавателем и названия их фиксируются.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

1. Что представляют собой смешанные UML-диаграммы?
2. Что такое ролевые диаграммы?
3. Какие существуют правила для смешивания UML-диаграмм различных типов?

ГЛАВА IV. ОБЩИЕ ПРИНЦИПЫ МЕТОДОЛОГИИ И СИСТЕМЫ ARIS

4.1. Программное обеспечение ARIS

Анализ деятельности предприятий и реорганизация бизнес-процессов – чрезвычайно сложная задача, требующая методической и инструментальной поддержки. Как уже известно, существует много программных продуктов, используемых бизнес-аналитиками для этих целей. Однако зачастую крупные предприятия отдают предпочтение программному продукту ARIS Toolset (ARIS). ARIS – архитектура интегрированных информационных систем, методология и тиражируемый программный продукт для моделирования бизнес-процессов организаций [3]. ARIS представляет собой векторный редактор, предназначенный для моделирования бизнес-процессов, в т.ч. для создания функциональных и поведенческих моделей информационных систем.

Разработчиком методологии ARIS и одноименной инструментальной среды является германская фирма IDS Scheer AG (Германия), в частности профессор А.В. Шеер. На сегодняшний день принято считать линейку инструментов ARIS лидирующей на мировом рынке в классе средств моделирования и анализа бизнес-процессов. Методология ARIS является достаточно рафинированной. Здесь все распределено, разграничено и специфицировано, процесс работы формализован в набор алгоритмов. Организация в ARIS рассматривается с четырех точек зрения:

- ✓ организационной структуры данных;
- ✓ функциональной структуры данных;
- ✓ структуры данных;
- ✓ структуры процессов.

При этом каждая из этих точек зрения разделяется еще на три подуровня:

- ✓ описания требований;
- ✓ описание спецификации;
- ✓ описания внедрения.

Классическое полнопакетное ПО ARIS Toolset имеет серьезное преимущество перед прочими продуктами. Это преимущество заключается в хорошо развитых графических средствах представления сформированных моделей. ARIS позволяет вывести в отчетный документ любую информацию, содержащуюся в базе данных проекта, а также получить аналитическую отчетность, структурированную по ряду признаков. ARIS является подходящим инструментом для детальной классификации, структурирования и наглядного представления операционных рисков. Оптимизация деятельности в ARIS

сводится к выделению, формализации и структурированию бизнес-процессов с целью формирования на их основе «сквозного» представления процессов организации. Перед внедрением продукта ARIS непременно должна быть проведена серьезная «ручная» проектно-аналитическая работа. Главным преимуществом методологии ARIS являются эргономичность и высокая степень визуализации бизнес-моделей, что делает данную методологию удобной и доступной в использовании всеми работниками организации (начиная от топменеджеров и заканчивая рядовыми работниками). Методология содержит более 80 моделей, и поэтому для осмысленного и полноценного использования требуются значительные временные ресурсы на изучение.

Одним из минусов пакета ARIS можно отметить его высокую стоимость и сложность освоения, которая произрастает из высокой функциональности продукта. Данный момент критичен для небольших предприятий и внедрение подобных систем не оправдано по многим причинам. Однако в 2009 году корпорация IDS Scheer выпустила ARIS Express – бесплатную упрощенную версию программы для моделирования бизнес-процессов [22].

Бесплатная версия программы поддерживает только базовые типы диаграмм, не имеет многопользовательской поддержки, не использует базу данных, не содержит инструментов для формирования отчетов и средств анализа модели. И самое главное: **ARIS Express** не поддерживает связи между создаваемыми объектами в отличие от полноценной платной версии, то есть отсутствует контроль целостности и непротиворечивости модели. Это означает, что при редактировании одной модели программа не будет вносить соответствующие изменения в другую модель, а также не будет проверять существуют ли должности, указываемые в качестве ответственных в процессе и т.д. [22]. Тем не менее это не отменяет ее пользы для освоения в учебных целях.

4.2. Методология ARIS

ARIS получила свое название от английского выражения «Architecture of Integrated Information Systems» (архитектура интегрированных информационных систем). Под архитектурой подразумевается совокупность технологий, обеспечивающих проектирование, управление, применение и реализацию бизнеса в виде «деловых» процедур бизнес-процессов предприятий и организаций, а также проектирование и создание интегрированных информационных систем поддержки бизнес-процессов.

ARIS представляет собой и методологию и программный продукт для моделирования бизнес-процессов организаций одновременно. В дальнейшем при упоминании о системе ARIS (либо инструментальной среде ARIS) будет подразумеваться

аппаратное и программное обеспечение, реализующие методологию ARIS. Когда упоминается методология ARIS –подход к структурированному описанию деятельности организации.

Методология ARIS представляет собой современный подход к структурированному описанию деятельности организации и представлению ее в виде взаимосвязанных и взаимодополняющих графических диаграмм, удобных для понимания и анализа. Данная методология объединяет в себе множество различных методик в рамках единого системного подхода.

Поскольку методология ARIS реализует принципы системного структурного анализа, то основным понятием ее служит структурный элемент (объект). Также как и структурный анализ методология ARIS использует декомпозицию и позволяет детализировать предмет моделирования с помощью альтернативных или дополняющих друг друга моделей.

На основе методологии ARIS любая организация рассматривается и визуально представляется как единая система, описание которой предусматривает пять различных «взглядов» [3]:

- ✓ организационная;
- ✓ функциональная;
- ✓ обрабатываемые данные;
- ✓ структуры бизнес-процессов;
- ✓ продукты и услуги.

Все эти пять подсистем организации должны быть связаны между собой в реальности и моделях. При помощи методологии ARIS становится возможным описание достаточно разнородных подсистем, которые представляются в виде взаимоувязанной и взаимосогласованной совокупности различных моделей. Эта совокупность моделей хранится в едином репозитории (рис. 4.1). Отличительными особенностями методологии ARIS от всех прочих существующих методологий является именно взаимосвязанность и взаимосогласованность моделей.

Впоследствии каждая из подсистем разбивается на элементарные блоки, в соответствии с правилами структурного анализа. Совокупность этих элементарных блоков или модулей и составляет нотацию структурной модели той или иной подсистемы организации.



Рис. 4.1. Структурная схема формирования репозитория [23]

В соответствии с правилами структурного анализа каждая из этих подсистем разбивается на элементарные блоки (модули), совокупность которых и составляет нотацию структурной модели той или иной подсистемы организации.

Естественно, все подсистемы не могут быть полностью обособленными. Они взаимно проникают друг в друга, и поэтому одни и те же элементарные модули могут использоваться для описания различных структурных моделей. Для устранения избыточности методология ARIS выделила пять типов представлений моделей, отражающих основные аспекты организации (см. рис. 4.2):

1. Организационные модели, описывающие иерархическую структуру системы, т.е. иерархию организационных подразделений, должностей, полномочий конкретных лиц, многообразие связей между ними, а также территориальную привязку структурных подразделений;

2. Функциональные модели, описывающие функции (процессы, операции), выполняемые в организации;

3. Информационные модели (т.е. модели данных), отражающие структуру информации, необходимой для реализации всей совокупности функций системы;

4. Модели процессов или управления, представляющие комплексный взгляд на реализацию деловых процессов в рамках системы и объединяющие вместе другие модели;

5. Модели входов и выходов, описывающие потоки материальных и нематериальных входов и выходов, включая потоки денежных средств [23].

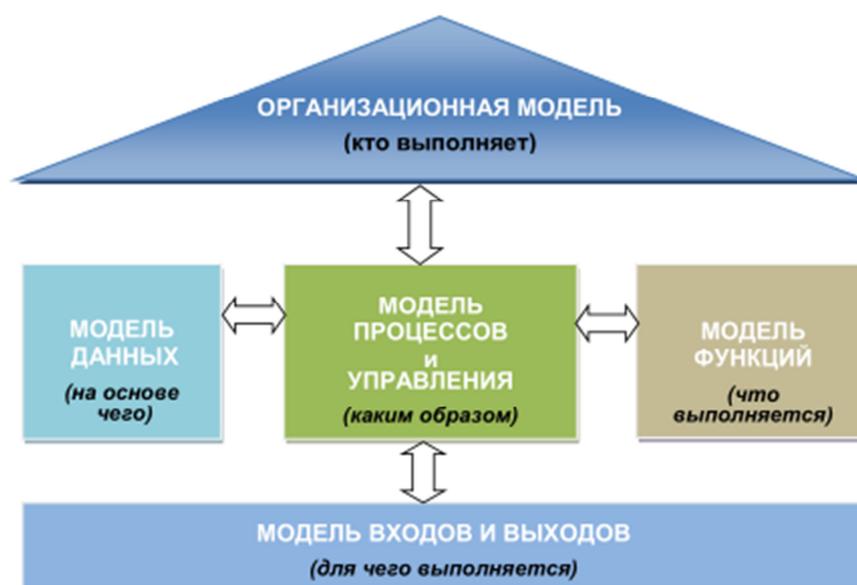


Рис. 4.2. Взаимосвязи моделей деятельности организации [23]

Типы представления являются первой компонентой архитектуры. Они позволяют структурировать бизнес-процессы и выделять их составные части, что делает рассмотрение более простым. Применение этого принципа позволяет с различных точек зрения описывать содержание отдельных частей бизнес-процесса, используя специальные методы, наиболее полно соответствующие каждой точке зрения. Это избавляет пользователя от необходимости учитывать множество связей и соединений. Другой особенностью методологии ARIS, обеспечивающей целостность разрабатываемой системы, является использование различных уровней описания, что поддерживает теорию жизненного цикла системы, существующего в сфере информационных технологий.

Для каждого «взгляда» придерживаются три уровня анализа (требования, спецификации, внедрения), что обеспечивает целостность разрабатываемой системы.

Каждый уровень соответствует определенной фазе жизненного цикла информационной системы:

1. уровень определения требований (что система должна делать);
2. уровень проектной спецификации (основные пути реализации системы);
3. уровень описания реализации (физическое описание конкретных программных и технических средств).

Каждый из уровней анализа состоит из своего комплекта моделей различных типов, в том числе диаграмм UML, диаграмм SAP R/3 и др. Каждый объект моделей ARIS имеет множество атрибутов, позволяющих контролировать процесс разработки моделей, определить условия для выполнения функционально-стоимостного анализа, имитационного моделирования, взаимодействия с work flow-системами и т.д.

4.3. Модели в ARIS

Под моделью понимается совокупность объектов, объединенных друг с другом различными связями, и ряда вспомогательных элементов. Каждую модель характеризуют типом, именем и свойствами. Тип модели определяет характер информации, описываемой моделью — организационную структуру, функции, данные, процессы или выходы. Имя модели является частью ее атрибутов. Модель ARIS является частью развернутой модели организации. В то же время модель может представлять отдельный объект, будучи его детальным описанием.

При построении моделей методология ARIS требуется соблюдать ряд требований. К ним относятся:

- ✓ корректность модели;
- ✓ релевантность (следует моделировать только те фрагменты реальной системы которые соответствуют назначению системы, т.е. модель не должна содержать
 - ✓ избыточной информации);
 - ✓ соизмеримость затрат и выгод;
 - ✓ прозрачность, т.е. понятность и удобство использования модели;
 - ✓ сравнимость моделей;
 - ✓ иерархичность;
 - ✓ систематизация структуры, что предполагает в качестве обязательного условия возможность интеграции моделей различных типов.

Модели классифицируются с помощью методологических фильтров. При этом каждая модель ARIS содержит:

- ✓ объекты – неделимые части модели, выделенные по какому-либо признаку, сформулированному в соответствии с методологией ARIS, и имеющие набор изменяемых характеристик-свойств, описывающих их поведение;
- ✓ связи между объектами – описанные взаимоотношения между объектами, имеющими свои свойства и характеристики, а также характеризуются внешним видом и атрибутами.

Модель может включать:

- ✓ внешние встроенные объекты, например, рисунки, документы текстовых редакторов и т.п.;
- ✓ текст, размещенный в любом месте модели;
- ✓ геометрические фигуры.

ЛАБОРАТОРНАЯ РАБОТА № 15

Знакомство с интерфейсом программы ARIS EXPRESS.

Моделирование организационной структуры предприятия

Цель работы: знакомство со средой программы ARIS Express, получение навыков разработки модели организационной структуры предприятия.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

4.4. Особенности интерфейса ARIS Express

При запуске программы ARIS Express появляется стартовая страница, показанная на рис. 4.3. На этой странице пользователь может выбрать диаграмму, необходимую для построения. При выборе соответствующего типа диаграмм, открывается другое окно с необходимым набором элементов.

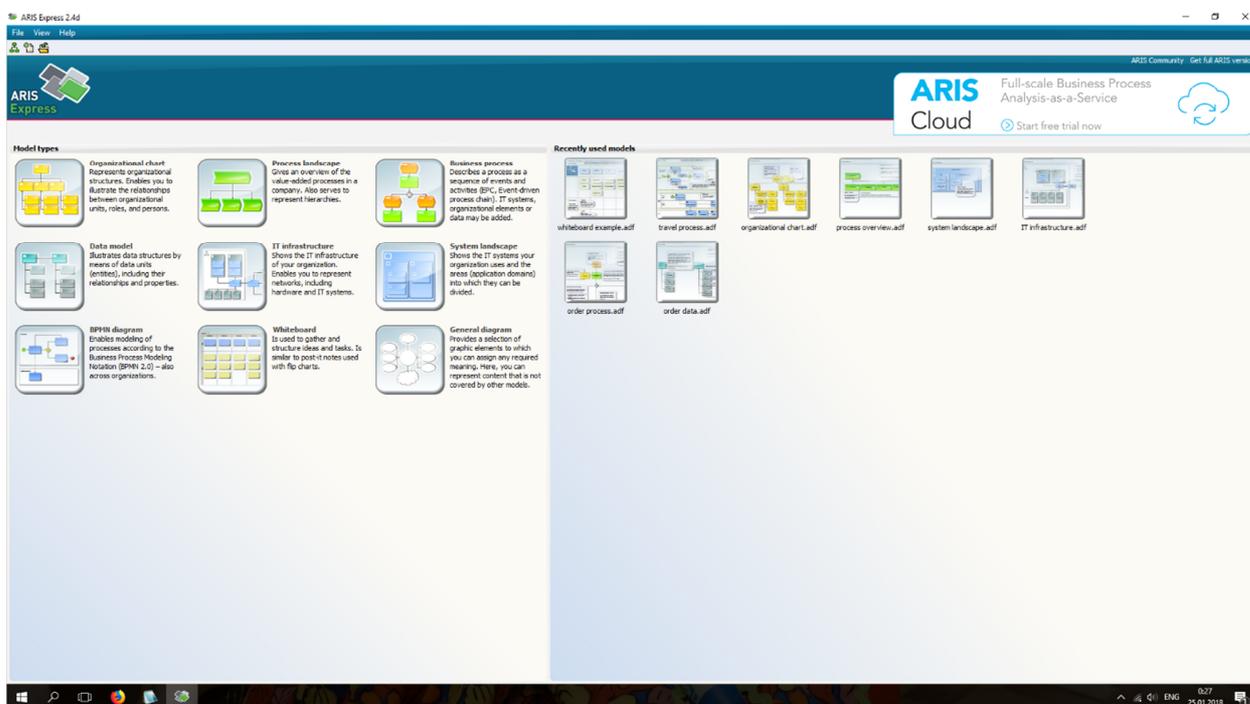


Рис. 4.3. Стартовая страница программы ARIS Express

Для создания новой диаграммы следует выполнить стандартные действия. Выбрать пункт меню **File / New**. При выборе, например, диаграммы «**Organization Chart**», появляется окно рабочей области по центру и панель символов справа (см. рис. 4.4). Для добавления необходимых элементов в рабочую область нужно просто перетащить их мышью.

Для редактирования элементов необходимо вызвать контекстное меню и нажать кнопку **Properties**. В появившемся диалоговом окне **Object properties** нужно перейти на вкладку **Attributes**. На вкладке **Attributes** можно задать следующие параметры:

- ✓ Имя, отображаемое на диаграмме (*Name*);
- ✓ Тип содержимого (*Type*);
- ✓ Комментарии и описания (*Description / Definition*);
- ✓ Автор;
- ✓ Прочее.

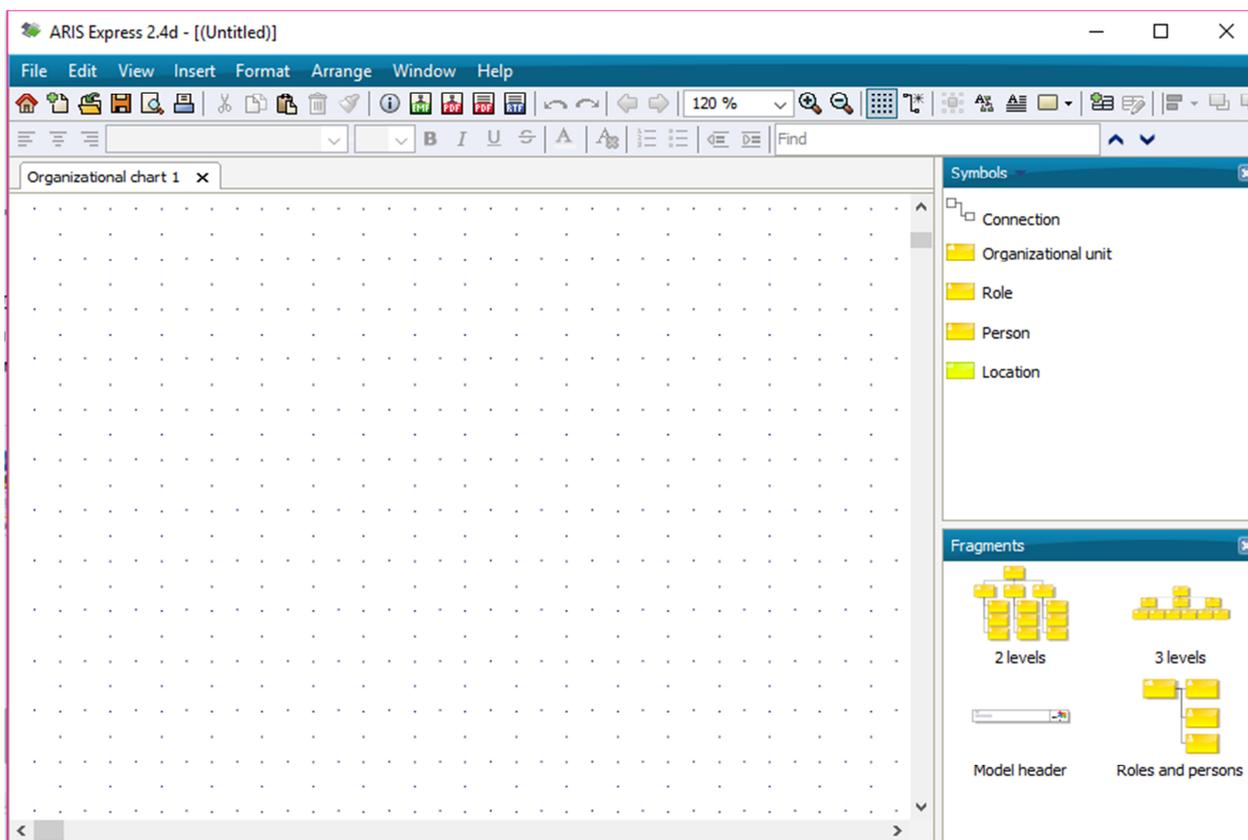


Рис. 4.4. Окно рабочей области при создании организационной диаграммы

Помимо атрибутов любому объекту можно изменить внешние характеристики (*Object appearance*):

- ✓ Цвет фона (Fill color);
- ✓ Цвет границы символа или линии (Line color);
- ✓ Стиль границы символа или линии (Line style);
- ✓ Толщина границы символа или линии (Weight)
- ✓ Высота (Height);
- ✓ Ширина (Width);
- ✓ Наличии тени и объема (3-D effect).

Несмотря на то, что программа ARIS Express является бесплатной и урезана в своей функциональности, созданные в ней модели можно переносить в базы данных, управляемые корпоративными инструментами семейства продуктов ARIS. Продукты семейства ARIS поддерживают импорт созданных в ARIS Express файлов формата **adf**.

Таким образом, решение по описанию бизнес-процессов в ARIS Express достаточно масштабируемо.

Помимо того, что все диаграммы можно создать в ARIS обычным методом, т.е. вручную, в программе имеется функция *Smart Design* (*Умный дизайн*). Эта функция позволяет пользователю создавать специальные таблицы (*sheet-view*) и на их основе получать необходимую информацию о компании. Основным преимуществом таблиц *sheet-view* является тот факт, что пользователи могут не заботиться о стандартных правилах моделирования и правилах размещения объектов. Их применение позволяет концентрироваться только на содержательной части.

4.5. Разработка модели организационной структуры предприятия

Организационная диаграмма (Organization chart) описывает организационные единицы различного уровня и описывает связь между ними. Данная модель является одной из важнейших, поскольку она описывает субъекты, определяющие входы и выходы потоков ресурсов предприятия, участвующие в построении и управлении бизнес-процессами. Как правило, с удобнее начинать проекты по моделированию бизнес-процессов с разработки организационной схемы предприятия.

Организационные модели относятся к категории иерархических и строятся от верхнего уровня структуры к нижнему. Как правило содержит в себе следующие данные:

- ✓ подразделения предприятия;
- ✓ должности и ФИО руководителей подразделений;
- ✓ физическое местоположения отделов на предприятии.

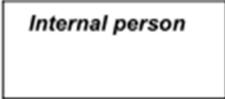
Модель верхнего уровня содержит самостоятельные подразделения (департаменты, блоки), входящие в структуру организации. Затем каждое из них детализируется на более низком уровне структурных подразделений (управления, отделы, сектора). Заканчивается описание нижним уровнем описания подразделений на уровне должностей (штатные единицы).

Данный тип диаграмм также удобно использовать при построении иерархии подчинённости в проектных группах, командах. Если описание касается сложных иерархических структур, то допустимо применение процесса декомпозиции, когда структура какого-либо подразделения может быть отражена на более детальной схеме.

Количество уровней в модели организационной структуры не регламентируется жестко и определяется потребностями организации. Между объектами организационной диаграммы устанавливаются взаимосвязи. В таблице 4.1. приводится сопоставление

классического представления элементов организационной структуры и элементов методологии ARIS.

Таблица 4.1. Графические элементы организационной диаграммы

Классическое представление	Представление в методологии ARIS	Наименование, описание элемента
		<p>Организационные единицы: различные организационно-экономические единицы предприятия, их можно классифицировать по функциональным или региональным аспектам.</p> <p>Цвет фигуры: желтый.</p>
		<p>Должность: элементарная организационная единица компании. Под должностью следует понимание штатной единицы, занимаемой конкретным сотрудником. В модели указывается наименование должности.</p> <p>Цвет фигуры: желтый</p>
		<p>Штатный сотрудник: Ф.И.О. штатного сотрудника или другой его идентификатор.</p> <p>Цвет фигуры: желтый.</p>
		<p>Расположение: территориальное местонахождение организационной единицы.</p> <p>Цвет фигуры: лайм</p>

Наиболее часто употребляемые связи между элементами модели показаны в таблице 4.2.

Таблица 4.2. Типы связей между элементами модели описания структуры подразделений

Элементы	Типы связей между элементами			
	Organization unit	Position	Internal position	Location
Organization unit	Состоит из (is composed of)	Состоит из (is composed of)	----	Располагается (is located is)
Position	Руководитель подразделения (is organization manager of)	----	----	----
Internal position	----	----	----	----
Location	----	----	Занимает должность (occupies)	----

В тех ячейках таблицы 4.2., где стоит прочерк, связи между указанными элементами модели не предусматриваются, либо востребованы крайне редко в процессе моделирования.

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Запустите программу ARIS Express;
2. Выберите пункт меню File / New / Organization Chart;

Задание 1.

Создайте организационную диаграмму «Времена года»

Задание 2.

1. Создайте организационную диаграмму компании «Стиль для всех» на основе приведенного описания.

Генеральный директор имеет в своем подчинении заместителей директора по финансам, по обеспечению, по развитию, по кадрам, по технике. Каждый заместитель директора имеет в подчинении соответствующий департамент.

В департамент по финансам входят: отдел планирования (в нем работают менеджер по планированию и специалист по финансам), бухгалтерия (в ней работают главный бухгалтер и бухгалтеры).

В департамент по обеспечению входят: отдел закупок, отдел рекламации, склад, отдел сбыта, отдел доставки. Отдел закупок состоит из начальника отдела, аккаунт-менеджера и менеджера по закупкам.

Отдел рекламации состоит из консультанта.

Штат склада включает начальника отдела и кладовщика.

Отдел сбыта состоит из менеджера по продажам и специалиста по продажам.

Отдел доставки состоит из курьера.

В департамент по развитию входят: отдел интернет-маркетинга (в нем работают редактор, копирайтер, рекламный аналитик), отдел продвижения (в нем работают специалист по CPA (сопровождение клиентов), специалист по SMM (маркетинг в социальных сетях), отдел контента (в нем работает фотограф).

В департамент по технике входят: отдел IT-разработки, отдел документации.

В департамент по кадрам входят: отдел кадров, юридический отдел.

Общие сведения: «Стиль для всех» – молодая, динамично развивающаяся на территории России компания, специализирующаяся на розничной продаже молодежной одежды. Компания на рынке уже 4 года, имеет более 350 000 ед. проданных товаров через интернет-магазин. Помимо Интернет-ресурса компания имеет склад, в котором хранятся товары. Склад и офисы с рабочими местами сотрудников, которые расположены в нескольких городах страны. Значительная часть ассортимента находится на складе, что позволяет оперативно реагировать на потребности клиентов.

Компания осуществляет отправку одежды стиля casual в любой регион России, начиная от крупных городов и заканчивая мелкими населенными пунктами.

В интернет-магазине функционируют и хорошо отлажены такие бизнес-процессы как поддержка пользователей и подготовка контента интернет-магазина. Функция поддержки пользователей необходима для полноценного контакта с потребителем, проведения кампаний по работе с лояльностью клиентов и формирования необходимого для коррекции деятельности компании пакета отзывов. Процесс базируется на функционале интернет-представительств организации и обслуживании клиентов через call-центр. В качестве основных показателей эффективности выступают оперативность реакции сотрудников и качество ответов на интересующие вопросы. Все рабочие места сотрудников оборудованы ПК и имеют инструкции по работе с клиентом.

Все обращения в контакт-центр разделяются по каналу обращения:

- ✓ входящий звонок клиента на горячую линию;
- ✓ входящий звонок клиента в Интернет-магазин;
- ✓ письмо клиента по электронной почте;
- ✓ сообщение клиента в чате онлайн-консультирования.

На данный момент создан каталог товаров для интернет-магазина, а также сформирован шаблон для описания, размещаемого на интернет-сайте товара. Фотостудии

была заказана фотосъемка товара. Сформированные описания товаров и фотографии загружаются в используемую организацией информационную систему.

Сформированные объекты товаров загружаются на сайт и становятся доступны клиентам. Размещённая на сайте информация проверяется ответственными сотрудниками.

Закупка товаров обычно производится у постоянных поставщиков (как правило, производителей) крупными партиями в соответствии с планом закупки. Компания активно развивается, постоянно ищет новых поставщиков для увеличения ассортимента товаров, в том числе за рубежом. На основании этого было принято решение по увеличению складского помещения, автоматизации процесса поиска и заключения договора с новым поставщиком.

Миссия компании: стать лидером в области продаж одежды, достойным высоким доверия покупателей, оказывать лучший сервис не только по региону, но и по всей стране, стать одним из законодателей модных трендов в сегменте продаж стильной одежды.

2. Сформируйте отчет, содержащий следующие элементы:

- а) Титульный лист;
- б) Описание предметной области, миссии компании.
- в) Организационную диаграмму.

Задание 3.

1. Изучите сайт вашего университета (www.tusur.ru) и составьте текстовое описание его организационной структуры;
2. Средствами программы ARIS Express разработайте модель организационной структуры университета;
3. По индивидуальному заданию преподавателя декомпозируйте одно из структурных подразделений с полным описанием его штата (наименованием должностей и ФИО сотрудников).

ПРИМЕЧАНИЕ: Задание выдается преподавателем каждому студенту индивидуально!

4. Сохраните созданную модель в файл.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

1. Что представляет собой методология ARIS?

2. Что представляет собой программа ARIS Express?
3. Что означает аббревиатура ARIS?
4. С каких точек зрения рассматривается организация в методологии ARIS?
5. Какие 5 типов моделей выделяются в методологии ARIS?
6. Что понимается под моделью ARIS?
7. Какие требования к построению моделей предусматривает методология ARIS?
8. Что такое организационная диаграмма и каково ее назначение?

ЛАБОРАТОРНАЯ РАБОТА № 16

Моделирование цепочки добавленного качества

Цель работы: получение навыков разработки цепочки добавленного качества для предприятия в программе ARIS Express

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Бизнес-процесс – это целенаправленная совокупность взаимосвязанных видов деятельности, преобразующая данные на входе в данные на выходе (товары и услуги), представляющих ценность для потребителя.

Принято выделять три вида бизнес-процессов:

1) **Управляющие** – бизнес-процессы, управляющие функционированием системы. Примером управляющего процесса может служить: корпоративное управление, стратегический менеджмент.

2) **Операционные** – бизнес-процессы, которые составляют основной бизнес компании и создают основной поток доходов. Такие как снабжение, производство, маркетинг, продажи.

3) **Поддерживающие** – бизнес-процессы, которые обслуживают основной бизнес. Например, бухгалтерский учет, подбор персонала, техническая поддержка.

Вместе с тем можно выделить иную классификацию бизнес-процессов:

✓ **сквозные** (межфункциональные) – процессы, проходящие через несколько подразделений организации или через всю организацию.

✓ **внутрифункциональные** (процессы подразделений) – процессы в рамках одного функционального подразделения организации.

✓ **функции** (операции) – процессы самого нижнего уровня декомпозиции деятельности организации, как правило, операции, выполняемые одним человеком.

Каждый бизнес-процесс имеет свои границы (вход и выход), конечного потребителя и своего владельца.

Владелец процесса – это должностное лицо или коллегиальный орган управления, имеющий в своем распоряжении ресурсы, необходимые для выполнения процесса, и несущий ответственность за результат процесса.

Выход (продукт) бизнес-процесса – материальный или информационных объект, или услуга, являющийся результатом выполнения процесса и потребляемый по отношению к процессу клиента (например, готовая продукция, документация, информация, персонал, услуги и т.д.).

Вход бизнес-процесса – продукт, который в ходе выполнения процесса преобразуется в выход. Входы процесса поступают в процесс извне (например, сырье, материалы, полуфабрикаты, документация, информация, персонал, услуги и т.д.).

Ресурс бизнес-процесса – материальный или информационный объект, постоянно используемый для выполнения процесса, но не являющийся входом процесса. Ресурсы бизнес-процесса находятся под управлением владельца этого процесса (например, информация, персонал, оборудование, программное обеспечение, инфраструктура, среда, транспорт, связь и т.д.).

Бизнес-процессы верхнего уровня (корневая модель бизнес-процессов) – описывают функционирование предприятия в целом.

В рамках представления процессов (Processes) базовыми являются следующие модели:

- ✓ диаграмма цепочки добавленного качества (*Value-added chain diagram*) (включает три вида процессов: основной, вспомогательный и процесс управления для всей организации);
- ✓ событийная цепочка процесса (*extended Event-driven Process Chain*);
- ✓ диаграмма окружения функции (*Function allocation diagram*).

Для детального описания деятельности предприятия строят иерархию моделей бизнес-процессов предприятия. Модель строится иерархически – от верхнего уровня функции к нижнему (через декомпозицию). Для описания бизнес-процессов верхнего уровня используется диаграмма *Value-added chain diagram*, название которой можно перевести как **Модель цепочки добавленного качества** (стоимости). В самом общем случае цепочка добавленного качества (стоимости) – это полная цепочка операций участников рынка формирующая ценностное предложение конечному потребителю.

Обычно выделяются и рассматриваются цепочки создания ценности внутри самой компании. В таком случае они будут представлять собой последовательность функциональных систем, входящих в жизненный цикл образования продукции и направленные на удовлетворение определенных потребностей потребителя.

Элементами цепочки создания ценности могут быть функциональные системы или подсистемы.

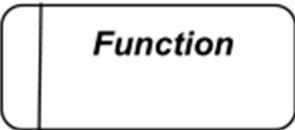
Основную роль в цепочке добавления качества выполняют функции, выходом которых являются продукты (услуги) с измененным качеством и добавленной стоимостью.

Эта модель описывает иерархию деятельности компании и последовательность деятельности на каждом иерархическом уровне. Представление деятельности компании в данной нотации подчиняется следующим правилам (см. табл. 4.3):

- ✓ функции могут размещаться в соответствии с последовательностью этапов создания продукции;
- ✓ добавлением качества и стоимости на каждом последующем этапе работ;
- ✓ между функциями могут устанавливаться иерархические связи или отношения, т.е. можно описывать иерархию (вложенность) функций и этапов сквозного процесса;
- ✓ функции могут быть разделены на подфункции.

Таблица 4.3.

Графические элементы модели описания цепочек добавленного качества

Классическое представление	Представление в нотации ARIS	Наименование, описание
		Группа бизнес-процессов, бизнес-процесс, функция (на этой модели обозначается одинаково): описание элемента работы, образующего один логический этап в рамках бизнес-процесса. Цвет фигуры: зеленый

Между объектами модели устанавливаются взаимосвязи подчинения. В таблице 4.4. представлены некоторые типичные связи объектов диаграммы цепочек добавленного качества. В этих случаях, как правило, используется процессно-ориентированное подчинение (is process-oriented superior), применимое при процессно-ориентированной деятельности функции (последовательность функций, составляющих процесс).

Таблица 4.4.

Типичные связи объектов диаграммы цепочек добавленного качества

Название связи на английском языке	Название связи на русском языке
------------------------------------	---------------------------------

Accepts	Утверждает результат
Contributes to	Способствует при выполнении
Executes	Выполняет
Is consumed by	Потребляется
Decides on	Принимает решение по
Has consulting role in	Участвует в качестве консультанта
Has output of	Имеет на выходе
Is input for	Является входом для
Is IT responsible for	Отвечает по IT за
Is process-oriented superior	Подчиняется подпроцессу
Is predecessor of	Предшествует
Is superior	Имеет в подчинении
Is technical responsible for	Отвечает за техническую часть
Must be informed on cancellation	Должен быть информирован о нестандартном завершении
Must be informed about	Должен быть информирован о выполнении
Must inform about result of	Должен информировать о результатах выполнения
Produces	Производит
Supports	Поддерживает

ПРАКТИЧЕСКАЯ ЧАСТЬ

Запустите программу ARIS Express и выберите тип диаграммы Process Landscape.

Задание 1. Ознакомиться с примером диаграммы цепочки добавленного качества компании «Стиль для всех». На диаграмме приведен бизнес-процесс верхнего уровня (рис. 4.5).

Задание 2. Построить в программе ARIS Express диаграмму, показанную на рис. 4.5.

Задание 3. Задать типы связей (Relationship type) всех связей диаграммы в соответствии с таблицей 4.4.

Задание 4. Ознакомиться с примером описания операционного бизнес-процесса «Интернет-маркетинг» (рис. 4.6).

Задание 5. Построить в программе ARIS Express диаграмму, показанную на рис. 4.6.

Задание 6. Самостоятельно построить диаграммы остальных операционных бизнес-процессов.

Задание 7. Построить собственный пример диаграммы цепочки добавленного качества для бизнес-процессов верхнего уровня и две диаграммы для операционного уровня.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

1. Что такое бизнес-процесс?
2. Какие виды бизнес-процессов существуют?
3. Какие виды бизнес-процессов позволяет описывать программа ARIS Express?
4. Какие выделяют укрупненные группы бизнес-процессов?
5. Какие параметры присущи каждому бизнес-процессу?
6. Какие модели в ARIS Express представляют бизнес-процессы верхнего уровня?
7. Что представляет собой модель цепочки добавленного качества?
8. Что играет основную роль в цепочке добавленного качества?
9. Каким правилам подчиняется представление деятельности компании в нотации Value-added chain diagramm?
10. Какая связь устанавливается между объектами модели в нотации Value-added chain diagramm?

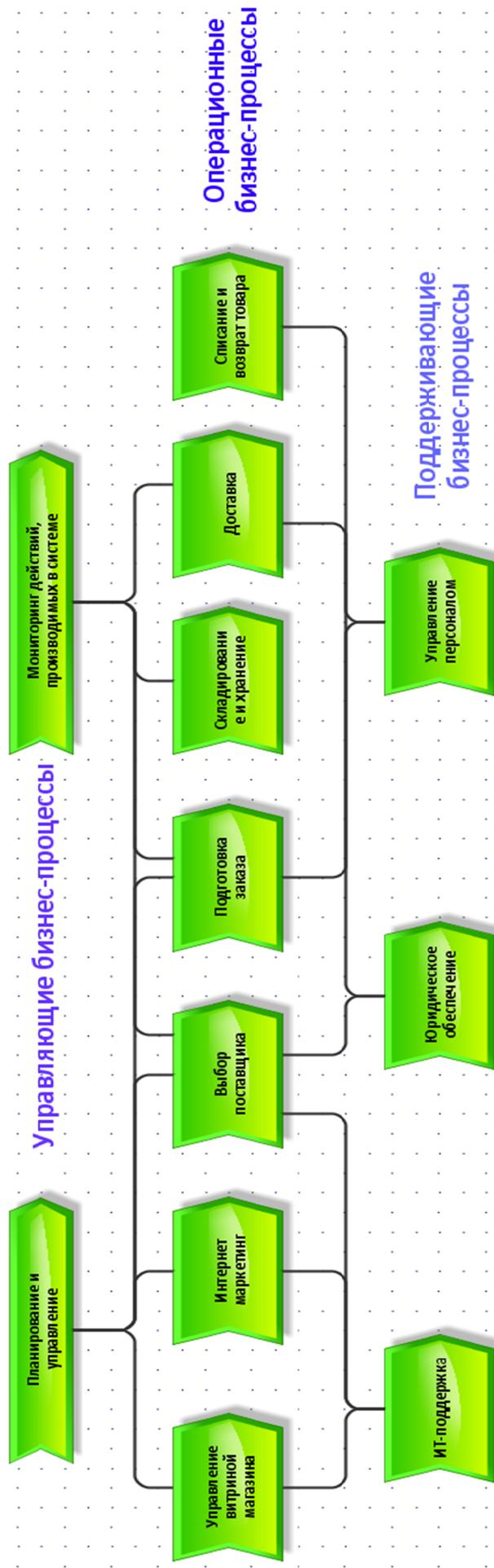


Рис. 4.5. Пример диаграммы основных бизнес-процессов верхнего уровня компании «Стиль для всех»

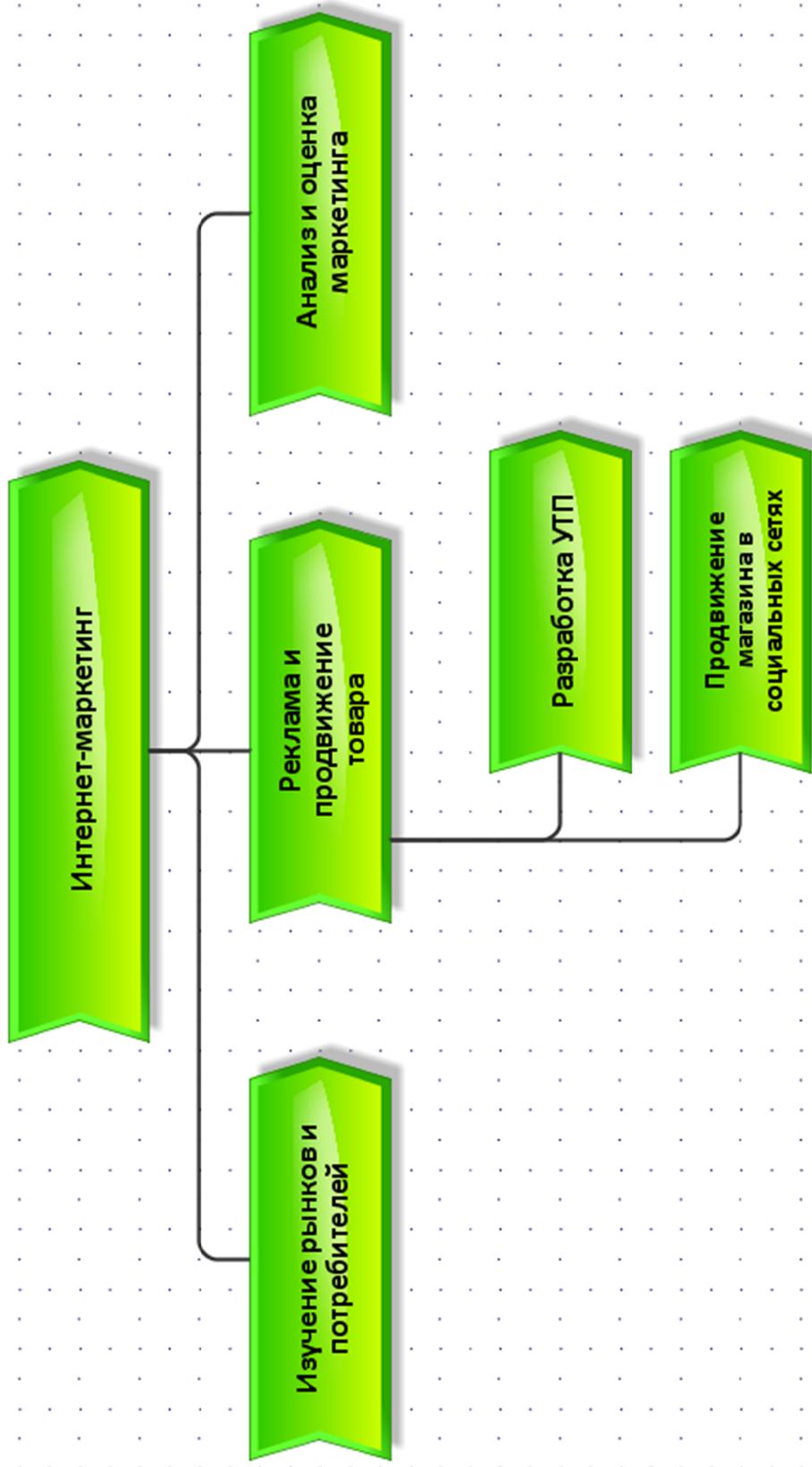


Рис. 4.6. Пример диаграммы бизнес-процессов работы отдела «Интернет-маркетинг»

ЛАБОРАТОРНАЯ РАБОТА № 17

Построение модели типа EPC

Цель работы: получение навыков разработки процессно-событийной модели в программе ARIS Express

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Процессно-событийная модель EPC предназначена для описания процессов, выполняемых в рамках одного подразделения, несколькими подразделениями или конкретными сотрудниками.

Процессно-событийная модель:

- ✓ позволяет выявлять взаимосвязи между организационной и функциональной моделями;
- ✓ отражает последовательность функциональных шагов (действий) в рамках одного бизнес-процесса, которые выполняются организационными единицами, а также ограничения по времени, налагаемые на отдельные функции.

В процессно-событийной модели есть два ключевых понятия «Событие» и «Функция». Ключевым моментом при разработке моделей типа EPC является четкое отличие этих двух понятий. **Событие** – это факт свершения чего-либо, причем не имеющий продолжительности во времени, либо это время ничтожно мало. Событие всегда вызывает необходимость исполнения функции и исполнение функции всегда заканчивается событием. Рассмотрим отличие этих понятий на примере телефонного звонка. Звонит телефон. Менеджер взял трубку для телефонного разговора. В данном случае «Звонит телефон» (факт)- это событие. «Телефонный разговор» (процесс) – это функция. Разговор завершен – снова событие. Таким образом, получается некая событийная цепочка, которая и определяет смысл и название нотации [24].

Для того, чтобы схема была более наглядной, нотация предусматривает еще несколько стандартных элементов (см. таблицу 4.5):

- ✓ **Должность** (исполнитель). Тот, кто выполняет данную функцию
- ✓ **Информация**. Любая информация, используемая для выполнения функции, кроме документальной. Например, телефонный звонок, инструкция по выполнению операции т.п.
- ✓ **Документ**. Элемент «Документ» предназначен для отображения носителей информации (бумажной или электронной). Т.е. представление информации в определенной структуре.

✓ **Программа (приложение).** Программное обеспечение, используемое для выполнения функции.

Все остальные элементы являются вспомогательными, и практически не регламентированы требованиями самой eEPC. Одним из преимуществ данной нотации является возможность добавлять свои собственные элементы. Этот факт необходимо зафиксировать во внутреннем стандарте для обеспечения единства понимания всеми того, как они выглядят и зачем применяются. Такое расширение не нарушает требований, если не нарушается связка событие-функция-событие, и предназначено лишь для улучшения восприятия информации или адаптации правил описания в какой-либо отраслевой специфике [24].

Существуют некоторые особенности расположения элементов. Все эти элементы так или иначе должны быть связаны с функцией. **Это общее правило: с событием не связывается ни один элемент, кроме функции!!!** Т.е. все эти элементы должны быть связаны стрелочками с функцией. Что касается стрелок и их направлений: принято считать, что если нет направления передачи информации, то вместо стрелки отображается просто линия. Если информация входит (поступает на вход), то направление стрелки от объекта к функции, если выходит, то наоборот.

Жестких требований по расположению элементов нет нотация не предъявляет, но для удобства восприятия принято их отображать на всех схемах одинаково (для однообразия и стройности схемы). Для унификации внешнего вида графических схем бизнес-процессов такие правила надо закрепить во внутреннем стандарте и следовать им.

Процессные модели представляют собой перечень основных и вспомогательных бизнес-процессов предприятия (см. табл. 4.5) с их подробным описанием (цели, участники, взаимосвязи и т.д.).

Одно событие может инициировать выполнение одновременно нескольких функций, и, наоборот, функция может быть результатом наступления нескольких событий и функций. Для такого ветвления процесса используются логические операторы («и», «или», «исключающие или») (см. табл. 4.6), а в модели EPC их называют точками принятия решений.

Процессно-событийная модель, содержащая расширенное представление описания бизнес-процессов (например, наиболее полная информация о конкретных входных и выходных документах, или более подробные названия выполняемых функций и т.д.) относится к типу eEPC.

Таблица 4.5. Элементы нотации eEPC и их описания

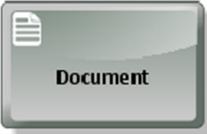
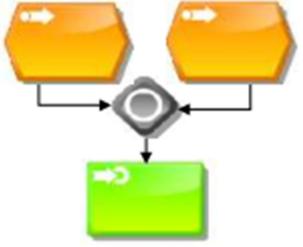
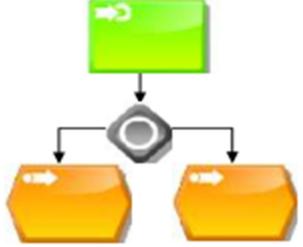
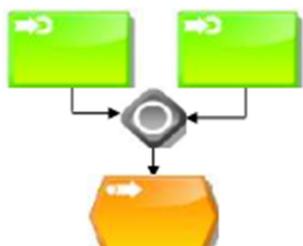
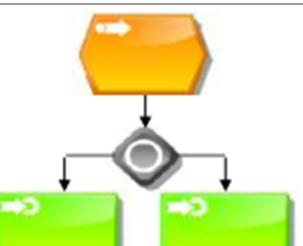
Название элемента	Графическое представление	Описание
Документ		Объект, отражающий реальные носители информации, например, флэш-накопитель, бумажный документ
IT-система		Объект, отражающий реальную прикладную систему
Функция		Объект «Функция» служит для описания функций (работ), которые выполняют подразделения или сотрудники предприятия
Логическое «ИЛИ»		Логический оператор, используемый, когда могут произойти одно или несколько событий, но как минимум одно должно произойти обязательно;
Логическое исключающее «ИЛИ»		Логический оператор, показывающий, что должно произойти одно и только одно событие. Одно событие исключает другое.
Логическое «И»		Логический оператор, используемый, когда произойдут два или более события одновременно
Событие		Служит для описания реальных состояний системы, влияющих и управляющих выполнением функций
Стрелка связи между объектами		Описывает тип отношений между другими объектами

Таблица 4.6. Типы ветвлений и соединения процесса на модели типа eEPC

Типы ветвлений	 Логическое «И»	 Исключающее «ИЛИ»	 Логическое «ИЛИ»
	Функция выполняется, если наступили все события	Функция начинает выполняться тогда, когда наступает только одно из событий	Функция начинает выполняться, если хотя бы одно из событий наступает
	После выполнения функции наступают все события	После выполнения функции наступает ровно одно из событий	После выполнения функций наступает хотя бы одно из событий
	Событие наступает, когда выполнены обе функции	Событие наступает после выполнения ровно одной функции	Событие наступает после выполнения хотя бы одной функции
	При наступлении событий обе функции выполняются	Не разрешено, поскольку событие не может принимать решения (только функции принимают решения)	

Рассмотрим, пример EPC модели для обработки входящего звонка. Пример взят с сайта Хабрахабр [24].

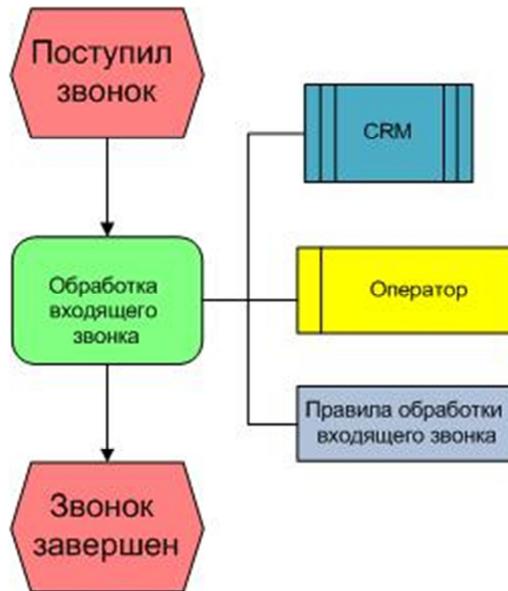


Рис. 4.7. Процессно-событийная модель «Обработка входящего звонка»

Оператор выполняет обработку входящего звонка (рис. 4.7), действуя в соответствии с правилами обработки входящих звонков и использует для этого программу «CRM». Ни входящих, ни исходящих документов при этом не используется

На рис. 4.8. показана более расширенная EPC-модель, включающая элементы логики. Пусть в данном примере будет так: в случае заинтересованности клиента дальнейшую работу с ним проводит менеджер по продажам и выставляет коммерческое предложение, которое отправляет по почте с использованием почтового клиента «MS Outlook». Если заинтересованности нет, то обработка звонка завершена [24].

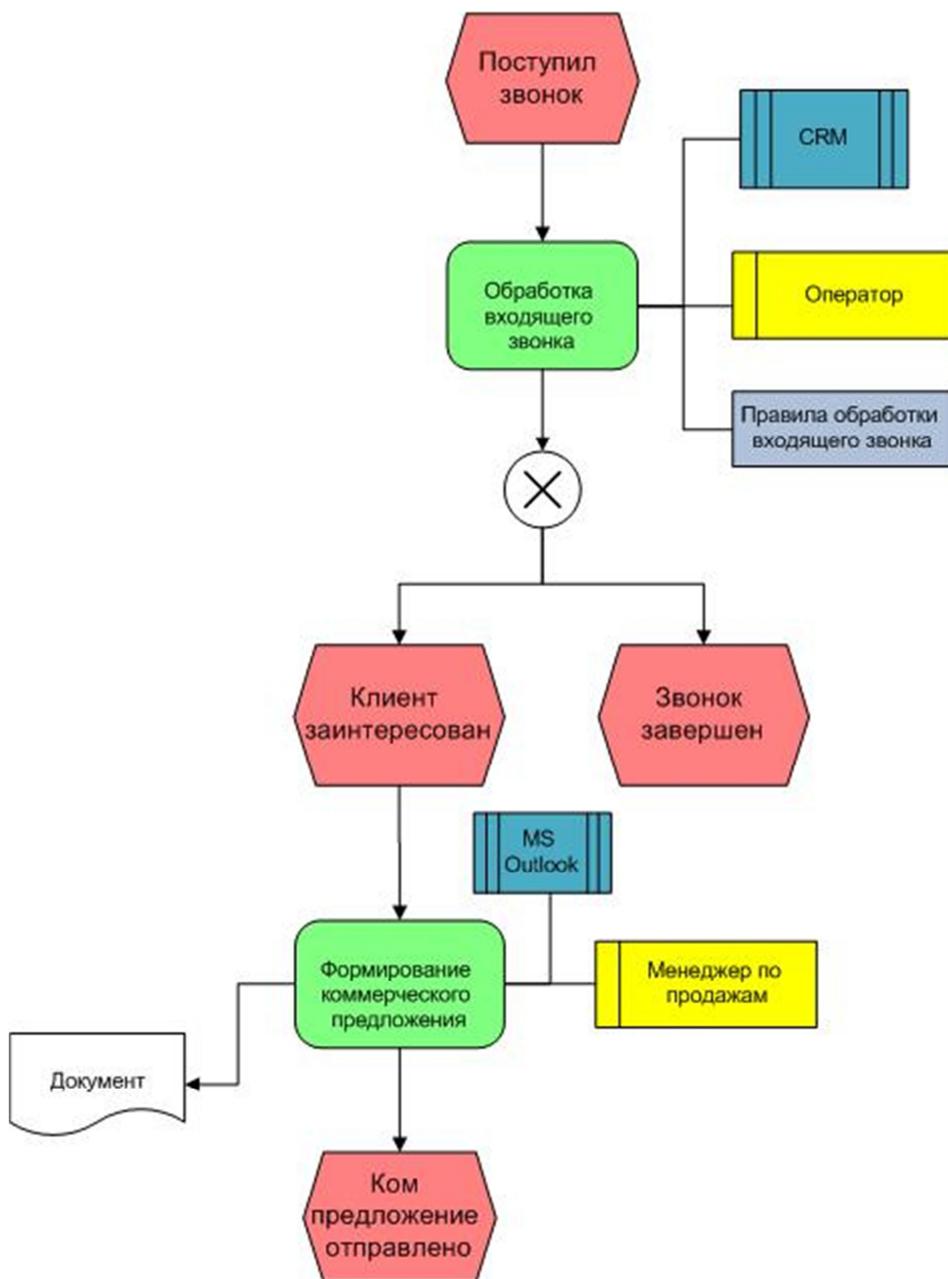


Рис. 4.8. Пример ветвления в процессно-событийной модели

Расширение нотации собственными элементами

eEPC является не совсем нотацией, а именно правилами описания. И эти правила не запрещают добавлять собственные элементы на схему. Главное, чтобы эти элементы были понятными, и существовал документ, где такие расширения элементов зафиксированы. Например, в примере на сайте [24] автор использует дополнительные элементы, которые возникали постепенно в процессе описания реальных процессов для различных задач, от простого описания для постановки задач для автоматизации. Этими элементами являются: база данных, файл, кластер информации, материал, картотека.



Файл с данными. Используется, если в результате выполнения операции создается файл данных, или файл используется для выполнения операции.

База данных. Используется при описании информационных потоков между автоматизированными системами.

Картотека. Используется для отображения бумажной картотеки или архива

Материальный поток. Используется для обозначения входящих и исходящих материальных потоков, а также ресурсов, потребляемых при выполнении процесса. Материальный поток отображается слева от сопровождающих его документов.

Кластер информации. Используется для обозначения структурированной информации (представление сущности). На диаграмме может применяться для обозначения документов, сформированных программным образом при использовании пользовательских приложений. В этом случае элемент «Кластер» располагается слева от соответствующего документа. Т.е. говорит о том, что пользователь не просто создал бумажный документ, но и создал его экземпляр в программе.

Диаграммы в нотации eEPC негласно принято рисовать сверху вниз или слева направо, хотя сама по себе нотация не предъявляет жестких требований в расположении элементов. Однако во избежание недоразумений при работе нескольких специалистов рекомендуется регламентировать и утверждать правила расположения элементов.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Запустить программу ARIS Express. Выбрать тип моделей Business Process.

Задание 1. Разработайте EPC-модель для процесса «Поиск поставщика» (рис. 4.9) по указанным ограничениям:

Компания «Стиль для всех» поставила цель расширить ассортимент товаров в интернет-магазине для расширения целевой аудитории. Во связи с этим перед компаний встали задачи расширить список не только отечественных поставщиков, но и иностранных.

Пример процесса показан на рис. 4.10.

Задание 2. Разработайте EPC-модель для процесса «Складирование и хранение товара на складе» (рис. 4.10).

Задание 3. По заданию преподавателя продумайте и опишите два процесса из списка, представленного ниже. Для каждого процесса составьте EPC-модель.

1. Управление витриной магазина;
2. Интернет-маркетинг;
3. Подготовка заказа;
4. Доставка;
5. Возврат и списание товара;
6. IT-поддержка;
7. Юридическое обеспечение;
8. Управление персоналом.

Задание 4. Разработайте EPC-модели, для процессов из предыдущей лабораторной работы.

Задание 5. По результатам лабораторной работы сформируйте отчет.

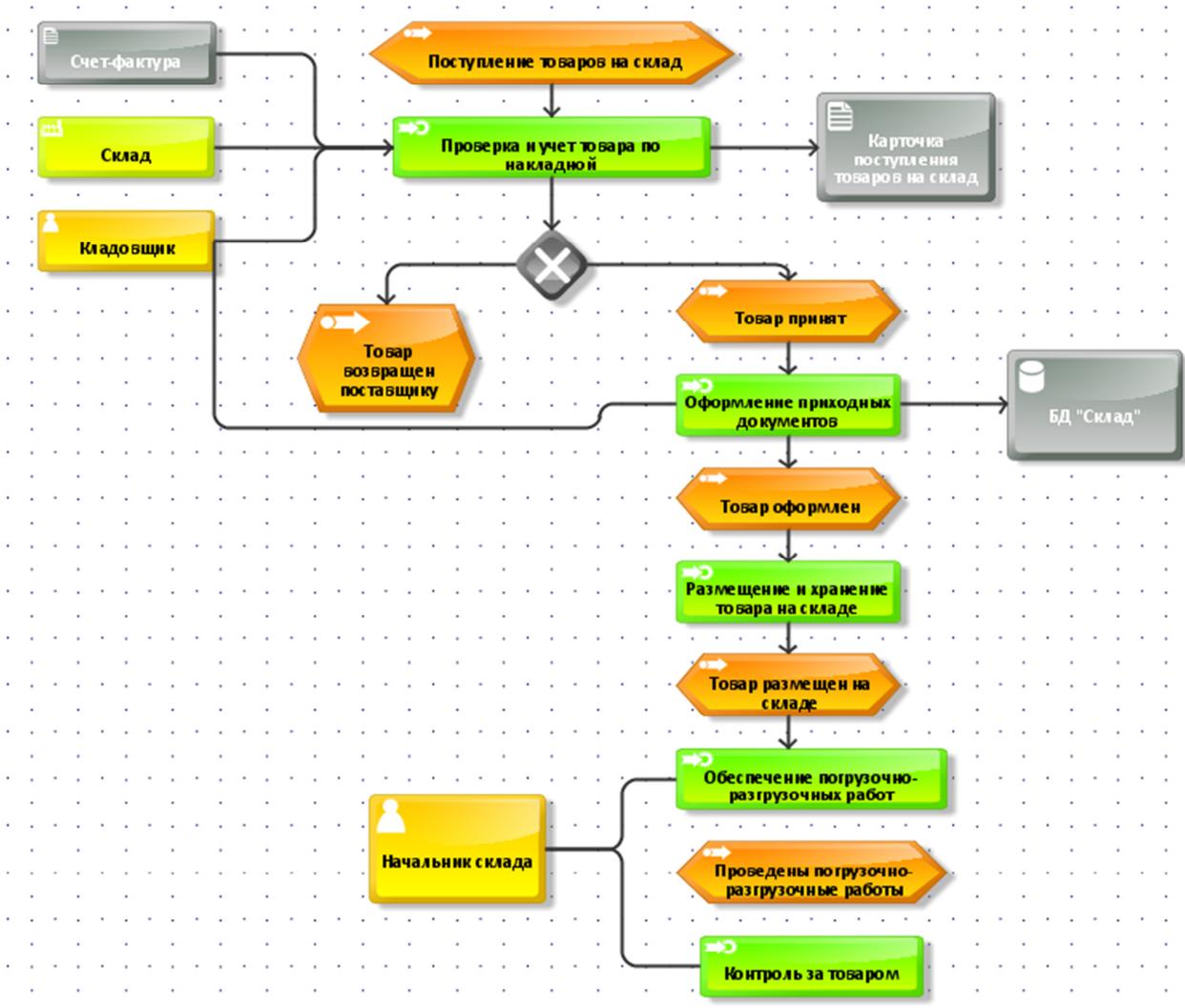


Рис. 4.9. EPC-модель процесса «Поиск пКаоставщика»

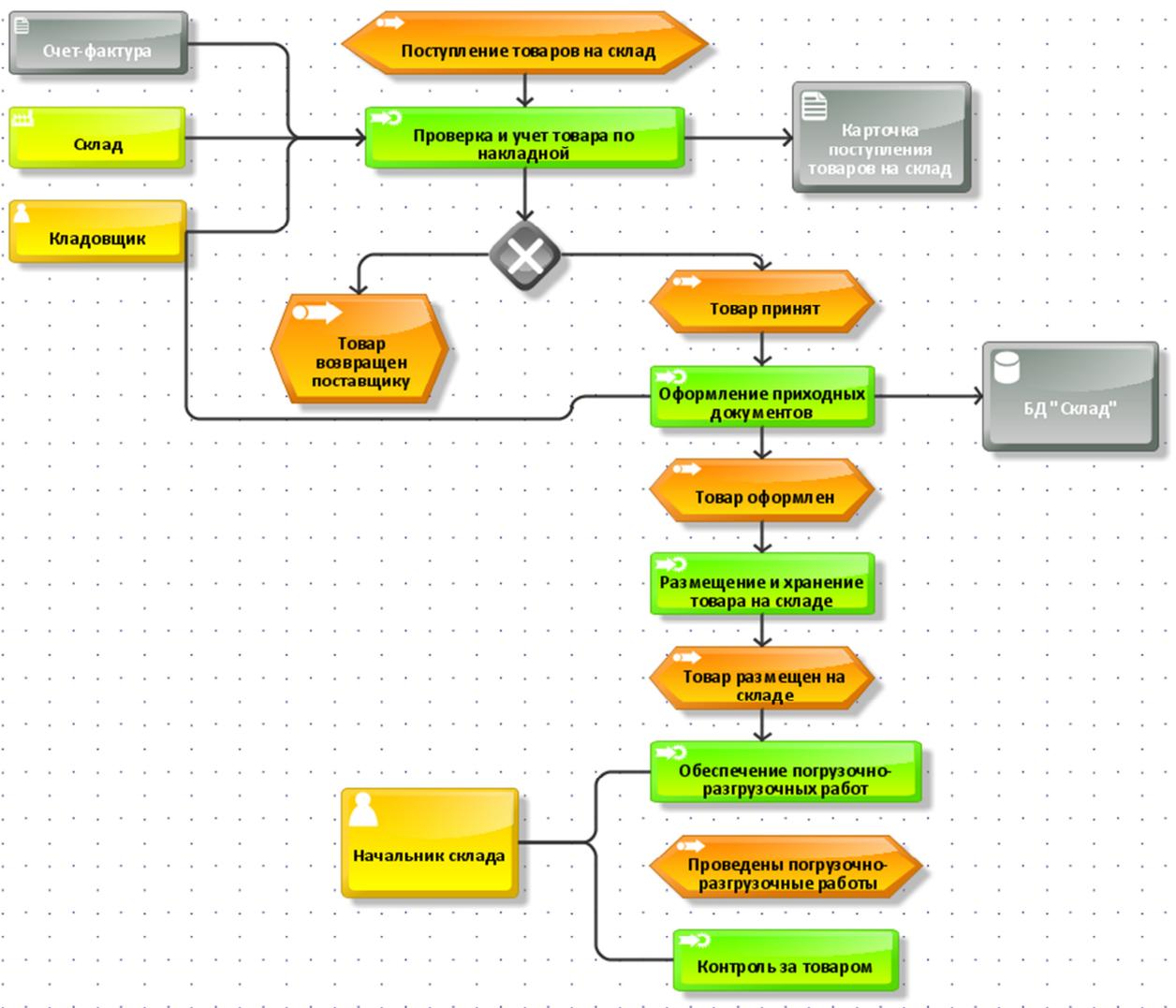


Рис. 4.10. EPC-модель процесса «Складирование и хранение товара»

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

1. Что такое EPC-модель и для чего она предназначена?
2. Какие два ключевых понятия входят в процессно-событийную модель?
3. Какие элементы нотации используются при построении процессно-событийной модели?
4. Какое главное правило связи между элементами EPC-моделей?
5. Можно ли добавлять свои элементы в нотации EPC?
6. Что отображает элемент «Документ» в EPC-моделях?
7. Что отображает элемент «IT-система» в EPC-моделях?
8. Что отображает элемент «Функция» в EPC-моделях?
9. Что отображает элемент «Событие» в EPC-моделях?

10. Как изображаются связи между элементами ЕРС-моделей?
11. Какие логические операторы применяются в ЕРС-моделях?

ЛАБОРАТОРНАЯ РАБОТА № 18

Построение модели интерактивной доски

Цель работы: получение навыка построения диаграммы интерактивной доски в программе ARIS Express

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Для удобного обсуждения идей с другими специалистами компании в программе ARIS Express имеется диаграмма Whiteboard (диаграмма интерактивной доски). В своем роде она имитирует обычную белую доску, на которую маркером записываются идеи. Однако тогда, когда эта условная доска входит в общий комплект диаграмм, это существенно повышает удобство работы в команде. Как известно, такие доски применяются для составления плана какого-либо проекта, записи и структурирования идей, возникающих в процессе переговоров или совещаний. Таким образом, Whiteboard – идеальный инструмент для сбора и анализа творческих совещаний, мозговых штурмов и т.п. [23]

Главным преимуществом модели Whiteboard является возможность внесения дополнений или корректировок ее элементов в любой момент времени без потери общей направленности и структуры проекта.

Рассмотрим структуру модели, предлагаемой интерактивной доски.

Интерактивная доска имеет структуру белого поля, которое разбито на ячейки. Ячейки образуются в результате пересечения столбцов и строк. В поле ARIS Express имеются следующие строки и столбцы для модели интерактивной доски:

- ✓ «**Overall goals**» содержит общие цели проекта, которые необходимо достичь.
- ✓ «**Goals**» - описывают цели, необходимо достичь на каждом этапе;
- ✓ «**Stage**» - описывает этапы проекта;
- ✓ «**Activities**» - перечень мероприятий, необходимых для достижения цели;
- ✓ «**Details**» содержит дополнительные важные детали каждого из этапов для успешной реализации проекта;

В строке «KPI instances» указывают ключевые показатели эффективности. Эти показатели деятельности подразделения или предприятия помогают организации в определении степени достижения стратегических и тактических (операционных) целей. Использование таких ключевых показателей эффективности даёт организации возможность оценить своё состояние и помочь в оценке реализации стратегии [23].

Для разработки модели в пустые ячейки помещаются одноименные элементы из панели инструментов Symbols. На панели Attributes можно задать описание каждого элемента.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1. Постройте модель интерактивной доски для интернет-магазина «Стиль для всех» (см. рис. 4.11) в соответствии с требованиями, указанными ниже.

- 1) Установить основные этапы проекта: 1) анализ; 2) решение; 3) реализация; 4) внедрение.
- 2) Предусмотреть следующие мероприятия: 1) построение функциональной модели; 2) анализ бизнес-процессов; 3) разработка вариантов оптимизации; 4) назначение руководителя проекта; 5) разработка и утверждение расписания; 6) утверждение бюджета; 7) проведение тендера; 8) встреча с поставщиками; 9) разработка всех этапов системы; 10) тестирование; 11) сдача системы 12) разработка документации; 13) корректирующие мероприятия.

3) В качестве целей установить:

- 1 этап – построение модели «ТО-ВЕ»;
- 2 этап – выбор поставщика;
- 3 этап – разработанная система;
- 4 этап – отлаженная эксплуатация.

В качестве деталей на этапе «Решение» установить: отправить запрос поставщикам. На этапе «Внедрение»: предварительное обучение персонала. В качестве количественной оценкой КРІ на этапе «Решение» установить: приглашение 3 участников тендера.

Задание 2. На основе разработанной модели интерактивной доски, постройте пример своей модели.

КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ТЕМЕ

1. Для чего применяется модель интерактивной доски?
2. Что является преимуществом данной модели?
3. Какие поля может содержать модель интерактивной доски?

ЛАБОРАТОРНАЯ РАБОТА № 19

Построение модели данных

Цель работы: получение навыка построения модели данных в программе ARIS Express

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

База данных (БД) является организованной структурой, используемой для хранения данных, например, любых сведений о явлениях, процессах, действиях и т.д. Создание БД, ее поддержка и обеспечения доступа пользователей к ней осуществляется с помощью специализированного программного инструментария – системы управления базами данных (СУБД). Ядром любой базы данных является модель данных. *Модель данных* – это совокупность структур данных и операций их обработки. С помощью модели данных могут быть представлены информационные объекты и взаимосвязи между ними. Выделяют три основных типа моделей данных: иерархическую, сетевую и реляционную.

Модель данных проекта представляет структуру данных на общем уровне, показывает, как данные, используемые в бизнес-процессах, взаимосвязаны [23].

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1. Создайте диаграмму модели данных для интернет-магазина «Стиль для всех» данных (см. рис. 4.12).

На диаграмму разместите объекты типа Entity: «Сотрудник», «Товар», «Покупатель», «Поставщик», «Реализация». В каждом объекте задать по одному первичному ключу (Primary key), при необходимости задать внешние ключи (Foreign key), а также несколько атрибутов (Attribute).

Задание 2. При помощи ресурсов Интернет найдите значение параметров Primary key, Foreign key, Attribute. Выпишите их в отчет по лабораторной работе. Обоснуйте принадлежность указанных категорий данным параметрам.

Задание 3. Самостоятельно изучить типы моделей данных (иерархическую, сетевую и реляционную). Кратко отметить в отчете к лабораторной работе.

Задание 4. Разработать собственную модель данных для регистрации поликлиники.

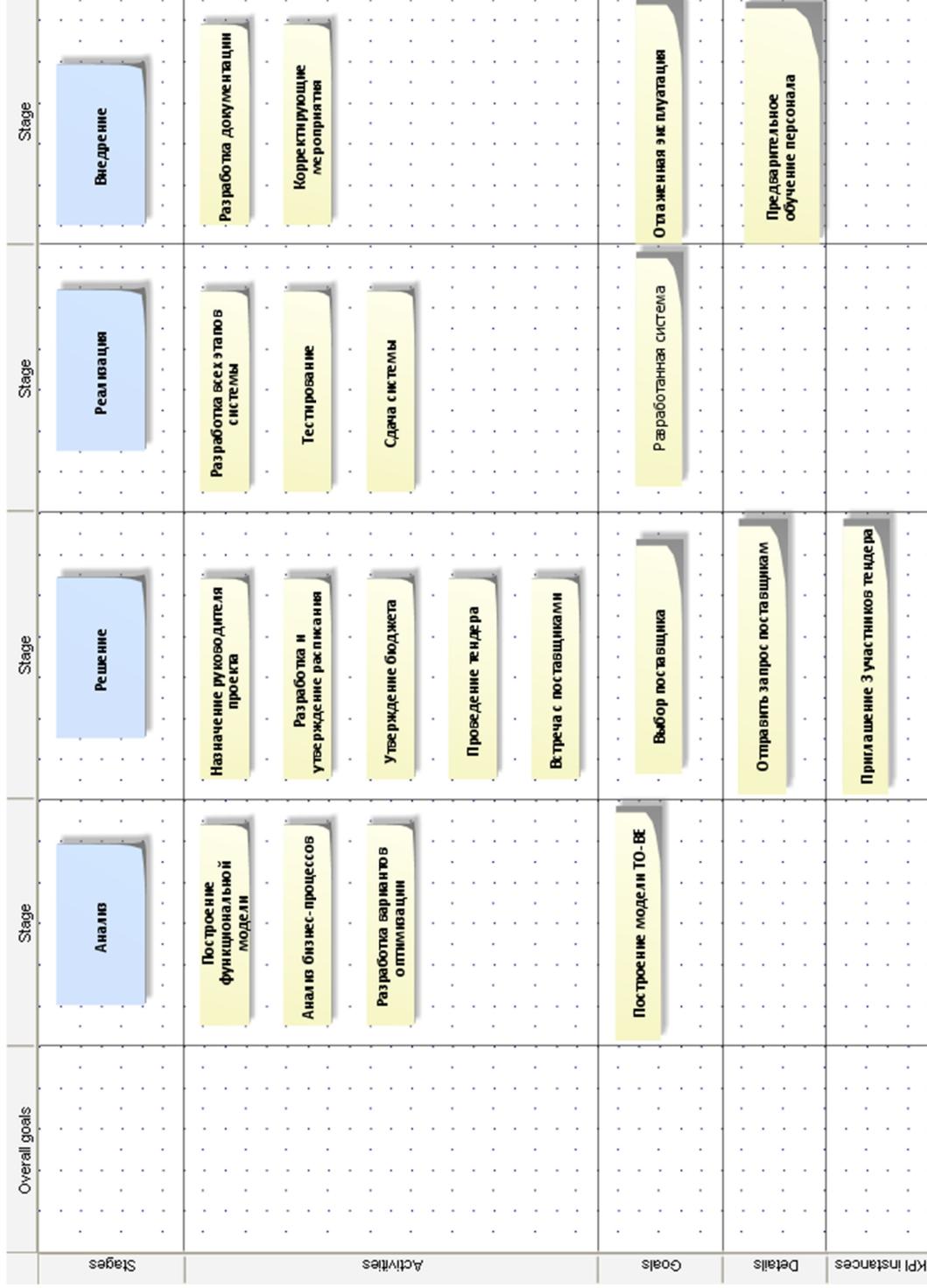


Рис. 4.11. Диаграмма модели белой доски для этапа разработки ИС компании «Стиль для всех»

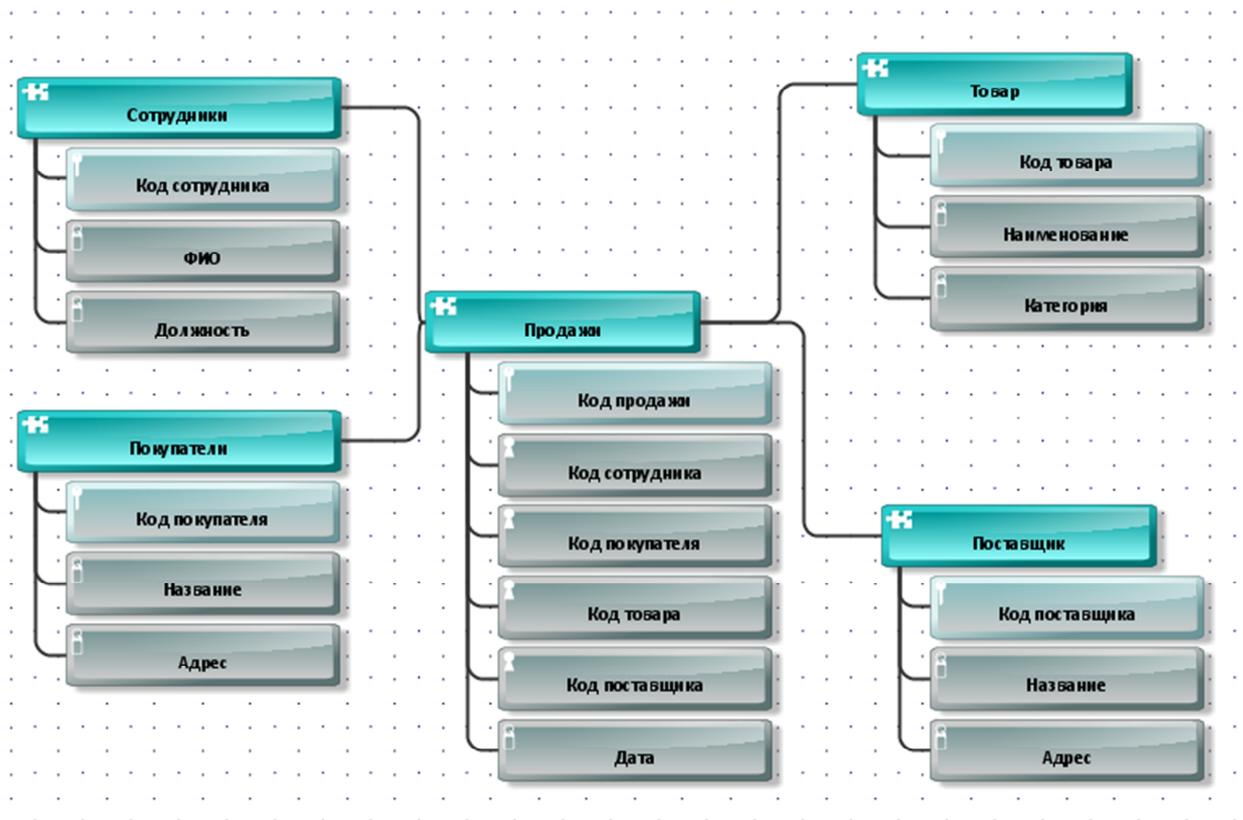


Рис. 4.12. Модель данных для интернет магазина «Стиль для всех»

ЛАБОРАТОРНАЯ РАБОТА № 20

Разработка комплексной модели предприятия по методологии ARIS

Цель работы: закрепление навыков работы с программой ARIS Express.

ПРАКТИЧЕСКАЯ ЧАСТЬ

На основании выполненных лабораторных работ по методологии ARIS разработать комплексную модель компании по своему выбору. Модель должна содержать следующие составляющие:

1. Организационную структуру с полным описанием штатных единиц и должностей;
2. Цепочку добавленного качества;
3. Описать основные процессы в нотации EPC;
4. Этапы создания модели описать на модели интерактивной доски;
5. Разработать структуру данных любой ИС, используемой в работе компании.

По итогам работы составить отчет о выполнении работы. Все темы предварительно согласуются с преподавателем.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ И ИСТОЧНИКОВ

1. Менеджмент качества. [Электронный ресурс]. Режим доступа: http://www.kpms.ru/QMS_automation.htm (Дата обращения 30.11.2017 г.)
2. Ильина О.П. Методические указания по изучению модуля «Информационные технологии процессного управления в менеджменте» по дисциплине «Информационные технологии в менеджменте» /СПбГУЭФ, 2012. – Санкт-Петербург. - 93 с.
3. ARIS. Материал из Википедии свободной энциклопедии. [Электронный ресурс] Режим доступа: <https://ru.wikipedia.org/wiki/ARIS> (Дата обращения 19.02.2018 г.)
4. Грекул В.И. Проектирование информационных систем: Информация. Интернет-Университет информационных технологий – НОУ ИНТУИТ. [Электронный ресурс]. Режим доступа: <https://www.intuit.ru/studies/courses/2195/55/info> (Дата обращения 18.12.2017 г.);
5. Методы и модели информационного менеджмента / Д.В. Александров, Р.И. Макаров, Е.Р. Хорошева; под. ред. А.В. Кострова. – М.: Финансы и статистика, 2007. - 336 с.;
6. Построение IDEF3 и IDEF0 диаграмм. [Электронный ресурс]. Режим доступа: http://fevt.ru/publ/dia_idef3_idef0/12-1-0-196 (Дата обращения 19.02.2018).
7. ELMA. Журнал о процессах, эффективности и управлении. Урок 1: Ввод в нотацию BPMN. [Электронный ресурс]. Режим доступа: https://www.elma-bpm.ru/journal/index.php?ELEMENT_ID=2894. (Дата обращения 19.02.2018).
8. ELMA. Журнал о процессах, эффективности и управлении. Урок 4: Практическое использование подпроцессов в BPMN. [Электронный ресурс]. Режим доступа: https://www.elma-bpm.ru/journal/index.php?ELEMENT_ID=2897. (Дата обращения 19.02.2018).
9. Краткое описание BPMN с примером. [Электронный ресурс]. Режим доступа: <https://habrahabr.ru/company/trinion/blog/331254/> (Дата обращения 19.02.2018).
10. Бабич А.В. Введение в UML: Информация – Интернет-Университет информационных технологий – НОУ ИНТУИТ [Электронный ресурс]. Режим доступа: <https://www.intuit.ru/studies/courses/1007/229/info> (Дата обращения 17.12.2017 г.);
11. Фаулер М. UML. Краткое руководство по стандартному языку объектного моделирования / М. Фаулер. – СПб.: Символ-Плюс, 2011. – 192 с.
12. Леоненков А. В. Нотация и семантика языка UML. Лекция 3. «Элементы графической нотации диаграммы вариантов использования». Интернет-университет

информационных технологий - НОУ ИНТУИТ. [Электронный ресурс]. – Режим доступа: <https://www.intuit.ru/studies/courses/32/32/lecture/1004> (Дата обращения 17.12.2017 г.).

13. Киммел П. UML. Универсальный язык программирования / П. Киммел. – М.: НТ-Пресс, 2008. – 272 с.;

14. Буч Г., Рамбо Дж., Якобсон И. Введение в UML от создателей языка / Г. Буч, Дж. Рамбо, И. Якобсон. – М.: ДМК Пресс, 2011. – 496 с.;

15. Стасышина Т.Л. Диаграммы прецедентов. [Электронный ресурс]. - Режим доступа: https://ciu.nstu.ru/kaf/persons/1914/page47048/diagramm_precedentov (Дата обращения 17.12.2017 г.);

16. Лекция 3. Виды диаграмм UML. Диаграммы классов. НОУ ИНТУИТ. Открытый национальный университет. [Электронный ресурс]. – Режим доступа: <https://www.intuit.ru/studies/courses/1007/229/lecture/5954?page=2>. (Дата обращения 19.02.2018).

17. Диаграмма классов. Материал из Википедии свободной энциклопедии [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%BE%D0%B2 (Дата обращения 19.02.2018).

18. INFORMICUS. Диаграммы классов UML. Логическое моделирование. [Электронный ресурс]. - <http://www.informicus.ru/Default.aspx?SECTION=6&id=73&subdivisionid=3>. (Дата обращения 19.02.2018).

19. Лекция 4. Диаграммы классов: крупным планом. НОУ ИНТУИТ Национальный открытый университет. [Электронный ресурс]. - <https://www.intuit.ru/studies/courses/1007/229/lecture/5956?page=3> (Дата обращения 19.02.2018).

20. Леоненков А.В. Нотация и семантика языка UML. Лекция 11. «Элементы графической нотации диаграммы деятельности». – Интернет-университет информационных технологий НОУ ИНТУИТ. [Электронный ресурс]. – Режим доступа: <https://www.intuit.ru/studies/courses/32/32/lecture/1020> (Дата обращения 17.12.2017 г.);

21. Леоненков А. В. Нотация и семантика языка UML. Лекция 8 «Элементы графической нотации диаграммы последовательности». Интернет-университет информационных технологий – НОУ ИНТУИТ. [Электронный ресурс]. – Режим доступа: <https://www.intuit.ru/studies/courses/32/32/lecture/1014> (Дата обращения 17.12.2017 г.);

22. ARIS Express – бесплатная программа для моделирования бизнес-процессов и оргструктуры / BPMSoft [Электронный ресурс]. – Режим доступа: <https://bpmsoft.org/aris-express/> (Дата обращения 19.02.2018).

23. Морозова В.И., Врублевский К.Э. Моделирование бизнес-процессов с использованием методологии ARIS: учебно-методическое пособие. / В.И. Морозова, К.Э. Врублевский. – М.: РУТ (МИИТ), 2017. – 47 с.

24. Использование нотации EPC для графического описания бизнес-процессов / Хабрахабр. [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/143273/>. (Дата обращения 19.02.2018).