## **Тема работы: «Функциональное моделирование с использованием пакета All Fusion Process Modeler»**

**1. Цель работы**

Формирование умений и навыков создания функциональной модели с использованием пакета All Fusion Process Modeler.

**2. Задание**

**Разработать иерархию функциональных диаграмм в соответствии с индивидуальным заданием.**

**3. Оснащение работы**

ПК, текстовый редактор, инструмент для визуального моделирования All Fusion Process Modeler.

**4. Основные теоретические сведения**

Дуглас Т. Росс предложил специальную технику моделирования, получившую название SADT (Structured Analysis & Design Technique). Метод SADT (IDEF0) (Structured Analysis and Design Technique) считается классическим методом процессного подхода к управлению. Основной принцип процессного подхода заключается в структурировании деятельности организации в соответствии с ее бизнес-процессами, а не организационно-штатной структурой. Именно бизнес-процессы, формирующие значимый для потребителя результат, представляют ценность, и именно их улучшением предстоит в дальнейшем заниматься.

В соответствии с этим принципом бизнес-модель должна выглядеть следующим образом:

1) верхний уровень модели должен отражать только контекст системы – взаимодействие моделируемого единственным контекстным процессом предприятия с внешним миром;

2) на втором уровне модели должны быть отражены основные виды деятельности (тематически сгруппированные бизнес-процессы) предприятия и их взаимосвязи. В случае большого их количества некоторые из них можно вынести на третий уровень модели. Но в любом случае под виды деятельности необходимо отводить не более двух уровней модели;

3) дальнейшая детализация бизнес-процессов осуществляется посредством бизнес-функций — совокупностей операций, сгруппированных по определенным признакам. Бизнес-функции детализируются с помощью элементарных бизнес-операций;

4) описание элементарной бизнес-операции осуществляется посредством задания алгоритма ее выполнения.

Методология IDEF-SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели системы какой-либо предметной области. Функциональная модель SADT отображает структуру процессов функционирования системы и ее отдельных подсистем, т.е. выполняемые ими действия и связи между этими действиями. Для этой цели строятся специальные модели, которые позволяют в наглядной форме представить последовательность определенных действий.

Основные элементы этого метода основываются на следующих концепциях:

- графическое представление блочного моделирования. Графика блоков и дуг SADT-диаграммы отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, соответственно входящими в блок и выходящими из него. Взаимодействие блоков друг с другом описывается посредством интерфейсных дуг, выражающих «ограничения», которые, в свою очередь, определяют, когда и каким образом функции выполняются и управляются;

- строгость и точность. Выполнение правил SADT требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналитика. Правила SADT включают: ограничение количества блоков на каждом уровне декомпозиции (правило 3-6 блоков — ограничение мощности краткосрочной памяти человека), связность диаграмм (номера блоков), уникальность меток и наименований (отсутствие повторяющихся имен), синтаксические правила для графики (блоков и дуг), разделение входов и управлений (правило определения роли данных);

- отделение организации от функции, т.е. исключение влияния административной структуры организации на функциональную модель.

Метод SADT может использоваться для моделирования самых разнообразных процессов и систем. Т.е. объектами моделирования являются системы.

В существующих системах метод SADT может быть использован для анализа функций, выполняемых системой, и указания механизмов, посредством которых они осуществляются.

Таким образом, при моделировании необходимо придерживаться следующих правил:

1) на диаграмме должно быть не менее 3-х и не более 6-и функциональных Блоков. Практика показывает, что этого вполне достаточно для построения полной функциональной модели современного предприятия любой отрасли;

2) диаграммы должны отображать информацию, не выходящую за рамки контекста, определенного целью и точкой зрения;

3) диаграммы должны иметь связанный интерфейс, когда номера Блоков, Дуги и ICOM коды имеют единую структуру;

4) уникальность имен функций Блоков и наименований Дуг;

5) четкое определение роли данных и разделение входов и управлений;

6) замечания для Дуг и имена функций Блоков должны быть краткими и лаконичными;

7) для каждого функционального Блока необходима как минимум одна управляющая Дуга;

8) модель всегда строится с определенной целью и с позиций конкретной точки зрения.

В процессе моделирования очень важным является четко определить направление разработки модели – ее контекст, точку зрения и цель.

**Контекст** модели очерчивает границы моделируемой системы и описывает ее взаимосвязи с внешней средой.

**Точка зрения** определяет позицию автора, т.е. что будет рассматриваться и под каким углом зрения.

Необходимо помнить, что одна модель представляет одну точку зрения. Для моделирования системы с нескольких точек зрения используется несколько моделей.

**Цель** отражает причину создания модели и определяет ее назначение. При этом все взаимодействия в модели рассматриваются именно с точки зрения достижения поставленной цели.

Модель IDEF-SADT представляет собой серию иерархически взаимосвязанных диаграмм с сопроводительной документацией, которая разбивает исходное представление сложной системы на отдельные составные части. Детали каждого из основных процессов представляются в виде более детальных процессов на других диаграммах. В этом случае каждая диаграмма нижнего уровня является декомпозицией некоторого процесса из более общей диаграммы. Описание модели построено в виде иерархической пирамиды, в вершине которой представляется самое общее описание системы, а основание представляет собой множество более детальных описаний. Поэтому на каждом шаге декомпозиции более общая диаграмма конкретизируется на ряд более детальных диаграмм.

Одной из наиболее важных особенностей метода SADT является постепенное введение все больших уровней детализации по мере создания диаграмм, отображающих модель.

Диаграммы более высокого уровня (А-0, А0) – являются наиболее общим описанием системы, представленным в виде отдельных Блоков. Декомпозиция этих Блоков позволяет достигать требуемого уровня детализации описания системы.

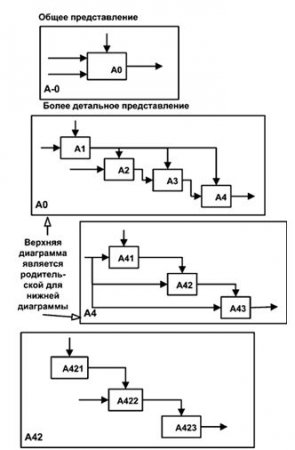
Разработка IDEF0 Диаграмм начинается с построения самого верхнего уровня иерархии (А-0) – одного Блока и интерфейсных Дуг, описывающих внешние связи рассматриваемой системы. Имя функции, записываемое в Блоке 0, является целевой функцией системы с принятой точки зрения и цели построения модели.

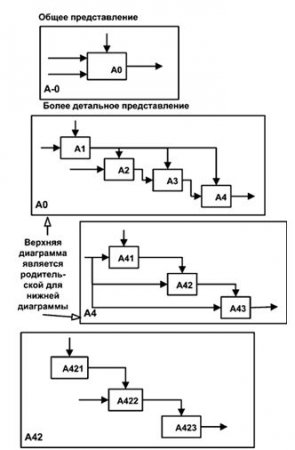
При дальнейшем моделировании Блок 0 декомпозируется на Диаграмме А0, где целевая функция уточняется с помощью нескольких Блоков, взаимодействие между которыми описывается с помощью Дуг. В свою очередь, функциональные Блоки на Диаграмме А0 могут быть также декомпозированы для более детального представления.

В результате, имена функциональных Блоков и интерфейсные Дуги, описывающие взаимодействие всех Блоков, представленных на Диаграммах, образуют иерархическую взаимосогласованную модель.

Хотя вершиной модели является Диаграмма уровня А-0, настоящей «рабочей вершиной или структурой» является Диаграмма А0, поскольку она является уточненным выражением точки зрения модели. Ее содержание показывает, что будет рассматриваться в дальнейшем, ограничивая последующие уровни в рамках цели проекта. Нижние уровни уточняют содержание функциональных Блоков, детализируя их, но, не расширяя границ модели.

На рисунке 2.1, где приведены четыре диаграммы и их взаимосвязи, показана структура SADT-модели. Каждый компонент модели может быть декомпозирован на другой диаграмме. Каждая диаграмма иллюстрирует «внутреннее строение» блока на родительской диаграмме.

[](http://e-educ.ru/uploads/posts/2010-07/1279719597_snimok.jpg)

[](http://e-educ.ru/uploads/posts/2010-07/1279719597_snimok.jpg)

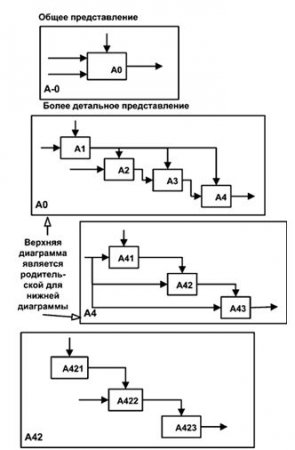
[](http://e-educ.ru/uploads/posts/2010-07/1279719597_snimok.jpg)

Рисунок 4.1 – Структура SADT-модели: декомпозиция диаграмм

Описание синтаксиса языка моделирования

Основными элементами на IDEF0 Диаграммах являются Блоки и Дуги.

Блоки служат для отображения функций (действий), выполняемых моделируемой системой. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении (например, “производить услуги”, а не “производство услуг”).

*глагол + объект действия + [дополнение]*

Пример: обрабатывать деталь на станке, передать документы в отдел, разработать план-график проведения анализа, опубликовать материалы...

Дуги служат для отображения информации или материальных объектов, которые необходимы для выполнения функции или появляются в результате ее выполнения (объекты, обрабатываемые системой). Такими объектами могут быть элементы реального мира (детали, вагоны, инструменты, станки, сотрудники и т.д.) или потоки данных и информации (документы, данные, инструкции и т.д.) и даже системы.

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

Различают стрелки четырех видов:

- I (Input) – вход, т. е. все, что поступает в процесс или потребляется процессом;

- С (Control) – управление или ограничения на выполнение операций процесса;

- О (Output) – выход или результат процесса;

- М (Mechanism) – механизм, который используется для выполнения процесса.

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

Необходимо отметить, что любой функциональный блок по требованиям стандарта должен иметь, по крайней мере, одну управляющую интерфейсную дугу и одну исходящую.

Деятельность представляет собой некоторое действие или набор действий, которые имеют фиксированную цель и приводят к некоторому конечному результату. Иногда деятельность называют просто процессом.

Методология IDEFO однозначно определяет, каким образом изображаются на диаграммах стрелки каждого вида ICOM. Стрелка Вход (Input) выходит из левой стороны рамки рабочего поля и входит слева в прямоугольник процесса. Стрелка Управление (Control) входит и выходит сверху. Стрелка Выход (Output) выходит из правой стороны процесса и входит в правую сторону рамки. Стрелка Механизм (Mechanism) входит в прямоугольник процесса снизу. Таким образом, базовое представление процесса на диаграммах IDEFO имеет следующий вид, как показано на рисунке 4.2.

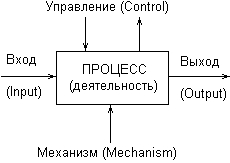


Рисунок 4.2 – Обозначение процесса и стрелок ICOM на диаграммах IDEF0

Техника построения диаграмм представляет собой главную особенность методологии IDEF-SADT. Место соединения стрелки с блоком определяет тип интерфейса, рисунок 4.3.

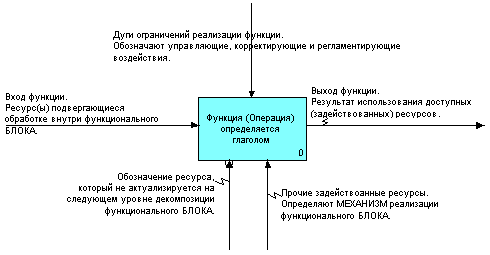


Рисунок 4.3 – Дуги, как ограничивающие и уточняющие факторы Блока

Функциональный блок преобразует входную информацию (данные, материалы, средства, задачи, цели и др.) в выходную (что требуется получить в результате выполнения данной функции). Управление определяет, когда и как это преобразование может или должно произойти. Механизм (или исполнители) непосредственно осуществляют это преобразование.

За каждой дугой закрепляется замечание, которое отображает суть информации или объекта. Замечание формулируется в виде оборота существительного, отвечающего на вопрос: «Что?».

Примерконтекстной диаграммы изображен на рисунке 4.4.

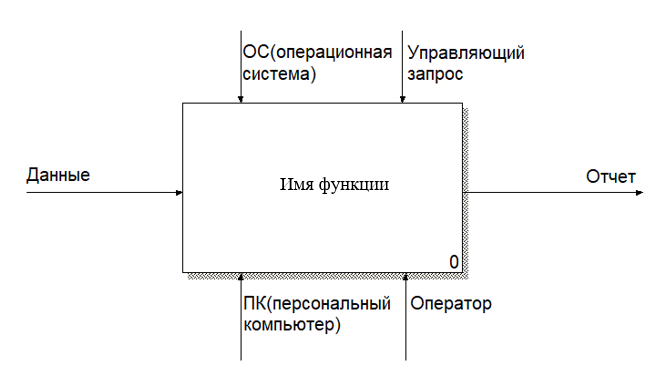


Рисунок 4.4 – Пример контекстной диаграммы IDEF0

Функциональные блоки на диаграмме изображаются в виде прямоугольников, внутри которых записывается имя функции и номер блока (в правом нижнем углу прямоугольника).

Блоки располагаются на диаграмме согласно их степени важности (по мнению автора модели). При этом доминирующим является тот блок, выполнение функции которого оказывает влияние на выполнение всех остальных функций, представленных на диаграмме. К примеру, это может быть блок, содержащий контролирующую или планирующую функцию, выходы которого являются управляющими для всех остальных функциональных блоков диаграммы.

Доминирующий блок помещается, как правило, в верхнем левом углу листа диаграммы, а наименее важный блок – в правом нижнем углу. Таким образом, ступенчатость блоков на диаграмме отражает мнение автора о доминировании одних блоков относительно других.

Очень важно помнить, что доминирование блоков на диаграмме не задаёт чёткой временной зависимости операций.

Стороны блока также имеют определенное значение. К левой границе блока присоединяются входные дуги, к верхней – управляющие дуги, к правой – выходные дуги, а к нижней – дуги механизмов.

При IDEF0 моделировании используются пять типов взаимосвязей между блоками, для описания их отношений.

1. Взаимосвязь по управлению, – когда выход одного блока влияет (является управляющей) на выполнение функции в другом блоке, рисунок 4.5.

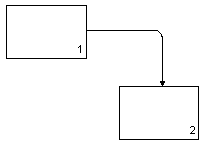


Рисунок 4.5 – Взаимосвязь по управлению

2. Взаимосвязь по входу, – когда выход одного блока является входом для другого, рисунок 4.6.

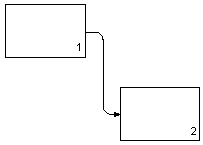


Рисунок 4.6 – Взаимосвязь по входу

3. Обратная связь по управлению, – когда выходы из одной функции влияют на выполнение других функций, выполнение которых в свою очередь влияет на выполнение исходной функции, рисунок 4.7.

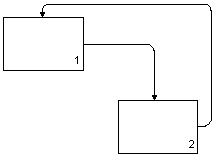


Рисунок 4.7. – Обратная связь по управлению

4. Обратная связь по входу, – когда выход из одной функции является входом для другой функции, выход которой является для него входом, рисунок 4.8.

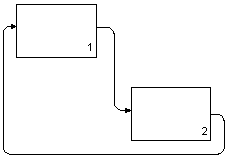


Рисунок 4.8. – Обратная связь по входу

5. Взаимосвязь «выход-механизм», – когда выход одной функции является механизмом для другой. Иначе говоря, выходная дуга одного блока является дугой механизма для другого. Такой тип связи встречается редко и относится чаще всего к подготовительным операциям, рисунок 4.9.

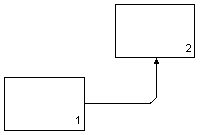


Рисунок 4.9. – Взаимосвязь «выход-механизм»

Построение диаграмм начинается с представления всей системы в виде простейшего компонента – одного блока и дуг, изображающих интерфейсы с функциями вне системы. Поскольку единственный блок отражает систему как единое целое, имя, указанное в блоке, является общим. Это верно и для интерфейсных дуг – они также соответствуют полному набору внешних интерфейсов системы в целом.

Затем блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки определяют основные подфункции исходной функции. Данная декомпозиция выявляет полный набор подфункций, каждая из которых показана как блок, границы которого определены интерфейсными дугами. Каждая из этих подфункций может быть декомпозирована подобным образом в целях большей детализации.

Во всех случаях каждая подфункция может содержать только те элементы, которые входят в исходную функцию. Кроме того, модель не может опустить какие-либо элементы, т.е., как уже отмечалось, родительский блок и его интерфейсы обеспечивают контекст. К нему нельзя ничего добавить, и из него не может быть ничего удалено.

Пример контекстной диаграммы и ее детализации представлен на рисунках 4.10, 4.11, 4.12.



Рисунок 4.10 – Пример контекстной диаграммы

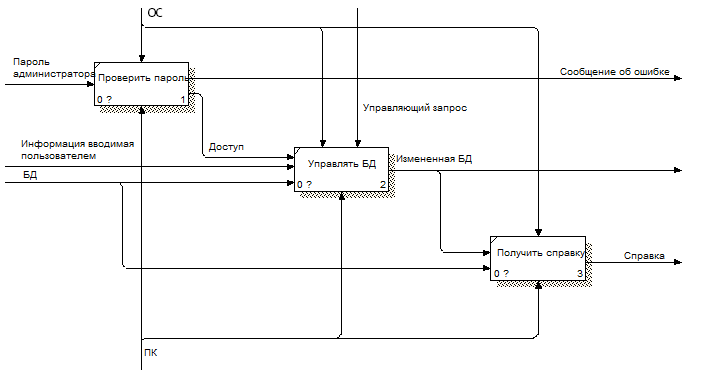


Рисунок 4.11 – Пример первого уровня детализации А0

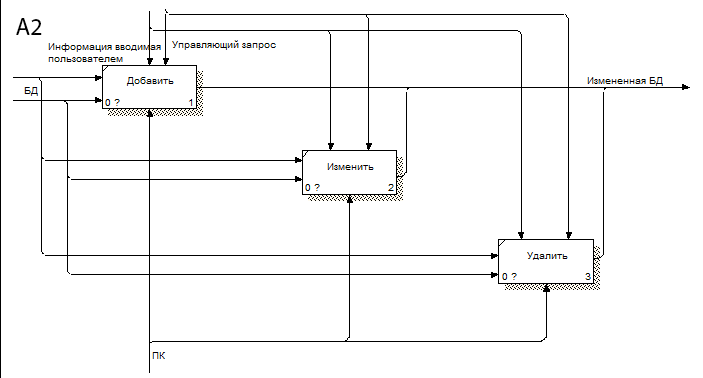


Рисунок 4.12 – Пример детализирующей диаграммы А2

AllFusion Process Modeler (далее BPwin) – CASE-средство для моделирования бизнес-процессов, позволяющая создавать диаграммы в нотации IDEF0, IDEF3, DFD. В процессе моделирования BPwin позволяет переключиться с нотации IDEF0 на любой ветви модели на нотацию IDEF3 или DFD и создать смешанную модель. BPwin поддерживает функционально-стоимостной анализ (ABC).

Работа с программой начинается с создания новой модели, для которой нужно указать имя и тип, рисунок 4.13.

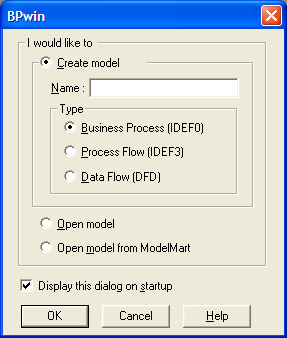


Рисунок 4.13 – Создание новой модели

От выбора типа модели зависит в каких нотациях можно производить декомпозицию работ. Так, если выбрать тип Business Process (IDEF0), то в созданной модели можно производить декомпозицию работ в нотациях IDEF0, IDEF3 и DFD; если выбран тип Data Flow (DFD) – в нотациях DFD и IDEF3; если же выбран тип Process Flow (IDEF3) – то только в нотации IDEF3.

После ввода имени модели и выбора ее типа BPWin сразу предложит задать параметры модели, рисунок 4.14:

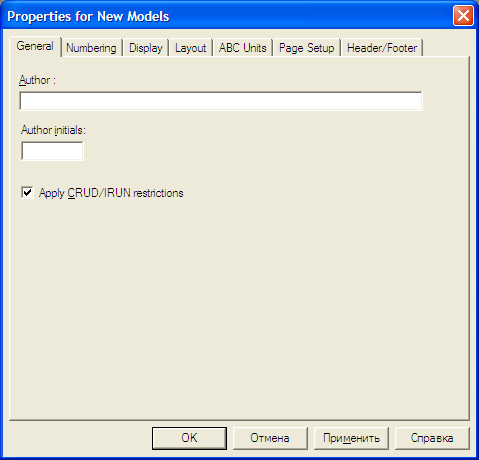


Рисунок 4.14 – Окно задания свойств модели

General – автор модели и его инициалы;

Numbering – формат нумерации работ и диаграмм и порядок ее отображения на диаграммах;

Display – список элементов отображения на диаграммах;

Layout – параметры расположения;

ABC Units – единицы функционально-стоимостного анализа;

Page Setup – параметры страницы;

Header/Footer – параметры верхнего и нижнего колонтитула.

После задания свойств модели появляется главное окно программы, рисунок 4.15, состоящее из трех основных частей:

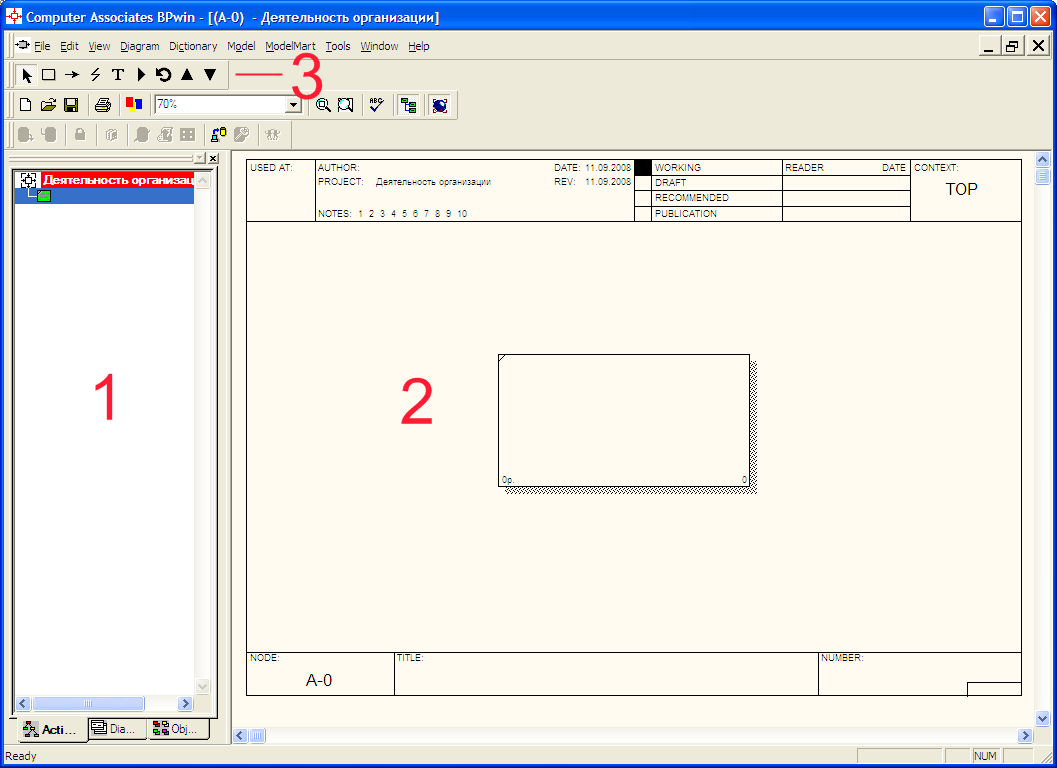


Рисунок 4.15 – Главное окно программы

1 – обозреватель модели (Model Explorer) – отображает структуру модели (имеющиеся диаграммы и их иерархию);

2 – основная часть – в ней отображаются диаграммы, с которыми ведется работа;

3 – панели инструментов, из которых наибольший интерес представляет панель инструментов Model Toolbox.

**Примечание.** В созданной модели с настройками по умолчанию некорректно отображаются русские символы. Чтобы устранить этот недостаток, необходимо подкорректировать используемые в модели шрифты. Для этого в меню Model -> Default Fonts необходимо последовательно пройтись по всем пунктам (рисунок 4.16), выбрать в выпадающем списке Script значение кириллический и поставить галочку Change all occurrences, рисунок 4.17.

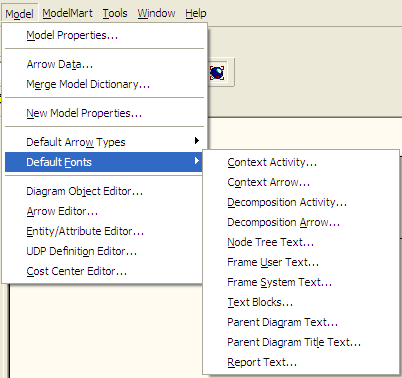


Рисунок 4.16 – Пункты меню, отвечающие за настройки шрифта

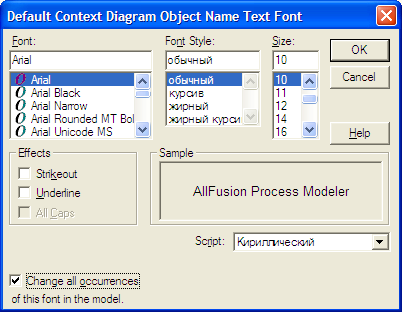


Рисунок 4.17 – Параметры шрифта

Панель инструментов Model Toolbox отвечает за создание разнообразных графических элементов модели. В зависимости от типа текущей диаграммы набор кнопок на ней меняется. Вид и назначение кнопок Model Toolbox приведены в таблице 4.1.

Таблица 4.1 – Вид и назначение кнопок Model Toolbox

|  |  |  |
| --- | --- | --- |
| **Вид кнопки** | **Название кнопки** | **Назначение кнопки** |
| pointer tool | Pointer Tool | Превращает курсор в стрелку указателя для того, чтобы можно было выделять объекты |
| Activity Box Tool  - IDEF0  Activity Box Tool  - DFD  Activity Box Tool  - IDEF3 | Activity Box Tool | Добавление на диаграмму новой работы |
| Precedence Arrow Tool | Precedence Arrow Tool | Добавление на диаграмму новой стрелки |
| Squiggle Tool | Squiggle Tool | Связывание названия стрелки с самой стрелкой |
| Text Tool | Text Tool | Добавление на диаграмму текста |
| Diagram Dictionary Editor | Diagram Dictionary Editor | Вызов окна менеджера диаграмм для просмотра имеющихся диаграмм по типам и переход к выбранной |
| Go to Sibling Diagram | Go to Sibling Diagram | Переход между стандартной диаграммой, деревом узлов и FEO диаграммой |
| Go to Parent Diagram | Go to Parent Diagram | Переход к родительской диаграмме |
| Go to Child Diagram | Go to Child Diagram | Переход к дочерней диаграмме |
| External Reference Tool   - DFD | External Reference Tool | Добавление на диаграмму внешней сущности |
| Data store Tool  - DFD | Data store Tool | Добавление на диаграмму хранилища данных |
| Junction Tool  - IDEF3 | Junction Tool | Добавление на диаграмму перекрестка |
| Referent Tool  - IDEF3 | Referent Tool | Добавление на диаграмму объекта ссылки |

Созданная модель уже содержит контекстную диаграмму с единственной работой ("черный ящик") в той нотации, которая была выбрана на этапе создания модели. Теперь необходимо дать этой работе название и при необходимости задать ее свойства. Для этого нужно вызвать окно свойств работы, дважды щелкнуть по ней мышью, рисунок 4.18.

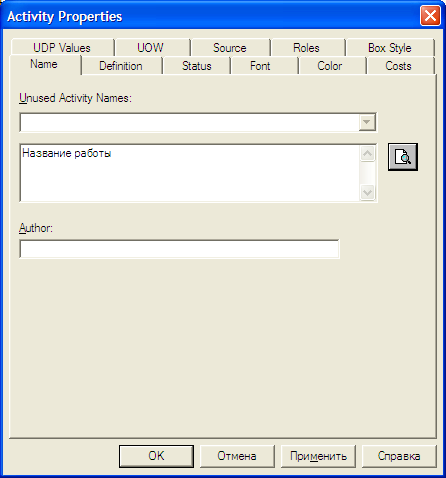


Рисунок 4.18 – Окно свойств работы

Далее необходимо разместить на диаграмме стрелки. Для этого следует нажать на Model Toolbox кнопку Precedence Arrow Tool (курсор примет форму крестика со стрелкой), щелкнуть по тому месту, откуда стрелка должна выходить и затем щелкнуть по тому месту, куда стрелка должна заходить (BPwin подсветит эти места при наведении на них курсора). Для задания названия стрелки нужно нажать на Model Toolbox кнопку Pointer Tool и затем дважды щелкнуть по стрелке. В появившемся окне Arrow Properties название работы вводится в поле Arrow Name или выбирается из списка имеющихся названий стрелок.

После размещения стрелок на диаграмме можно проводить декомпозицию ее работ. Для этого следует нажать на Model Toolbox кнопку Go to Child Diagram и затем щелкнуть по работе, которую нужно декомпозировать. Появится окно, в котором необходимо выбрать в какой нотации проводить декомпозицию и количество дочерних работ, рисунок 4.19.

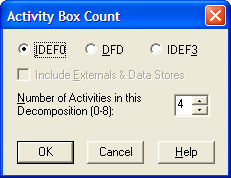


Рисунок 4.19 – Создание дочерней диаграммы

После создания дочерней диаграммы BPwin автоматически создаст указанное число работ и разместит граничные стрелки по краям диаграммы. Далее следует связать граничные стрелки со входами работ (при необходимости можно добавить новые граничные стрелки) и связать работы между собой. Дальнейшая декомпозиция работ проводится аналогичным образом.

**5. Порядок выполнения работы**

1. Построить диаграмму уровня А-0, состоящую из одного блока и дуг, изображающих интерфейсы с функциями вне системы в соответствии с индивидуальным заданием.

2. Выполнить детализацию диаграммы А-0 посредством построения диаграммы А0, содержащей основные подфункции исходной задачи.

3. Выполнить декомпозицию функциональных блоков до необходимой детализации.

4. Построить иерархию функциональных диаграмм с помощью CASE-средства для моделирования бизнес-процессов All Fusion Process Modeler.

**6. Контрольные вопросы и задания**

1. В чем суть методологии SADT?

2. Что является объектом моделирования в методе SADT?

3. Назовите основные элементы IDEF0.

**7. Рекомендуемая литература**

1. **Дубейковский, В.** Эффективное моделирование с AllFusion Process Modeler 4.1.4 и AllFusion PM / В. И. Дубейковский. Изд: "ДИАЛОГ-МИФИ", 2007
2. **Коровкина, Н.** Проектирование информационных систем. Учебник и практикум для СПО / [Н. Коровкина](https://www.google.by/search?hl=ru&tbo=p&tbm=bks&q=inauthor:%22%D0%9D%D0%B8%D0%BD%D0%B0+%D0%9A%D0%BE%D1%80%D0%BE%D0%B2%D0%BA%D0%B8%D0%BD%D0%B0%22&source=gbs_metadata_r&cad=5), [Г. Левочкина](https://www.google.by/search?hl=ru&tbo=p&tbm=bks&q=inauthor:%22%D0%93%D0%B0%D0%BB%D0%B8%D0%BD%D0%B0+%D0%9B%D0%B5%D0%B2%D0%BE%D1%87%D0%BA%D0%B8%D0%BD%D0%B0%22&source=gbs_metadata_r&cad=5), [В. Грекул](https://www.google.by/search?hl=ru&tbo=p&tbm=bks&q=inauthor:%22%D0%92%D0%BB%D0%B0%D0%B4%D0%B8%D0%BC%D0%B8%D1%80+%D0%93%D1%80%D0%B5%D0%BA%D1%83%D0%BB%22&source=gbs_metadata_r&cad=5) – Litres, 2019

# **Практическая работа № 5 (6 часов)**

## **Тема работы: «Моделирование бизнес-процесса»**

**1. Цель работы**

Формирование умений и навыков создания модели IDEF3 и модели бизнес-процесса BPMN.

**2. Задание**

Разработать модели IDEF3 и BPMN в соответствии с индивидуальным заданием.

**3. Оснащение работы**

ПК, текстовый редактор, CASЕ-средства для разработки моделей бизнесс-процесса.

**4. Основные теоретические сведения**

Одним из средств, используемых на этапе структурного анализа, являются модель IDEF3, представляющая собой способ описания процессов с использованием структурированного метода, позволяющего эксперту в предметной области представить положение вещей как упорядоченную последовательность событий с одновременным описанием объектов, имеющих непосредственное отношение к процессу.

Основой модели IDEF3 служит так называемый сценарий бизнес-процесса, который выделяет последовательность действий или подпроцессов анализируемой системы. Поскольку сценарий определяет назначение и границы модели, довольно важным является подбор подходящего наименования для обозначения действий. Для подбора необходимого имени применяются стандартные рекомендации по предпочтительному использованию глаголов и отглагольных существительных, например «обработать заказ клиента» или «обработка заказа клиента».

Компонентами модели являются: действие, связь,соединение или перекресток, указатель.

Диаграммы IDEF3отображают действие в виде прямоугольника. Действия именуются с использованием глаголов или отглагольных существительных. Действиям присваивается уникальный иерархический идентификационный номер. Этот номер не используется вновь даже в том случае, если в процессе построения модели действие удаляется. На рисунке 5.1 представлено изображение действия.

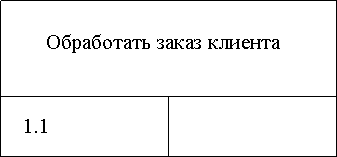


Рисунок 5.1 - Изображение и нумерация действия в диаграмме IDEF3

Связи выделяют существенные взаимоотношения между действиями. Все связи в IDEF3 являются однонаправленными, и, хотя стрелка может начинаться или заканчиваться на любой стороне блока, обозначающего действие, диаграммы IDEF3 обычно организуются слева направо таким образом, что стрелки начинаются на правой и заканчиваются на левой стороне блоков. В приведенной ниже таблице приведены три возможных типа связей.

Таблица 5.1 – Типы связей

|  |  |  |
| --- | --- | --- |
| **Изображение** | **Название** | **Назначение** |
| image032 | Временное предшествование (Temporal pre­cedence) | Исходное действие должно завершиться, прежде чем конечное действие сможет начаться |
| image032 | Объектный поток (Object flow) | Выход исходного действия является входом конечного действия. Из этого, в частности, следует, что исходное действие должно завершиться, прежде чем конечное действие сможет начаться |
| image032 | Нечеткое отношение (Relationship) | Вид взаимодействия между исходным и конечным действиями задается анали­тиком отдельно для каждого случая ис­пользования такого отношения |

Связь типа *«временное предшествование»* показывает, что исходное действие должно полностью завершиться, прежде чем начнется выполнение конечного действия. Связь должна быть поименована таким образом, чтобы человеку, просматривающему модель, была понятна причина ее появления. Во многих случаях завершение одного действия инициирует начало выполнения другого, как показано на рисунке 5.2. В этом примере автор должен принять рекомендации рецензентов, прежде чем начать вносить соответствующие изменения в работу.

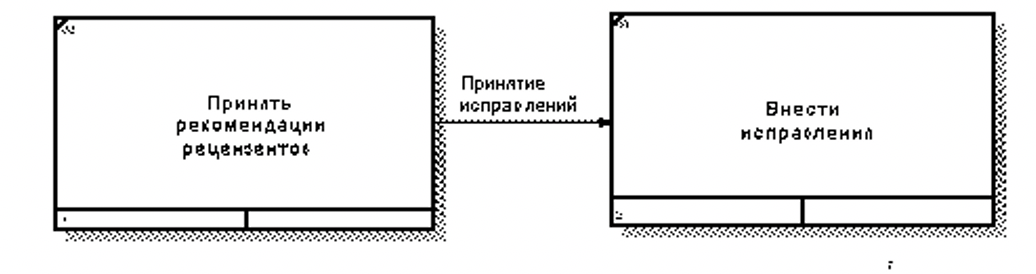


Рисунок 5.2 - Связь типа “временное предшествование” между действиями 1 и 2.

Связьтипа *«объектный поток»*. Одна из наиболее часто встречающихся причин использования связи типа «объектный поток» заключается в том, что некоторый объект, являющийся результатом выполнения исходного действия, необходим для выполнения конечного действия. Такая связь обозначается двойной стрелкой. Наименования потоковых связей должны четко идентифицировать объект, который передается с их помощью. Временная семантика объектных связей аналогична связям предшествования, это означает, что порождающее объектную связь исходное действие должно завершиться, прежде чем конечное действие может начать выполняться. Связь типа объектный поток представлена на рисунке 5.3.

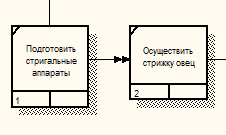


Рисунок 5.3 - Связь типа “объектный поток” между действиями 1 и 2.

Связьтипа *«нечеткое отношение».* Связи этого типа используются для выделения отношений между действиями, которые невозможно описать с использованием объектных связей. Значение каждой такой связи должно быть определено, поскольку связи типа «нечеткое отношение» сами по себе не предполагают никаких ограничений. Одно из применений нечетких отношений — отображение взаимоотношений между параллельно выполняющимися действиями. Наиболее часто нечеткие отношения используются для описания специальных случаев связей предшествования, например для описания альтернативных вариантов временного предшествования. В таблице 5.2 объединены три типа соединений.

Таблица 5.2 – Типы объединений

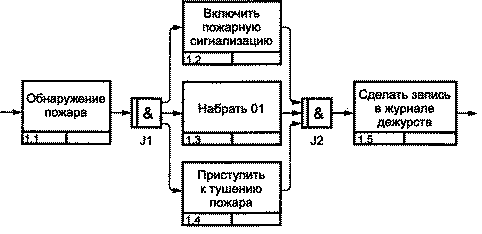
|  |  |  |  |
| --- | --- | --- | --- |
| **Графическое обозначение** | **Название** | **Вид** | **Правила инициации** |
| Статья 54 - Картинка 2 | Соединение «и» | Разворачи­вающее | Каждое конечное действие обязательно инициируется |
|  |  | Сворачи­вающее | Каждое исходное действие обязательно должно завершиться |
| Статья 54 - Картинка 3 | Соединение «эксклюзивное "или"» | Разворачи­вающее | Одно и только одно конечное действие инициируется |
|  |  | Сворачи­вающее | Одно и только одно исходное действие должно завершиться |
| Статья 54 - Картинка 4 | Соединение «или» | Развора­чивающее | Одно или несколько конечных действий инициируются |
|  |  | Сворачи­вающее | Одно или несколько исходных действий должны завершиться |

Завершение одного действия может инициировать начало выполнения сразу нескольких других действий или, наоборот, определенное действие может требовать завершения нескольких других действий до начала своего выполнения. Соединениясоединяютвнутренние потоки и используются для описания ветвления процесса:

* разворачивающие соединения используются для разбиения потока. Завершение одного действия вызывает начало выполнения нескольких других;
* сворачивающие соединения объединяют потоки. Завершение одного или нескольких действий вызывает начало выполнения другого действия.

Рассмотрим примеры использования разворачивающих и сворачивающих соединений.

«И»-соединения. Соединения этого типа инициируют выполнение конечных действий. Все действия, присоединенные к сворачиваю­щему «и»-соединению, должны завершиться, прежде чем начнется выполнение следующего действия. На рисунке 5.4 после обнаружения пожара инициируются включение пожарной сигнализации, вызов пожарной охраны, и начинается тушение пожара. Запись в журнал производится только тогда, когда все три перечисленных действия завершены.

  
Рисунок 5.4 - “И”- cоединения

Соединение «эксклюзивное "или"». Вне зависимости от количества действий, связанных со сворачивающим или разворачивающим соединением «эксклюзивное «или», инициировано будет только одно из них, и поэтому только оно будет завершено перед тем, как любое действие, следующее за сворачивающим соединением «эксклюзивное «или», сможет начаться. Если правила активации соединения известны, они обязательно должны быть документированы либо в его описании, либо пометкой стрелок, исходящих из разворачивающего соединения, как показано на рисунке 5.5.

На рисунке 5.5 соединение «эксклюзивное «или» используется для отображения того факта, что студент не может одновременно быть направлен на лекции по двум разным курсам.

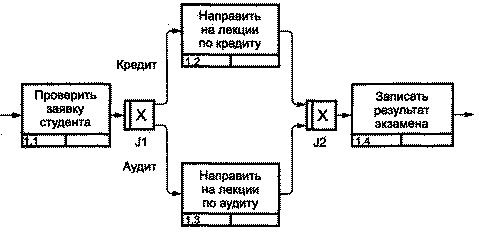


Рисунок 5.5 – Соединение «эксклюзивное “или”»

Соединение«или» предназначено для описания ситуаций, которые не могут быть описаны двумя предыдущими типами соединений. Аналогично связи нечеткого отношения соединение «или» в основном определяется и описывается непосредственно системным аналитиком.

Указатели — это специальные символы, которые ссылаются на другие разделы описания процесса. Они используются при построении диаграммы для привлечения внимания пользователя к каким-либо важным аспектам модели.

Указатель изображается на диаграмме в виде прямоугольника, похожего на изображение действия. Имя указателя обычно включает его тип (например, ОБЪЕКТ, UOB и т.п.) и идентификатор.

Таблица 5.3 – Указатели

|  |  |
| --- | --- |
| **Тип указателя** | **Назначение** |
| ОБЪЕКТ (OBJECT) | Для описания того, что в действии принимает участие какой-либо заслуживающий отдельного внимания объект |
| ССЫЛКА (GOTO) | Для реализации цикличности выполнения действий. Указатель ССЫЛКА может относиться и к соединению |
| ЕДИНИЦА ДЕЙ­СТВИЯ (Unit of Behavior — UOB) | Для многократного отображения на диаграмме одного и того же действия. Например, если действие «Подсчет наличных» выполняется несколько раз, в первый раз оно создается как действие, а последующие его появления на диаграмме оформляются указателями UOB |
| ЗАМЕТКА (NOTE) | Для документирования любой важной информации общего характера, относящейся к изображенному на диаграм­мах. В этом смысле ССЫЛКА служит альтернативой методу помещения текстовых заметок непосредственно на диаграммах |
| УТОЧНЕНИЕ (Ela­boration — ELAB) | Для уточнения или более подробного описания изобра­женного на диаграмме. Указатель УТОЧНЕНИЕ обычно используется для описания логики ветвления у соеди­нений |

На рисунке 5.6 изображен пример описания процесса с использованием методологии IDEF3.

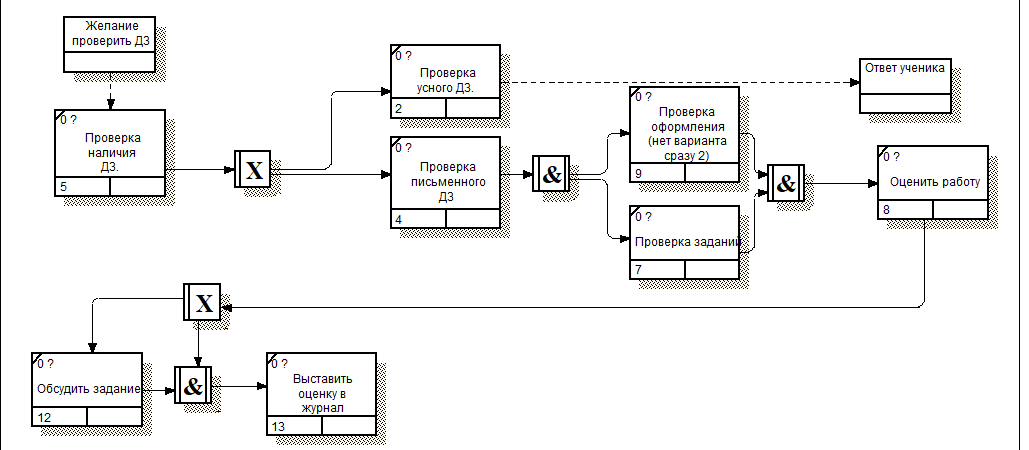


Рисунок 5.6 – Описание процесса в методологии IDEF3

BPMN от английского *Business Process Model and Notation* (*нотация и модель бизнес-процессов*), т.е. система условных обозначений и их описания в XML для [моделирования бизнес-процессов](https://ru.wikipedia.org/w/index.php?title=%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D0%B2&action=edit&redlink=1).

Основная цель BPMN — создание стандартного набора условных обозначений, понятных всем бизнес-пользователям. Моделирование в BPMN осуществляется посредством диаграмм с небольшим числом графических элементов. Выделяют четыре основные категории элементов:

* объекты потока управления: события, действия и логические операторы (развилки);
* 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.

Объекты потока управления разделяются на три основных типа: события (events), действия (activities) и логические операторы (gateways).

События изображаются окружностью и означают какое-либо происшествие. События инициируют действия или являются их результатами. Согласно расположению в процессе события могут быть классифицированы на начальные ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *start*), промежуточные (*intermediate*) и завершающие (*end*). Начиная с BPMN 1.1 различают события обработки и генерации.

На рисунке 5.7 представлены графические символы обозначения различных типов событий.

[](https://commons.wikimedia.org/wiki/File:BPMN_Events.png?uselang=ru)

Рисунок 5.7 – Графические символы обозначения различных типов событий

Простые события (*plain events*) это нетипизированные события, использующиеся чаще всего для того, чтобы показать начало или окончание процесса.

События-сообщения (*message events*) показывают получение и отправку сообщений в ходе выполнения процесса.

События-таймеры (*timer events*) моделируют события, регулярно происходящие во времени. Также позволяют моделировать моменты времени, периоды и тайм-ауты.

События-ошибки (*error events*) позволяют смоделировать генерацию и обработку ошибок в процессе. Ошибки могут иметь различные типы.

События-отмены (*cancel events*) инициируют или реагируют на отмену транзакции.

События-компенсации (*compensation events*) инициируют компенсацию или выполняют действия по компенсации.

События-условия (*conditional events*) позволяют интегрировать бизнес-правила в процесс.

Действияизображаются прямоугольниками со скругленными углами. Среди действий различают задания и подпроцессы. Графическое изображение свёрнутого подпроцесса снабжено знаком плюс у нижней границы прямоугольника.

Задание (*task*) — это единица работы, элементарное действие в процессе.

Множественные экземпляры (*multiple instances*) действия показывают, что одно действие выполняется многократно, по одному разу для каждого объекта. Например, для каждого объекта в заказе клиента выполняется один экземпляр действия. Экземпляры действия могут выполняться параллельно или последовательно.

Циклическое действие (*loop activity*) выполняется, пока условие цикла верно. Условие цикла может проверяться до или после выполнения действия.

Развёрнутый подпроцесс (*expanded subprocess*) является сложным действием и содержит внутри себя собственную диаграмму бизнес-процессов.

Свёрнутый подпроцесс (*collapsed subprocess*) также является составным действием, но скрывает детали реализации процесса.

Ad-hoc-подпроцесс (*ad-hoc subprocess*) содержит задания. Задания выполняются до тех пор, пока не выполнено условие завершения подпроцесса.

Графическое обозначение различных типов действий представлено на рисунке 5.8.

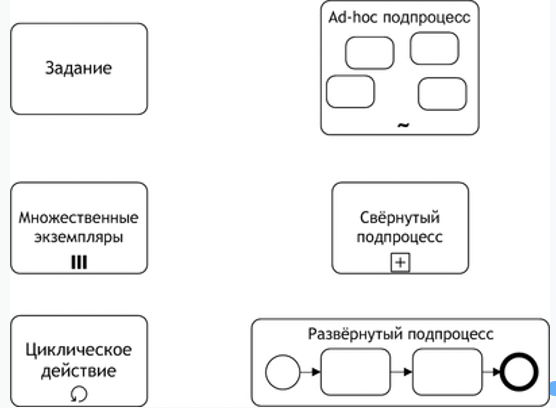


Рисунок 5.8 – Графическое обозначение различных типов действий

Логические операторы (развилки) изображаются ромбами и представляют точки принятия решений в процессе. С помощью логических операторов организуется ветвление и синхронизация потоков управления в модели процесса.

Оператор [исключающего *«или»*](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8E_2), управляемый данными ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *data-based exclusive gateway*). Если оператор используется для ветвления, то поток управления направляется лишь по одной исходящей ветви. Если оператор используется для синхронизации, то он ожидает завершения выполнения одной входящей ветви и активирует выходной поток.

Оператор исключающего *«или»*, управляемый событиями (*event-based exclusive gateway*) направляет поток управления лишь по той исходящей ветви, на которой первой произошло событие. После оператора данного типа могут следовать только события или действия-обработчики сообщений.

Оператор [включающего *«или»*](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B7%D1%8A%D1%8E%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) (*inclusive gateway*) активирует одну или более исходящих ветвей, в случае, когда осуществляется ветвление. Если оператор используется для синхронизации, то он ожидает завершения выполнения всех активированных ветвей и активирует выходной поток.

Оператор «[и](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%8A%D1%8E%D0%BD%D0%BA%D1%86%D0%B8%D1%8F)» (*parallel gateway*), использующийся для ветвления, разделяет один поток управления на несколько параллельных. При этом все исходящие ветви активируются одновременно. Если оператор используется для синхронизации, то он ожидает завершения выполнения всех входящих ветвей и лишь затем активирует выходной поток.

Сложный оператор (*complex gateway*) имеет несколько условий, в зависимости от выполнения которых активируются исходящие ветви. Оператор затрудняет понимание диаграммы, так как условия, определяющие семантику оператора, графически не выражены на диаграмме. Вследствие этого использование оператора нежелательно.

Символы логических операторов представлены на рисунке 5.9.

****

Рисунок 5.9 – Символы логических операторов

Логические операторы (развилки) изображаются ромбами и представляют точки принятия решений в процессе. С помощью логических операторов организуется ветвление и синхронизизация потоков управления в модели процесса.

Оператор [исключающего *«или»*](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8E_2), управляемый данными ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *data-based exclusive gateway*). Если оператор используется для ветвления, то поток управления направляется лишь по одной исходящей ветви. Если оператор используется для синхронизации, то он ожидает завершения выполнения одной входящей ветви и активирует выходной поток.

Оператор исключающего *«или»*, управляемый событиями (*event-based exclusive gateway*) направляет поток управления лишь по той исходящей ветви, на которой первой произошло событие. После оператора данного типа могут следовать только события или действия-обработчики сообщений.

Оператор [включающего *«или»*](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B7%D1%8A%D1%8E%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) (*inclusive gateway*) активирует одну или более исходящих ветвей, в случае, когда осуществляется ветвление. Если оператор используется для синхронизации, то он ожидает завершения выполнения всех активированных ветвей и активирует выходной поток.

Оператор «[и](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%8A%D1%8E%D0%BD%D0%BA%D1%86%D0%B8%D1%8F)» (*parallel gateway*), использующийся для ветвления, разделяет один поток управления на несколько параллельных. При этом все исходящие ветви активируются одновременно. Если оператор используется для синхронизации, то он ожидает завершения выполнения всех входящих ветвей и лишь затем активирует выходной поток.

Сложный оператор (*complex gateway*) имеет несколько условий, в зависимости от выполнения которых активируются исходящие ветви. Оператор затрудняет понимание диаграммы, так как условия, определяющие семантику оператора, графически не выражены на диаграмме. Вследствие этого использование оператора нежелательно.

Объекты потока управления связаны друг с другом соединяющими объектами. Существует три вида соединяющих объектов: потоки управления, потоки сообщений и ассоциации.

Поток управленияизображается сплошной линией, оканчивающейся закрашенной стрелкой. Поток управления задаёт порядок выполнения действий. Если линия потока управления перечеркнута диагональной чертой со стороны узла, из которого она исходит, то она обозначает поток, выполняемый по умолчанию. Типы потоков управления показаны на рисунке 5.10.

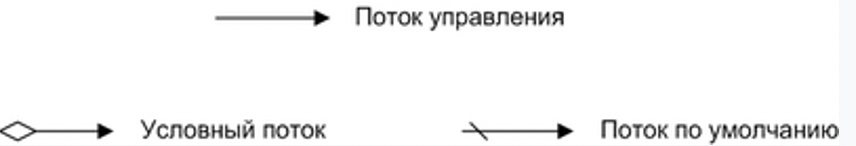


Рисунок 5.10 – Типы потоков управления

Поток сообщенийизображается штриховой линией, оканчивающейся открытой стрелкой, как показано на рисунке 5.11. Поток сообщений показывает, какими сообщениями обмениваются участники.

[https://upload.wikimedia.org/wikipedia/commons/c/cf/BPMN_MessageFlow.png](https://commons.wikimedia.org/wiki/File:BPMN_MessageFlow.png?uselang=ru)

Рисунок 5.11 – Поток сообщений

Ассоциацииизображаются пунктирной линией, заканчивающейся стрелкой (рисунок 5.12). Ассоциации используются для ассоциирования артефактов (данных или текстовых аннотаций) с объектами потока управления.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5.12 – Типы ассоциаций

Роли - визуальный механизм организации различных действий в категории со сходной функциональностью. Существует два типа ролей:

Пулы изображаются прямоугольником, который содержит несколько объектов потока управления, соединяющих объектов и артефактов.

Дорожки представляют собой часть пула. Дорожки позволяют организовать объекты потока управления, связывающие объекты и артефакты.

Типы полей показаны на рисунке 5.13.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 5.13 –Типы ролей

Артефакты позволяют разработчикам отображать дополнительную информацию в диаграмме. Это делает диаграмму более удобочитаемой и насыщенной информацией. Существуют три предопределённых вида артефактов:

Данные показывают читателю, какие данные необходимы действиям для выполнения и какие данные действия производят.

Группа изображается прямоугольником с закругленными углами, граница которого — штрихпунктирная линия. Группа позволяет объединять различные действия, но не влияет на поток управления в диаграмме.

Текстовые аннотации используются для уточнения значения элементов диаграммы и повышения её информативности.

Изображение различных типов артефактов представлено на рисунке 5.14.

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Рисунок 5.14 - Изображение различных типов артефактов

Пример диаграммы BPMN представлен на рисунке 5.15

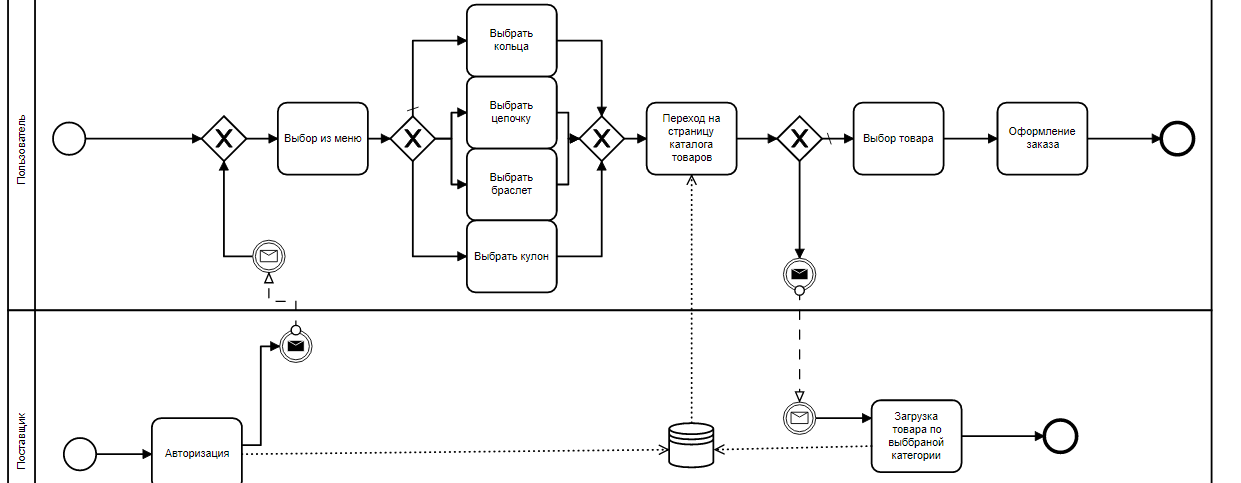


Рисунок 5.15 – Пример диаграммы BPMN

**5. Порядок выполнения работы**

1. Определить основные компоненты диаграммы IDEF3 и действия.

2. Определить связи и соединения между компонентами диаграммы IDEF3.

3. Построить диаграмму IDEF3.

4. Определить компоненты, необходимые для построения модели BPMN.

5. Определить связи модели BPMN.

6. Построить модель BPMN.

**6. Контрольные вопросы и задания**

1. Перечислите типы соединений, используемых при построении модели IDEF3.

2.В каких случаях используется соединение типа «и»?

3. В каких случаях используется соединение типа «или»?

4. В каких случаях используется соединение типа «эксклюзивное или»?

5. Для чего используются указатели?

6. Перечислить компоненты модели BPMN.

7. Описать назначение компонентов BPMN.

**7. Рекомендуемая литература**

1. **Маклаков, С. В.** Bpwin и Erwin. CASE-средства разработки информационных систем / С. В. Маклаков. М.: Диалог-МИФИ, 2016.

2. **Репин В.В., Елиферов В.Г.** Процессный подход к управлению. Моделирование бизнес-процессов. — М.: Манн, Иванов и Фербер, 2015

3. **Рудаков, А. В.** Технология разработки программных продуктов: учебное пособие для студентов учреждений среднего профессионального образования / А. В. Рудаков, Г. Н. Федорова. – 4-е изд., стер. – М.: Издательский центр «Академия»; 2016.

4. **Федоров И.Г.** Моделирование бизнес-процессов в нотации BPMN 2.0 / Научно-практическое издание. — М: МЭСИ, 2015