Business Process Modeling Notation Specification

Графический язык моделирования бизнес-процессов BPMN Спецификация (избранные главы)

Введение

Данный документ представляет собой субъективный перевод отдельных глав и параграфов спецификации языка моделирования бизнес-процессов BPMN (Business Process Management Notation).

Нотация BPMN была разработана организацией «Business Process Management Initiative (BPMI)», и поддерживается группой компаний «Object Management Group», после слияния организаций в 2005 году. Текущая версия BPMN – 1.1. Оригинальная спецификация изготовлена группой компаний «Object Management Group».

Нотация BPMN положена в основу системы управления бизнес-процессами «ELMA», разработанной компанией «EleWise», а также ряда других приложений для бизнес-среды.

Предлагаемый перевод содержит только избранные содержательные главы оригинальной спецификации. Тем не менее, нумерация глав, параграфов и пунктов сохранена в соответствии с оригиналом, в которой можно найти недостающие части текста.

Настоящий документ создан компанией «EleWise» и предназначен для внутреннего использования, а также для общего ознакомления всех желающих с принципами построения моделей бизнес-процессов на графическом языке BPMN.

Ссылки

Сайт системы управления бизнес-процессами «ELMA»
elma.elewise.ru

Сайт компании «EleWise»
www.elewise.ru

Сайт BPMN с оригинальными спецификациями от «OMG» (на англ. языке)
bpmn.org

СОДЕРЖАНИЕ

Глава 7. И предыдущие....3
Глава 8. Диаграммы бизнес-процессов....3
§ 8.1. Перечень основных графических элементов диаграмм бизнес-процессов (BPD)....3
§ 8.2. Полный перечень диаграмм бизнес-процессов....6
§ 8.3. Использование текста, цвета и линий в моделировании диаграмм....15
§ 8.4. Правила соединения элементов потока....15
§§ 8.4.1. Правила соединения потоков операций....15
§§ 8.4.2. Правила соединения потоков сообщений....16
§ 8.5. Атрибуты диаграмм бизнес-процессов....17
§ 8.6. Процессы (Processes)....17
§§ 8.6.1. Атрибуты процесса....18
Глава 9. Графические элементы диаграмм бизнес-процессов (Business Process Diagram Graphical Objects)....20
§ 9.1. Общие атрибуты графических объектов....20
§ 9.2. Общие атрибуты элементов потока....20
§ 9.3. События (Events)....20
§§ 9.3.1. Общие атрибуты событий....20
§§ 9.3.2. Стартовое событие (Start)....20
§§ 9.3.3. Конечное событие (End)....25
§§ 9.3.4. Промежуточное событие (Intermediate Event)....30
§ 9.4. Действия (Activities)....37
§§ 9.4.1. Общие атрибуты действия....37
§§ 9.4.2. Подпроцесс (Sub-Process)....41
§§ 9.4.3. Задача (Task)....50
§ 9.5. Шлюз (Gateway)....57
§§ 9.5.1. Общие свойства шлюзов....58
§§ 9.5.2. Эксклюзивные шлюзы (ИЛИ) (Exclusive Gateways (XOR))....59
§§ 9.5.3. Неэксклюзивный шлюз (ИЛИ) (Inclusive Gateways (OR))....68
§§ 9.5.4. Комплексный шлюз (Complex Gateway)....72
§§ 9.5.5. Параллельный шлюз (И) (Parallel Gateway(AND))....75
§ 9.6. Зоны Ответственности: пулы и дорожки (Swimlanes: Pools and Lanes)....77
§§ 9.6.1. Общие атрибуты зон ответственности....78
§§ 9.6.2. Пул (Pool)....78
§§ 9.6.3. Дорожка (Lane)....80
§ 9.7. Артефакты (Artifacts)....81
§§ 9.7.1. Общие понятия артефактов....81
§§ 9.7.2. Объект данных (Data Object)....82
§§ 9.7.3. Текстовая аннотация (Text Annotation)....84
§§ 9.7.4. Группа (Group)....85
Глава 10. Объекты соединения диаграммы бизнес-процесса....86
§ 10.1. Графические элементы соединения (Graphical Connecting Objects)....86
§§ 10.1.1. Общие атрибуты элементов соединения....87
§§ 10.1.2. Поток операций (Sequence Flow)....87
§§ 10.1.3. Поток сообщений....90
§§ 10.1.4. Ассоциация....92
§ 10.2. Типы потоков операций (Sequence Flow Mechanisms)....93
§§ 10.2.1. Стандартный поток операций (Normal Flow)....94
§§ 10.2.2. Поток исключений (Exception Flow)....117
§§ 10.2.3. Произвольный процесс (Ad Hoc)....118
§ 10.3. Компенсирующая ассоциация....119

Глава 7. И предыдущие

В оригинальной спецификации языка BPMN главы с первой по седьмую являются вводными. По этой причине данные главы в настоящем переводе опущены. Обратитесь к оригинальной спецификации за дополнительной информацией и полным текстом следующих глав.
Глава 1. Ограничения спецификации.
Глава 2. Согласованность моделей с правилами языка BPMN.
Глава 3. Нормативные документы.
Глава 4. Термины и определения.
Глава 5. Символы.
Глава 6. Дополнительная информация.
Глава 7. Обзор.

Глава 8. Диаграммы бизнес-процессов

Данная глава содержит краткое описание графических элементов BPMN и отношений между ними.

Главная цель разработки BPMN – создание простой и доступной нотации для бизнес-аналитиков. Также необходимо отметить, что существуют потенциально противоречащие требования о том, что BPMN предоставляет возможность изображать сложные бизнес-процессы и находить соответствующие им элементы языка исполнения моделирования бизнес-процессов (BPM). Для того, чтобы разобраться в том, каким образом BPMN удовлетворяет вышеизложенным требованиям, необходимо обратить внимание на классификацию графических элементов BPMN.

Первая группа содержит набор основных графических элементов BPMN, удовлетворяющих требованиям простой графической нотации (simple notation). Входящие в данную группу графические элементы определяют наглядность и восприятие BPMN. Большинство бизнес-процессов моделируются с использованием элементов только этой группы. Вторая группа содержит полный перечень элементов BPMN, включающий также основные элементы, что позволяет удовлетворять требованиям комплексной графической нотации (powerful notation) и управлять более сложными ситуациями моделирования. Графические элементы нотации поддерживаются атрибутами для хранения метаинформации объектов, необходимой для установления соответствия с языком исполнения или для других целей моделирования бизнес-процессов.

§ 8.1. Перечень основных графических элементов диаграмм бизнес-процессов (BPD)

Важно отметить, что одной из причин создания BPMN явилась необходимость построения простого механизма для проектирования как простых, так и сложных моделей бизнес-процессов. Для удовлетворения двух этих противоречащих требований был применен подход систематизации графических элементов нотации по категориям. Результатом явился небольшой перечень категорий нотаций, позволивший людям, работающим с диаграммами BPMN, без труда распознавать основные типы элементов и осуществлять корректное чтение схем. Основные категории элементов допускают внутренние вариации, а также добавление информации для удовлетворения требований сложности без внесения значительных изменений в общую структуру диаграммы для легкости её понимания. Существуют четыре основные категории элементов:

- Элементы потока (Flow Objects);

- Соединяющие элементы (Connecting Objects);
- Зоны ответственности (Swimlanes);
- Артефакты (Artifacts).

Элементы потока являются важнейшими графическими элементами, определяющими ход бизнес-процесса. Элементы потока, в свою очередь, делятся на:

- События (Events);
- Действия (Activities);
- Шлюзы (Gateways).

Выделяют три вида соединяющих Элемента потока, связывающихся друг с другом и с другими элементами:

- Поток операций (Sequence Flow);
- Поток сообщений (Message Flow);
- Ассоциация (Association).

Существуют два способа группировки основных элементов моделирования с помощью Зон ответственности:

- Группировка с помощью Пула (Pool);
- Группировка с помощью Дорожки (Lane).

Артефакты используются для добавления дополнительной информации о Процессе. Выделяют три типовых Артефакта, что, однако, не запрещает разработчикам моделей бизнес-процессов либо программам моделирования добавлять необходимое количество Артефактов. Для широкого круга пользователей, а также для вертикальных рынков существует возможность стандартизации более полного перечня Артефактов. Таким образом, текущий перечень Артефактов включает в себя следующие элементы:

- Объект данных (Data object);
- Группа (Gruop);
- Аннотация (Annotation).

Таблица 8.2 содержит перечень основных графических элементов моделирования, изображенных при помощи графических нотаций.

Таблица 8.1 – Основные графические элементы моделирования

Элемент	**Описание**	**Нотация**
Событие (Event)	Событие – это то, что происходит в течение бизнес-процесса и оказывает влияние на его ход. Чаще всего событие имеет причину (триггер) или воздействие (результат). Изображается в виде круга со свободным центром, предназначенным для дифференцировки внутренними маркерами различных триггеров или их результатов. Согласно влиянию Событий на ход бизнес-процесса, выделяют три типа: Стартовое событие (Start), Промежуточное событие (Intermediate) и Конечное событие (End).	
Действие (Activity)	Действие – общий термин, обозначающий работу, выполняемую исполнителем. Действия могут быть либо элементарными, либо неэлементарными (составными). Выделяют следующие виды действий, являющихся частью модели Процесса: Процесс (Process), Подпроцесс (Sub-Process) и Задача (Task). Задача и Подпроцесс изображаются в виде прямоугольника с закругленными углами. Процесс либо не имеют границ, либо находятся внутри Пула.	

Шлюз (Gateway)	Шлюзы используются для контроля расхождений и схождений потока операций. Таким образом, данный термин подразумевает ветвление, раздвоение, слияние и соединение маршрутов. Внутренние маркеры указывают тип контроля развития бизнес-процесса.	

Таблица 8.2 - Основные графические элементы моделирования диаграммы бизнесс-процесса (BPD)

Элемент	Описание	Нотация
Поток операций (Sequence Flow)	Поток операций служит для отображения того порядка, в котором организованы действия Процесса.	
Поток сообщений (Message Flow)	Поток сообщений служит для отображения обмена сообщениями между двумя участниками, готовыми эти сообщения отсылать и принимать. На диаграмме BPMN два отдельно взятых Пула представляют собой двух участников процесса (бизнес-объекты или бизнес-роли).	
Ассоциация (Association)	Ассоциация служит для установления связи между информацией и элементами потока. Текстовые объекты, а также графические объекты, не относящиеся к элементам потока, могут соотноситься с Элементами потока.	
Пул (Pool)	В BPMN Пул представляет собой Участника Процесса. Пул также может выступать в качестве Зоны ответственности или графического контейнера, отвечающего за разделение определенного набора действий, относящихся к другим Пулам, что обычно встречается в ситуациях типа «бизнес для бизнеса» (B2B).	Name
Дорожка (Lane)	Дорожка обеспечивает разделения внутреннего пространства Пула как по вертикали, так и по горизонтали. Служит для упорядочивания и категоризации действий.	Name Name Name
Объект данных (Data Object)	Объект данных относится к Артефактам, т.к. не оказывает непосредственного влияния ни на Поток операций, ни на Поток сообщений. Однако Объект данных предоставляет	Name

	информацию о том, какие действия необходимо выполнить и/или каков результат этих действий.	
Группа (блок, содержащий группу объектов, предназначенных для документирования) (Group)	Группировка объектов, не оказывающих влияния на Поток операций. Такого рода группировка может использоваться в целях составления документации или анализа. Группы также могут использоваться для идентификации операции, распределенной внутри Пула.	
Текстовая аннотация (связана с Ассоциацией) (Text Annotation)	Текстовая аннотация служит в качестве механизма, позволяющего разработчику модели бизнес-процесса вводить дополнительную информацию для тех, кто работает с BPMN диаграммами.	Descriptive Text Here

§ 8.2. Полный перечень диаграмм бизнес-процессов

Таблица 8.3 содержит более полный перечень основных графических элементов моделирования бизнес-процессов, изображенных при помощи графических нотаций.

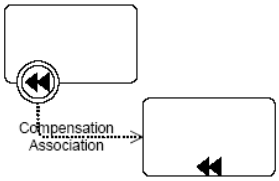
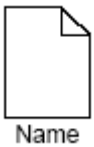
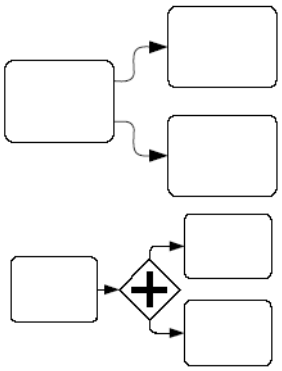
Таблица 8.3 – Развернутый перечень графических элементов моделирования диаграмм бизнес-процессов

Элемент	Описание	Нотация
Событие (Event)	Событие – это то, что происходит в течение бизнес-процесса и оказывает влияние на его ход. Обычно событие имеет причину (триггер) или воздействие (результат). Согласно влиянию Событий на ход бизнес-процесса, выделяют три типа: Стартовое событие (Start), Промежуточное событие (Intermediate) и Конечное событие (End).	Название или источник
Состав потока: Стартовое событие, Промежуточное событие, Конечное событие (Flow Dimension)		

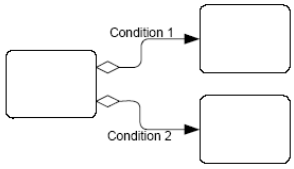
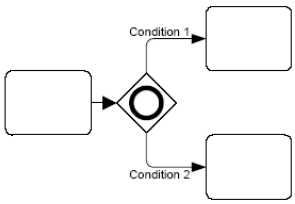
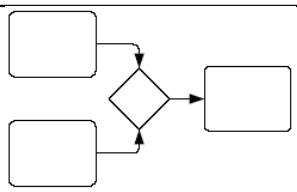
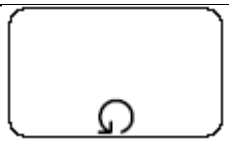
Стартовое событие (Неопределенный, Сообщение, Таймер, Правило, Связь, Множественный) (Start)	Как видно из названия, Стартовое событие указывает на то, в какой точке берет начало тот или иной бизнес-процесс.	Начало
Промежуточное событие (Неопределенный, Сообщение, Таймер, Ошибка, Отмена, Компенсация, Правило, Связь, Множественный) (Intermediate)	Как видно из названия, Промежуточное событие происходит на отрезке, ограниченном Стартовым и Конечным Событиями. Промежуточное событие оказывает влияние на ход процесса, однако, не может являться началом или завершением Процесса.	Промежуточное событие
Конечное событие (Неопределенный, Сообщение, Ошибка, Отмена, Компенсация, Связь, Завершение, Множественный) (End)	Как видно из названия, Конечное событие указывает на то, в какой точке завершается тот или иной бизнес-процесс.	Завершение
Тип (например, Сообщение, Таймер, Ошибка, Отмена, Компенсация, Правило, Связь, Множественный, Завершение) (Type Dimension)	Стартовые события и большинство Промежуточных событий имеют триггеры, определяющие причины происхождения Событий данных типов. Существует множество причин, инициирующих События. Конечные события могут определять результат, являющийся следствием окончания Потока операций.	Message Timer Error Cancel Compensation Rule Link Multiple Terminate
Задача (элементарное действие) (Task)	Задача представляет собой элементарное действие, включенное в состав Процесса. Используется в случае, если Процесс не детализируется далее в данной Модели.	
Процесс/Подпроцесс (Process/Sub-Process)	Подпроцесс представляет собой комплексное действие, включенное в состав Процесса. Такой вид действия считается составным, т.к. может быть разбит на составляющие благодаря использованию поддействий (sub-activities).	См. следующие две фигуры

Свернутый Подпроцесс (Collapsed Sub-Process)	Диаграмма не отображает детали Подпроцесса. Знак «плюс» находится в центре нижней части фигуры, символизирующей Подпроцесс, и указывает на то, что данное действие является Подпроцессом. В данном случае детали Процесса находятся на нижнем уровне.	
Развернутый Подпроцесс (Expanded Sub-Process)	Границы Подпроцесса расширены. Внутри границ просматриваются детали. Важно отметить, что Поток операций не может пересекать границ Подпроцесса.	Name
Шлюз (Gateway)	Шлюзы используются для контроля расхождений и схождений множественных Потоков операций. Таким образом, данный термин подразумевает ветвление, раздвоение, слияние и соединение маршрутов.	
Типы Шлюзов (Gateway Control Types)	Фигуры в виде ромба влияют на потоки. Выделяют следующие типы шлюзов: • Эксклюзивные ИЛИ (XOR) – исключающие условия и объединения. Могут основываться как на данных (Data-Based), так и на событиях (Event-Based). Данный тип Шлюзов, основанный на данных, отображается как с маркером «X», так и без него. • ИЛИ (OR) – включающие условия и объединение. • Комплексные (Complex) – сложные условия и ситуации (например, 3 из 5). • И (AND) – раздвоение и слияние. Шлюзы каждого из типов оказывают влияние как на входящие, так и на исходящие потоки.	Exclusive (XOR) Data-Based or Event-Based Inclusive (OR) Complex Parallel (AND)
Поток операций (Sequence Flow)	Поток операций служит для отображения того порядка, в котором выполняются действия Процесса.	См. семь следующих фигур
Стандартный поток операций (Normal Flow)	Стандартный поток операций относится к потокам, берущим начало от Стартового события и следующим по ходу выполнения действий по альтернативным либо параллельным маршрутам до Конечного события - точки, в которой	

	деятельность прекращается.	
Неконтролируемый поток операций (Uncontrolled Flow)	Неконтролируемый поток операций относится либо к потокам, на которые не воздействую никакие условия, либо к потокам, не проходящим через Шлюзы. Простейшими примерами Неконтролируемого потока операций могут послужить отдельно взятый Поток операций, объединяющий два действия, или составной Поток операций, сходящийся в действии или расходящийся от него. Для каждого Неконтролируемого потока операций возникает «Токен», проходящий от ресурсного до целевого объекта.	
Условный поток операций (Conditional Flow)	Поток операций может зависеть от условных выражений, оценивающихся согласно времени выполнения для того, чтобы определить, будет ли использоваться поток или нет. В случае, если Условный поток операций является исходящим от действия, то у основания линии изображается небольшой ромбик (см. фигуру справа). Если же Условный поток операций является исходящим от Шлюза, то никакого ромбика у основания линии не будет (см. фигуру ряда выше).	
Поток операций по умолчанию (Default Flow)	Для Эксклюзивных Условий, основанных на данных, и для Неэксклюзивных Условий, основанных на данных, предназначен лишь один тип потоков – Условный поток операций по умолчанию. Поток операций данного типа используется в том случае, если все остальные исходящие Условные потоки операций не являются верными во время выполнения действия. Для изображения таких Потоков операций используются диагональная черточка, располагающиеся у основания линии (см. фигуру справа).	
Поток исключений (Exception Flow)	Поток исключений встречается за пределами Стандартного потока операций. Основывается на Промежуточных событиях, возникающих в ходе Процесса.	Exception Flow
Поток сообщений	Поток сообщений используется для	

(Message Flow)	отображения потока сообщений между двумя участниками Процесса, готовыми принимать и отсылать сообщения. На диаграмме BPMN два отдельно взятых Пула представляют собой двух Участников Процесса.	
Компенсирующая ассоциация (Compensation Association)	Компенсирующая ассоциация происходит за рамками Стандартного потока операций. Основой такого рода Ассоциации служит Промежуточное событие «Отмена», инициируемое ошибкой, совершенной в ходе операции, либо Событие «Компенсация». Целью Компенсирующей ассоциации является компенсирующее действие.	
Объект данных (Data Object)	Объект данных относится к Артефактам, т.к. не оказывает непосредственного влияния ни на Поток операций, ни на Поток сообщений. Однако Объект данных предоставляет информацию о том, какие действия необходимо выполнить и/или каков результат этих действий.	
Раздвоение (И-Разделение) (Fork (And-Split)	Термин «раздвоение» служит в BPMN для обозначения разделения на два или более параллельных маршрутов (данное явление также называется «И-Разделение»). Раздвоение происходит в том случае, если предпочтение отдается параллельному выполнению действий, нежели последовательному. Существуют два типа Раздвоения: Множественный исходящий поток операций (см. Фигуру справа вверху) и Параллельный (И) Шлюз (см. фигуру справа ниже). Множественный исходящий поток операций (Multiple Outgoing Sequence Flow) представляет собой Неконтролируемый поток операций, являющийся предпочтительным в большинстве ситуаций. Параллельный (И) Шлюз (Parallel (AND) Gateway) используется реже, обычно – в сочетании с другими видами Шлюзов.	

Соединение (И-Соединение) (Join (AND-Join))	Термин «Соединение» используется в BPMN для обозначения слияния двух или более параллельных маршрутов в один (данное явление также называется И-Соединение или синхронизация). Параллельный (И) Шлюз предназначается для объединения множественных потоков.	
Условие, Точка ветвления (ИЛИ-Разделение) (Decision, Branching Point; (OR-Split))	Условиями являются Шлюзы, находящиеся в рамках бизнес-процесса, где контрольный поток движется по одному или нескольким альтернативным маршрутам.	См. фигуры в следующих пяти ячейках
Эксклюзивный шлюз (Exclusive)	Эксклюзивный шлюз (XOR) ограничивает поток таким образом, что лишь один маршрут из нескольких альтернативных может быть выбран во время выполнения. Существуют два типа Эксклюзивных шлюзов: Эксклюзивные шлюзы, основанные на данных, и Эксклюзивные шлюзы, основанные на событиях.	Data-Based or Event-Based
Эксклюзивный шлюз, основанный на данных (Data-Based)	Данный вид Шлюзов представляет собой Точку ветвления, в которой выбор маршрута основывается на условных выражениях, хранимых в исходящем Потоке операций. Может быть выбран лишь один из альтернативных маршрутов.	Condition 1 Default
Эксклюзивный шлюз, основанный на событиях (Event-Based)	Данный вид Шлюзов представляет собой Точку ветвления, в которой выбор маршрута основывается на Событии, происходящим в данной точке в ходе Процесса. Отдельно взятое Событие, обычно являющееся получением Сообщения, определяет выбор необходимого маршрута. Также могут использоваться другие типы Событий, например, Событие «Таймер». Может быть выбран лишь один маршрут. Существуют два пути получения сообщения: через Задачи типа «Получение» (см. фигуру справа вверху) и Промежуточные события «Сообщение» (см. фигуру справа ниже).	[Type: Receive] [Type: Receive]

Неэксклюзивный шлюз (Inclusive)	Данный вид Шлюзов представляет собой Точку ветвления, в которой выбор маршрута основывается на условных выражениях, хранимых в исходящем Потоке операций. В некотором смысле, данный вид Шлюзов представляет собой группировку связанных между собой независимых Бинарных Шлюзов (Да/Нет). Т.к. любой из маршрутов является независимым, то могут использоваться любые сочетания маршрутов (от нуля до максимального числа сочетаний маршрутов). Однако при построении диаграмм необходимо учитывать то, что должен использоваться хотя бы один маршрут. Для проверки того, что выбран по меньшей мере один маршрут, может быть использовано Условие по умолчанию. Существую два вида данного типа Шлюзов. Первый тип использует совокупность условных Потоков операций. На схеме выделяется при помощи небольших ромбиков (см. фигуру справа вверху). Второй тип использует ИЛИ Шлюзы, обычно – в сочетании с другими видами Шлюзов (см. фигуру справа ниже).	 
Слияние (ИЛИ Соединение) (Merging (OR-Join))	Термин «слияние» используется в BPMN для обозначения исключающего объединения двух или более маршрутов в один (данное явление также называется ИЛИ-Соединение). Шлюз «Слияние» (XOR) предназначается для отображения слияния множественного потока. В случае, если все Входящие потоки операций являются альтернативными, то нет необходимости использовать Шлюз. Это означает, что Неконтролируемый поток операций оказывает такое же влияние на ход Процесса.	
Цикличность (Looping)	В BPMN существуют два механизма, обеспечивающих цикличность внутри Процесса.	См. фигуры в следующих двух ячейках
Цикличность действия (Activity Looping)	Атрибуты Задач и Подпроцессов указывают на то, будут ли они повторяться или будут выполнены единожды. Существуют два вида циклов: Стандартный и Многоэкземплярный. Небольшой маркер цикличности отображается в центре у нижней границы	

	фигуры действия.	
Цикличность Потока операций (Sequence Flow Looping)	Циклы могут появляться благодаря присоединению Потока операций к «противоположному» объекту. Объект является противоположным в том случае, если данный объект имеет исходящий Поток операций, ведущий к ряду других Потоков операций, последний из которых является Входящим потоком операций для исходного объекта.	
Многоэкземплярный (Multiple Instances)	Атрибуты Задач и Подпроцессов указывают на то, будут ли они повторяться или будут выполнены единожды. Небольшой параллельный маркер отображается в центре у нижней границы фигуры действия.	
Перерыв в Процессе (что-то, способное приостановить Процесс и не подающееся управлению) (Process Break)	Перерыв в Процессе представляет собой участок Процесса, указывающий то, на каком его отрезке произойдет ожидаемая задержка. Для отображения действительного хода Процесса используется Промежуточное действие (см. фигуру справа вверху). Также необходимо добавить, что Артефакт Перерыва в Процессе по желанию разработчика модели или программы моделирования может быть отнесен к Событиям, что подчеркнет расположение задержки внутри потока.	Announce Issues for Vote Voting Response Received Increment Tally
Транзакция (Transaction)	Транзакция представляет собой Подпроцесс, поддерживаемый особым протоколом, гарантирующим то, что между всеми участвующими сторонами заключено соглашение о том, что действие следует либо завершить, либо отклонить. Графические элементы действия указывают на то, является ли действие Транзакцией. Граница, выполненная двойной линией, указывает на то, что данный Подпроцесс является сделкой.	
Вложенный/Встроенный Подпроцесс (Nested/Embedded Sub-Process (Inline Block))	Вложенный (или встроенный) Подпроцесс представляет собой действие, имеющее тот же набор данных, как и родительский Процесс. Данный тип Подпроцесса является противоположным независимому Подпроцессу, который может быть использован заново, и на	Для отображения вложенного Подпроцесса маркер не предусмотрен

	который ссылается родительский Процесс. При использовании Потока операций данные должны передаваться основному, а не вложенному Подпроцессу.	
Группа (блок, содержащий группу объектов, предназначенных для документирования) (Group)	Группировка объектов, не оказывающих влияния на Поток операций. Такого рода группировка может использоваться в целях составления документации или анализа. Группы также могут использоваться для идентификации операции, распределенной внутри Пула.	
Соединитель страниц (Off-Page Connector)	Предназначен в основном для печати. Данный графический элемент указывает, на каком этапе Потока операций произошло потеря страницы, а затем возобновляется на другой странице. В качестве соединителя страниц может использоваться Промежуточное событие «Связь».	
Ассоциация (Association)	Ассоциация служит для установления связи между информацией и элементами потока. Текстовые объекты, а также графические объекты, не относящиеся к Элементам потока, могут соотноситься с Элементами потока.	
Текстовая аннотация (связана с Ассоциацией) (Text Annotation)	Текстовая аннотация служит в качестве механизма, позволяющего разработчику модели бизнес-процесса вводить дополнительную информацию для тех, кто читает BPMN диаграммы.	Descriptive Text Here
Пул (Pool)	В BPMN Пул представляет собой Участника Процесса. Пул также может выступать в качестве Зоны ответственности или графического контейнера, отвечающего за разделение определенного набора действий, относящихся к другим Пулам, что обычно встречается в ситуациях типа «бизнес для бизнеса» (B2B).	Name
Дорожки (Lanes)	Объект данных относится к Артефактам, т.к. не оказывает непосредственного влияния ни на Поток операций, ни на Поток сообщений.	Name Name Name Name

§ 8.3. Использование текста, цвета и линий в моделировании диаграмм

Текстовые аннотации объектов используются разработчиком модели с целью отобразить дополнительную информацию о Процессе или атрибутах объекта, находящегося внутри Процесса.

- Элементы потока и сам поток МОГУТ носить текстовые метки (labels) (например, имя потока и/или названия других его атрибутов). Текстовые метки могут помещаться как внутри фигуры, так и над и под ней. Месторасположение текстовых меток, а также их направление может быть любым в зависимости от задумки разработчика модели или программы моделирования.
- Заливка графического элемента МОЖЕТ БЫТЬ как белого цвета, так и прозрачной.
    - Графическая нотация допускает использование какого-либо другого цвета заливки для удовлетворения требований разработчика модели или программы моделирования (например, выделение значения атрибута объекта).
- Элементы потока и маркеры МОГУТ БЫТЬ того размера, который удовлетворяет требованиям разработчика модели или программы моделирования
- Линии, используемые в моделировании диаграмм, МОГУТ БЫТЬ черными.
    - Графическая нотация допускает использование других цветов линий для удовлетворения требований разработчика модели или программы моделирования (например, выделение значения атрибута объекта).
    - Графическая нотация допускает использование разного дизайна линий для удовлетворения требований разработчика модели или программы моделирования (например, выделение значения атрибута объекта), однако, при условии, что выбранный дизайн линий НЕ ДОЛЖЕН противоречить ни одному из вариантов языка BPMN. Таким образом, дизайн линий, используемых для изображения Потока операций, Потока сообщений, а также Ассоциаций, изменяться НЕ ДОЛЖЕН.

§ 8.4. Правила соединения элементов потока

Входящий Поток операций может быть присоединен к любой точкой Элемента потока (слева, справа, сверху, снизу). Подобно ему, Исходящий поток операций может брать начало из любой точки Элемента потока (слева, справа, сверху, снизу). Поток сообщений обладает теми же свойствами, что и Поток операций. Язык BPMN может подстраиваться под предъявляемые требования, однако, для соединения Элементов потока разработчикам моделей настоятельно рекомендуется использовать имеющийся опыт, что облегчит понимание создаваемых диаграмм и сделает ход изображаемого бизнес-процесса прозрачным и доступным для понимания. Это особенно важно в том случае, если в диаграмме присутствуют такие графические элементы, как Поток операций или Поток сообщений. В данном случае оптимальным вариантом является выбор направления Потока операций, располагающегося либо слева направо, либо сверху вниз, а затем и выбор направления Потока сообщений, который необходимо расположить под углом в 90° по отношению к уже выбранному Потоку операций. При выполнении всех вышеперечисленных требований создаются удобные для работы диаграммы.

§§ 8.4.1. Правила соединения потоков операций

Таблица 8.4 содержит изображения Элементов потока, используемые языком BPMN, а также указывает, каким образом данные графические элементы соединяются друг с другом посредством Потока операций. Символ ↗ обозначает, что графический элемент,

изображенный в одной из строк таблицы, может соединяться с графическим элементом, изображенным в соответствующей колонке. В таблице не указывается количество входящих и исходящих соединений графического элемента, зависящее от различных конфигураций. Следующая глава содержит детальную информацию о правилах соединения каждого отдельно взятого графического элемента. *Обратите внимание, что в случае, если Подпроцесс занимает всю протяженность диаграммы, то графические элементы, находящиеся внутри данного Подпроцесса, не могут быть соединены с графическими элементами, находящимися за его пределами. Подобно этому, Поток операций не может пересекать границ Пула.*

Таблица 8.4 – Правила Соединения Потока Операций

От/К		Name	Name			
		↗	↗	↗	↗	↗
Name		↗	↗	↗	↗	↗
Name		↗	↗	↗	↗	↗
		↗	↗	↗	↗	↗
		↗	↗	↗	↗	↗

Примечание: В таблице отображены лишь графические элементы, имеющие Входящие или Исходящие потоки операций. Такие объекты, как Пул, Дорожка, Объект данных и Текстовая аннотация, в таблице не содержатся.

§§ 8.4.2. Правила соединения потоков сообщений

Таблица 8.5 содержит изображения объектов моделирования BPMN, а также указывает, каким образом данные объекты соединяются друг с другом посредством Потока сообщений. Символ ⇗ обозначает, что графический элемент, изображенный в одной из строк таблицы, может соединяться с графическим элементом, изображенным в соответствующей колонке. В таблице не указывается количество входящих и исходящих соединений графического элемента, зависящее от различных конфигураций. Следующая глава содержит детальную информацию о правилах соединения каждого отдельно взятого графического элемента. Обратите внимание, что Поток сообщений не соединяется с объектами, расположенными в пределах одной и той же Дорожки Участника.

Таблица 8.5 – Правила Соединения Потока сообщений

От/К		Name (Pool)	Name	Name		
Name (Pool)						
Name						
Name						

Примечание: В таблице отображены лишь графические элементы, имеющие входящие или исходящие Потоки сообщений. Такие объекты, как Дорожка, Шлюз, Объект данных и Текстовая аннотация, в таблице не содержатся.

§ 8.5. Атрибуты диаграмм бизнес-процессов

Параграф не включен в настоящий перевод. Обратитесь к оригинальной спецификации.

§ 8.6. Процессы (Processes)

Процесс представляет собой действие, производимое внутри компании или организации. В спецификации BPMN Процесс изображается в качестве диаграммы, состоящей из Элементов потока и представляющей собой совокупность Действий и Шлюзов, влияющих на порядок выполнения этих Действий. В сущности, понятие процесса является иерархическим. Процессы могут быть отнесены к любому уровню: от корпоративных Процессов до Процессов, выполняющихся отдельно взятым работником. Для достижения общей цели в бизнесе Процессы нижнего уровня могут быть объединены.

Обратите внимание, что спецификация BPMN дает довольно точное определение Процессу. Что касается бизнес-процесса, то в BPMN он имеет более общее определение, а именно: набор действий, выполняемых внутри организации либо между организациями. Таким образом, на диаграмме бизнес-процесс может содержать более одного Процесса. В свою очередь, каждый отдельно взятый Процесс может включать собственные Подпроцессы; в таком случае он находится внутри Пула. Отдельные Процессы могут быть

независимыми в контексте Потока операций, однако, к таким Процессам возможно присоединение Потока сообщений.

§§ 8.6.1. Атрибуты процесса

Данная таблица содержит информацию об атрибутах Процесса:

Таблица 8.7 – Атрибуты Процесса

Атрибут	Описание
Id: Object	Уникальный идентификатор, позволяющий различать графические элементы на диаграмме.
Name: String	Текстовое описание графического элемента.
ProcessType (None | Private | Abstract | Collaboration) None : String	Содержит информацию о том, какому языку нижнего уровня соответствует Пул. По умолчанию Процесс имеет тип «None». Процесс «Private» МОЖЕТ соответствовать выполняемому процессу BPEL4WS. Процесс типа «Abstract» также называется открытым интерфейсом Процесса (либо других веб-сервисов) и МОЖЕТ соответствовать абстрактному процессу BPEL4WS. Процесс типа «Collaboration» имеет две дорожки, представляющие собой бизнес-роли (например, покупатель и продавец) и передающие отношения между ними. Такие дорожки могут соответствовать языкам типа ebXML или WSChoreography. Однако данный документ не содержит описание такого рода соответствий.
Status (None | Ready | Active | Cancelled | Aborting | Aborted | Completing | Completed) None : String	Статус Процесса определяется в ходе выполнения данного Процесса при помощи механизмов процесса. Данный атрибут может использоваться внутри **выражений назначений**.
GraphicalElements (0-n) : Object	Определяет все типы графических элементов (например, Событие, Действие, Шлюз, Артефакт), включенных в бизнес-процесс.
Assignments (0-n) : Assignment	В Процесс МОЖЕТ БЫТЬ добавлено одно или более выражений назначений. Assignment СЛЕДУЕТ рассматривать как атрибут, определенный DesignTime.
Properties (0-n) : Property	В Процесс МОГУТ БЫТЬ добавлены Properties, определенные разработчиком модели. По отношению к Процессу, такого рода Properties являются локальными. Все Задачи, объекты Подпроцессов, а также вставленные Подпроцессы ДОЛЖНЫ иметь доступ к Properties. Полное изображаемое имя таких Properties выглядит следующим образом: “<process name>.<property name>” (например, “Add Customer.Customer Name”). В случае, если один Процесс встроен в другой, то полное изображаемое

	имя СЛЕДУЕТ размещать перед названием родительского Процесса, независимо от того, как много родительских Процессов содержится до Процесса верхнего уровня.
AdHoc False : Boolean	Логически атрибут, значение которого по умолчанию - «False». Это помогает определить, имеет ли данный Процесс тип «AdHoc» или нет. Действия, включенные в Процесс такого типа, не контролируются и не упорядочиваются; их выполнение осуществляет исполнитель действий. В случае, если значение атрибута равно «True», то маркер AdHoc ДОЛЖЕН БЫТЬ помещен в центр нижней части фигуры Процесса или Подпроцесса внутри данного Процесса типа AdHoc.
[AdHoc = True only] **AdHocOrdering** (0-1) (Sequential | Parallel) Parallel : String	В случае, если Процесс имеет тип AdHoc (атрибут AdHoc со значением «True»), то ДОЛЖЕН БЫТЬ добавлен атрибут AdHocOrdering. Данный атрибут помогает определить, могут ли действия, включенные в Процесс, выполняться параллельно или должны быть последовательными. Значение, определенное по умолчанию, равно «Parallel», а значение «Sequential» является ограничением, которое может стать необходимостью, благодаря общим ресурсам.
[AdHoc = True only] **AdHocCompletionCondition** (0-1) : Expression	В случае, если Процесс имеет тип AdHoc (атрибут AdHoc со значением «True»), то ДОЛЖЕН БЫТЬ добавлен атрибут AdHocCompletionCondition. Данные атрибут помогает определить, при каких условиях завершится Процесс.
SuppressJoinFailure False : Boolean	Добавляется для соответствия BPEL4WS. Данный атрибут помогает определить, будет ли подавляться ошибка BPEL4WS joinFailure для всех действий процесса BPEL4WS.
EnableInstanceCompensation False : Boolean	Добавляется для соответствия BPEL4WS. Данный атрибут помогает определить, будет ли производиться Компенсация (Compensation) после того, как Процесс был удачно завершен.
Categories (0-n) : String	Разработчик модели МОЖЕТ добавлять один или более атрибутов Categories, используемых для отчетов или анализа.
Documentation (0-1) : String	Разработчик модели МОЖЕТ добавлять текстовые документы, содержащие информацию о Процессе.

Глава 9. Графические элементы диаграмм бизнес-процессов (Business Process Diagram Graphical Objects)

Данная глава содержит подробное описание графических элементов BPMN и принципы построения диаграмм бизнес-процессов.

§ 9.1. Общие атрибуты графических объектов

Параграф не включен в настоящий перевод. Обратитесь к оригинальной спецификации.

§ 9.2. Общие атрибуты элементов потока

Параграф не включен в настоящий перевод. Обратитесь к оригинальной спецификации.

§ 9.3. События (Events)

Событие – это то, что происходит в ходе бизнес-процесса и оказывает влияние на его течение. Обычно событие имеет причину (триггер) или воздействие (результат). Термин «событие» является достаточно широким для того, чтобы охватить множество понятий бизнес-процесса. Начало действия, окончание действия, смена статуса документа, поступившее сообщение и т.д., - все это можно отнести к Событиям. Однако спецификация BPMN устанавливает ограничение на употребление термина «событие» для обозначения событий, влияющих на ход действия Процесса или его синхронизацию. BPMN выделяет три основных типа Событий: Стартовое событие (Start), Промежуточное событие (Intermediate) и Конечное событие (End).

Стартовые и большинство Промежуточных событий имеют триггеры, определяющие причину возникновения того или иного события. Существует множество причин, инициирующих События. Конечные события могут определять результат, являющийся следствием окончания Потока операций. Существует множество видов результатов.

Все типы Событий изображаются в виде кругов. Как было упомянуто выше, благодаря разнообразному дизайну линий различаются три типа Событий потока. Любое Событие имеет свободный центр, предназначенный для добавления определенных иконок, установленных спецификацией BPMN и разработчиком модели, что помогает определять триггер или результат данного События.

§§ 9.3.1. Общие атрибуты событий

Пункт не включен в настоящий перевод. Обратитесь к оригинальной спецификации.

§§ 9.3.2. Стартовое событие (Start)

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.

Стартовое событие изображается в виде круга со свободным центром, как и Промежуточный и Конечный типы Событий. Свободный центр предназначен для добавления маркеров, определяющих вид События.

- Стартовое событие представляет собой круг, выполненный тонкой линией (см. фигуру 9.1).
- Текст, цвет, размер, а также линии, используемые для изображения Стартового события, ДОЛЖНЫ соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм». Однако следует учитывать следующее исключение:

- Толщина линии ДОЛЖНА БЫТЬ тонкой настолько, чтобы без труда можно было отличить Стартовое событие от Промежуточного или Конечного.

Фигура 9.1 – Стартовое событие

На протяжении изложения материала данного документа авторами будет обсуждаться то, в каком направлении движется Поток операций внутри Процесса. Для облегчения усвоения материала обратимся к понятию «токен» (Token). Токен пересекает Поток операций и проходит в Процессе через Элементы потока, что позволяет наблюдать за ходом Процесса, отслеживая маршрут (маршруты) Токена внутри данного Процесса. Токен обладает уникальной особенностью, называемой Набором идентификаторов Токена. Набор идентификаторов Токена позволяет различать многочисленные Токены, существующие благодаря совпадающим экземплярам Процесса, а также разделению самого Токена для параллельной обработки данных внутри экземпляра отдельно взятого Процесса. Такое параллельное разделение Токена создает нижний уровень Набора идентификаторов Токена, но определяется Токен благодаря всем уровням Набора идентификаторов данного Токена.

Стартовое событие создает Токен, который в итоге должен быть использован к Конечному событию (Конечное событие является скрытым, если не отображается на диаграмме). Маршруты Токена должны прослеживаться на схеме Потока операций, Шлюзов, а также Действий, включенных в Процесс. На схеме НЕ ДОЛЖНО БЫТЬ каких-либо скрытых потоков, включенных в Стандартный поток операций, т.е. на схеме всегда должен присутствовать либо Поток операций, либо определенный графический элемент (например, Промежуточное событие), показывающие все возможные маршруты Токенов. Примером скрытого потока может служить ситуация, когда Токен подходит к Шлюзу, однако, ни один из Шлюзов не является верным. Токен подходит к концу Процесса, имеющему определенные графические нотации. Также Токены могут быть направлены через исключение, управляющее Промежуточными событиями, выступающими в роли вынужденного завершения действия. Обратите внимание, что Токен не пересекает Потока сообщений до тех пор, пока в потоке передаются именно сообщения.

Стартовое событие имеет следующие особенности:

- Стартовое событие является НЕОБЯЗАТЕЛЬНЫМ. Процессы верхнего уровня, а также Развернутые Подпроцессы МОГУТ начинаться со Стартового события, однако, данное условие не является обязательным.

Примечание: Диаграмма бизнес-процесса может включать Процессы разных уровней (например, диаграмма может включать Развернутые Подпроцессы). Использование Стартового и Конечного событий зависит от уровней диаграммы.

- В случае, если Процесс является комплексным и/или исходные условия – не очевидны, то РЕКОМЕНДУЕТСЯ использовать Стартовое событие.
- В случае, если диаграмма содержит Конечное событие, то ДОЛЖНО БЫТЬ по меньшей мере одно Стартовое событие.
- В случае, если диаграмма содержит Стартовое событие, то в ней НЕ ДОЛЖНО содержаться каких-либо других графических Элементов потока, не имеющих входящих Потоков операций. Все остальные элементы диаграммы ДОЛЖНЫ соединяться хотя бы с одним Потоком операций.

  - Исключением являются Действия, относящиеся к Компенсации, т.е. имеющие маркер «Компенсация». Действие Компенсация НЕ ДОЛЖНО соединяться с каким-либо Входящим потоком операций, даже в случае, если в Процессе содержится Стартовое событие.
  - Исключением является также и Промежуточное событие, которое МОЖЕТ существовать без соединения с каким-либо Входящим потоком операций (например, в ситуации, когда данное Промежуточное событие соединено с границами какого-либо Действия).
- В случае, если диаграмма не содержит Стартового события, то Элементы потока, не имеющие Входящих потоков операций, должны создаваться, когда создается Процесс. Существует мнение о том, что лишь диаграмма может содержать лишь одно скрытое Стартовое событие, из чего следует то, что все исходные Элементы потока будут запущены в одно и то же время.
  - Исключением являются действия, относящиеся к Компенсации, т.е. имеющие маркер «Компенсация». Действия Компенсации не относятся к Стандартному Потоку операций и НЕ ДОЛЖНЫ изображаться во время отображения Процесса.
- Существуют множественные Стартовые события, относящиеся к Процессу того или иного уровня.
  - Каждое отдельно взятое Стартовое событие представляет собой независимое событие. Именно поэтому экземпляр Процесса СЛЕДУЕТ создавать в момент запуска Стартового события.

Примечание: Ход Процесса может стать более сложным для понимания в случае, если используется множественное Стартовое событие. РЕКОМЕНДУЕТСЯ как можно реже использовать такой вид Стартового события, т.к. при работе с диаграммой у пользователей, кроме разработчика модели, могут возникнуть трудности в понимании её значения.

При запуске Стартового события для каждого Исходящего потока операций создаются Токены, идущие от запущенного События. Набор идентификаторов, определенный для каждого из Токенов, устанавливается таким образом, чтобы был очевиден факт того, что данные Токены берут начало из одного параллельного Раздвоения (И-Разделения), а также для указания номеров Токенов в данной группе. Токены начнут свое движение, для которого запуск какого-либо другого Стартового события значения не имеет.

В случае, если для запуска Процесса необходимо наличие более одного События (например, Процесс открывается двумя сообщениями), то Стартовые события должны подразумевать одно и тоже действие внутри данного Процесса. Атрибуты действия указывают на то, когда данное действие может быть начато. Если атрибуты действия указывают на то, что действие зависит от входов, в таком случае все Стартовые события необходимо запустить до начала самого Процесса. К тому же для использования различных Стартовых событий в одном и том же экземпляре Процесса требуется корреляционный механизм. Вероятнее всего корреляция будет управляться с помощью атрибутов Событий, однако, этот вопрос является открытым и адресован более поздней версии данного документа.

Механизмы запуска стартовых событий

Существует множество способов инициирования бизнес-процесса. Триггеры Стартовых событий создаются с целью сделать прозрачным общий механизм активизации отдельно

взятого Процесса. BPMN выделяет шесть видов Стартовых Событий: Неопределенный, Сообщение, Таймер, Правило, Связь и Множественный.

Таблица 9.4 содержит информацию о видах Триггеров и графических маркерах, используемых для изображения каждого вида:

Таблица 9.4 – Типы Стартовых Событий

Триггер	Описание	Маркер
Неопределенный (None)	Данный тип Стартового события не указывается разработчиком модели. Такой маркер используется и для изображения Подпроцесса, ход которого инициирован родительским Процессом.	
Сообщение (Message)	Сообщение поступает от участника Процесса и запускает Стартовое событие.	
Таймер (Timer)	Определенный временной интервал или цикл (например, еженедельно по понедельникам в 9.00 утра), инициирующий возникновение Процесса.	
Правило (Rule)	Стартовое событие данного типа возникает в случае, если условия для правил, типа «Повышение индекса S&P 500 с момента открытия составляет более 10%» или «Температура свыше 300С», становятся верными.	
Связь (Link)	Представляет собой механизм для соединения Конечного события (результа) одного Процесса со Стартовым событием (триггером) другого Процесса. Обычно такими Процессами являются два Подпроцесса внутри одного родительского Процесса.	
Множественный (Multiple)	Подразумевает наличие множества способов инициирования Процесса. Однако для запуска Процесса требуется лишь один из таких способов. Атрибуты Стартового события указывают, какой из видов Триггеров был использован.	

Атрибуты стартового события

Таблица 9.5 содержит информацию об атрибутах Стартового события, продолжающих список общих атрибутов События:

Таблица 9.5 - Атрибуты Стартового События

Атрибут	Описание
Trigger (None | Message | Timer| Rule | Link | Multiple) None : String	Атрибут Стартового события, указывающий тип Триггера, используемого для Стартового события. Значение данного атрибута по умолчанию равно «None». Следующие восемь секций таблицы содержат информацию об атрибутах, необходимых для каждого типа Триггеров. Перечень Триггеров МОЖЕТ быть расширен в связи с добавление новых их типов. Новые типы Триггеров МОГУТ приобретать новые маркеры, вписывающиеся в границы События. Новые маркеры могут добавляться разработчиком программы моделирования.
[Message Trigger only] **Message** : Message	В случае, если Триггер имеет тип «Message», то ДОЛЖЕН БЫТЬ определен атрибут «Message». Дополнительную информацию см. в разделе B, 11.4, «Message», стр. 269 оригинала спецификации.
[Message Trigger only] **Implementation** (Web Service | Other | Unspecified) : Web Service	Данный атрибут Стартового события указывает на то, какая технология будет использоваться для получения сообщения. Технологией по умолчанию является Веб-сервис (Web service).
[Timer Trigger only] **TimeDate (0-1)** : Date	В случае, если Триггер имеет тип «Timer», то МОЖЕТ БЫТЬ введено значение TimeDate. Если значение TimeDate не введено, то ДОЛЖНО БЫТЬ введено значение TimeCycle (см. описание атрибута ниже).
[Timer Trigger only] **TimeCycle (0-1)** : String	В случае, если Триггер имеет тип «Timer», то МОЖЕТ БЫТЬ введено значение TimeCycle. Если значение TimeCycle не введено, то ДОЛЖНО БЫТЬ введено значение TimeDate (см. описание атрибута выше).
[Rule Trigger only] **RuleName** : Rule	В случае, если Триггер имеет тип «Rule», то ДОЛЖЕН БЫТЬ определен атрибут «Rule». Дополнительную информацию см. в разделе B, 11.9, «Rule», стр. 271 оригинала спецификации.
[Link Trigger only] **LinkId** : String	В случае, если Триггер имеет тип «Link», то ДОЛЖНО БЫТЬ введено значение «LinkId».
[Link Trigger only] **ProcessRef** : Process	В случае, если Триггер имеет тип «Link», то ДОЛЖНО БЫТЬ введено значение «ProcessRef». Указанный Процесс МОЖЕТ являться Процессом События «Связь».
[Multiple Trigger only] **Triggers** (2-n) : Trigger	В случае, если Триггер имеет тип «Multiple», то ДОЛЖНО БЫТЬ определено по меньшей мере два Триггера. Для каждого Триггера ДОЛЖНЫ БЫТЬ указаны соответствующие данные (см. выше). Триггер НЕ ДОЛЖЕН иметь тип «None» или «Multiple».

Соединение потока операций

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут являться объектами Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций».

- Стартовое событие ДОЛЖНО БЫТЬ объектом Потока сообщений. Оно НЕ ДОЛЖНО БЫТЬ соединено с каким-либо Входящим потоком операций.
    - Исключением является ситуация, когда Стартовое событие используется в Развернутом Подпроцессе и соединяется с границами данного Подпроцесса. В таком случае Поток операций, относящийся к Процессу более верхнего уровня, МОЖЕТ БЫТЬ соединен со Стартовым событием, а не с границей Подпроцесса.
- Стартовое событие ДОЛЖНО являться источником Потока операций.
- Множественный поток операций МОЖЕТ начинаться со Стартового события. Необходимо создание нового параллельного маршрута для каждого Потока операций, имеющего Стартовое событие.
- Атрибут Условие для всех исходящих Потоков операций ДОЛЖЕН иметь значение «None».
- В случае, если Процесс не содержит Стартового события, то все Элементы потока, не имеющие Входящих потоков операций, ДОЛЖНЫ стать началом независимого параллельного маршрута.

Каждый маршрут обладает уникальным Токеном, пересекающим Поток операций.

Соединение потока сообщений

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут являться объектами Потока сообщений, обратитесь к пункту 8.4.2 «Правила Соединения Потоков Сообщений».

Примечание: Все Потоки сообщений должны соединять два независимых Пула. Потоки сообщений могут быть соединены как с границами Пулов, так и с Элементами потока, расположенными внутри Пулов, однако, не могут соединять два объекта, расположенные внутри одного Пула.

- Стартовое событие МОЖЕТ БЫТЬ целью Потока сообщений и иметь как несколько входящих Потоков сообщений, так и ни одного. Любой Поток сообщений, являющийся входящим для Стартового события, представляет собой механизм создания (триггер) Процесса. Для запуска нового Процесса требуется лишь один из триггеров.
    - При наличии входящего Потока сообщений атрибут триггера Стартового события ДОЛЖЕН иметь значение «Сообщение» либо «Составной элемент».
        - При наличии нескольких входящих Потоков сообщений атрибут триггера Стартового события ДОЛЖЕН иметь значение «Составной элемент».
- Стартовое событие НЕ ДОЛЖНО являться источником Потока сообщений; оно также НЕ ДОЛЖНО соединяться с какими-либо исходящими Потоком сообщений.

§§ 9.3.3. Конечное событие (End)

Как видно из названия, Конечное событие указывает на то, в какой точке завершается тот или иной бизнес-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 изображается в виде круга со свободным центром, как и Стартовый и Промежуточный типы Событий. Свободный центр предназначен для добавления маркеров, определяющих вид События.

- Конечное событие представляет собой круг, выполненный одиночной, жирной, черной линией (см. фигуру 9.2).
- Текст, цвет, размер, а также линии, используемые для изображения Конечного события, должны соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм». Однако следует учитывать следующее исключение:
    - Толщина линии должна быть жирной настолько, чтобы без труда можно было отличить Конечное событие от Стартового и Промежуточного.

Фигура 9.2 – Конечное событие

Конечное событие является точкой, где заканчивается Токен, созданный на том же уровне Процесса при запуске Стартового события. В случае, если в параллельном Потоке операций запланировано Конечное событие, то Токены будут заканчиваться в момент соединения с Конечным событием. Все Токены, созданные внутри Процесса, должны завершиться в точке Конечного события до полного окончания данного Процесса. В других случаях, когда Процесс является Подпроцессом, он может быть завершен заранее благодаря исключению из потока Промежуточного события. В подобных случаях Токены будут завершены до Промежуточного события, соединенного с Подпроцессом.

Конечное событие имеет следующие особенности:

- Процессы МОГУТ содержать несколько Конечных событий.
- Значок Конечного события является ОПЦИОНАЛЬНЫМ. Процесс определенного уровня - Процесс верхнего уровня или Развернутый Подпроцесс - МОЖЕТ удовлетворять данному требованию, однако, это не является необходимым условием.
    - В случае, если диаграмма содержит Стартовое событие, то ДОЛЖНО БЫТЬ по меньшей мере одно Конечное событие.
    - В случае, если диаграмма содержит Конечное событие, то в ней НЕ ДОЛЖНО содержаться каких-либо других графических Элементов потока, не имеющих исходящих Потоков операций. Все остальные элементы диаграммы ДОЛЖНЫ соединяться хотя бы с одним Потоком операций.
        - Исключением являются действия, относящиеся к Компенсации, т.е. имеющие маркер «Компенсация». Действие Компенсация НЕ ДОЛЖНО соединяться с каким-либо исходящим Потоком операций, даже в случае, если в Процессе содержится Конечное событие.
    - В случае, если диаграмма не содержит Конечного события, то Элементы потока, не имеющие исходящих Потоков операций (не являющиеся ресурсами для Потоков операций), указывают на окончание какого-либо маршрута, содержащегося в Процессе. Однако никакой маршрут НЕ ДОЛЖЕН БЫТЬ завершен до тех пор, пока к концу не придут остальные параллельные маршруты.
        - Исключением являются действия, относящиеся к Компенсации, т.е. имеющие маркер «Компенсация». Действие Компенсация не

является элементом Стандартного потока операций и НЕ МОЖЕТ указывать на завершение Процесса.

Примечание: Диаграмма бизнес-процесса может содержать Процессы разных уровней (например, Развернутый Подпроцесс). Использование Стартового и Конечного событий является свободным для Процессов любого уровня, содержащихся в диаграмме.

В случае, если Конечное событие не включено в состав Процесса, то Токен, содержащийся в данном Процессе, соединяется с Элементами потока, чей маршрут ведет к завершению Процесса, и завершается в момент окончания обработки данных (окончания маршрута). Это происходит таким образом, словно Токен имел своей целью достичь Конечного события. В момент, когда все Токены данного экземпляра Процесса заканчиваются, Процесс считается завершенным.

Результаты конечных событий

Разработчик модели BPMN может определить результат использования Конечного события в Процессе, а именно – Результат Конечного события.

Таблица 9.6 содержит информацию о видах Результатов и графических маркерах, используемых для изображения каждого вида:

Таблица 9.6 – Типы Конечных Событий

Результат	Описание	Маркер
Неопределенный (None)	На модели не отображается данный тип События. Такой маркер используется и для изображения завершения Подпроцесса, что является причиной возвращения потока к родительскому Процессу.	
Сообщение (Message)	Данный тип Конечного события подразумевает то, что в момент завершения Процесса другому его участнику отправляется сообщение.	
Ошибка (Error)	Данный тип Конечного события означает, что должна быть создана проименованная ошибка. Данная ошибка будет поймана Промежуточным событием из числа событий Процесса.	
Отмена (Cancel)	Данный тип Конечного события используется внутри транзакционного Подпроцесса. Указывает на необходимость отмены данной Транзакции и запускает Промежуточное событие типа «Отмена», присоединенное к Подпроцессу. Также конечное событие данного типа указывает на необходимость отсылки сообщения «Отмена протокола Транзакции» всем сторонам, имеющим отношение к Транзакции.	

Компенсация (Compensation)	Данный тип Конечного события указывает на необходимость Компенсации. Идентификатор Компенсации запускает Промежуточное событие при ходе Процесса в обратном направлении.	
Связь (Link)	Представляет собой механизм для соединения Конечного события (результата) одного Процесса со Стартовым событием (триггером) другого. Обычно такими Процессами являются два Подпроцесса внутри одного родительского Процесса. Токен, подходящий к Конечному событию «Связь», сразу переходит к соответствующим Стартовому или Промежуточному событиям другого Процесса.	
Завершение (Terminate)	Данный тип Конечного события указывает на необходимость немедленного завершения всех действий, включенных в Процесс. Это касается всех Многоэкземплярных действий Процесса. Процесс завершается без возможности компенсации и управления Событиями.	
Множественный (Multiple)	Подразумевает наличие множества способов завершения Процесса. Используются все способы (например, может быть отправлено несколько сообщений). Атрибуты Конечного события указывают на то, какой вид Результата был использован.	

Атрибуты конечного события

Таблица 9.7 содержит информацию об атрибутах Конечного события, продолжающих список общих атрибутов События:

Таблица 9.7 – Атрибуты Конечного События

Атрибут	Описание
Result (None | Message | Error | Cancel | Compensation | Link | Terminate | Multiple) None : String	Атрибут, указывающий тип результата, ожидаемого при завершении Процесса. Значение данного атрибута по умолчанию равно «None». Данный атрибут со значением «Cancel» НЕ ДОЛЖЕН применяться до тех пор, пока в Процессе, являющемся Транзакцией, присутствует Событие. Перечень Результатов МОЖЕТ БЫТЬ расширен в связи с добавление новых их типов. Новые типы Результатов МОГУТ приобретать новые маркеры, подходящие под ограничение События. Новые маркеры могут добавляться разработчиком программы моделирования.

[Message Result only] **Message** : Message	В случае, если Результат имеет тип «Message», то ДОЛЖЕН БЫТЬ определен атрибут «Message». Дополнительную информацию см. в части B, 11.4, «Message», стр. 269.
[Message Result only] **Implementation** (Web Service | Other | Unspecified) Web Service : String	Данный атрибут указывает на то, какая технология будет использоваться для отправки сообщения. Технологией по умолчанию является Веб-сервис (Web service).
[Error Result only] **ErrorCode** : String	В случае, если Результат имеет тип «Error», то ДОЛЖНО БЫТЬ введено ErrorCode.
[Compensation Result only] **Activity** : Object	В случае, если Результат имеет тип «Compensation», то ДОЛЖЕН БЫТЬ использован объект Действия, предназначенный для компенсации.
[Link Result only] **LinkId** : String	В случае, если Результат имеет тип «Link», то ДОЛЖНО БЫТЬ введено LinkId.
[Link Result only] **ProcessRef** : Process	В случае, если Результат имеет тип «Link», то ДОЛЖНО БЫТЬ введено ProcessRef. Указанный Процесс МОЖЕТ являться Процессом События «Связь».
[Multiple Result only] **Results (2-n)** : Result	В случае, Результат имеет тип «Multiple», то ДОЛЖНО БЫТЬ по меньшей мере два Результата. Для каждого Результата ДОЛЖНЫ БЫТЬ указаны соответствующие данные (см. выше). Результат НЕ ДОЛЖЕН иметь тип «None», «Terminate» или «Multiple» .

Соединение потока операций

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций».

- Конечное событие ДОЛЖНО являться целью Потока операций.
- Конечное событие МОЖЕТ БЫТЬ соединено с несколькими Входящими потоками операций.

Поток операций МОЖЕТ идти как по альтернативным, так и по параллельным маршрутам. Для удобства создания модели каждый маршрут МОЖЕТ соединяться с различными графическими элементами Конечного события. Конечное событие является точкой (Sink), куда прибывают Токены. Все Токены, созданные при запуске Стартового события, должны в конечном итоге подойти к точке Конечного события. Процесс не может быть завершен до тех пор, пока все Токены не будут употреблены.

- Конечное событие НЕ МОЖЕТ являться источником Потока операций, следовательно, оно НЕ ДОЛЖНО БЫТЬ соединено ни с одним Потоком операций.
    - Исключением является ситуация, когда Конечное событие используется в Развернутом Подпроцессе и соединяется с границами данного Подпроцесса. В таком случае Поток операций, относящийся к Процессу более верхнего

уровня, МОЖЕТ БЫТЬ соединен с Конечным событием, а не с границей Подпроцесса.

Соединение потока сообщений

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока сообщений, обратитесь к пункту 8.4.2 «Правила Соединения Потоков Сообщений».

Примечание: Все Потоки сообщений должны соединять два независимых Пула. Потоки сообщений могут быть соединены как с границами Пулов, так и с Элементами потока, расположенными внутри Пулов, однако, не могут соединять два объекта, расположенные внутри одного Пула.

- Конечное событие НЕ ДОЛЖНО являться целью Потока сообщений, следовательно, оно не может быть соединено ни с одним входящим Потоком сообщений.
- Конечное событие МОЖЕТ являться источником Потока сообщений, следовательно, оно может быть соединено с одним или более исходящими Потоками сообщений.

§§ 9.3.4. Промежуточное событие (Intermediate Event)

Промежуточное событие происходит на отрезке, ограниченном Стартовым и Конечным событиями, после того, как был Процесс запущен. Промежуточное событие влияет на ход Процесса, однако, не может являться началом или завершением Процесса. Промежуточное событие может быть задействовано для того, чтобы:

- Указать отрезок Процесса, где ожидаются или отправляются сообщения,
- Указать отрезок Процесса, на котором ожидаются задержки,
- Нарушить ход Стандартного потока операций при помощи исключений,
- Указать на необходимость дополнительной работы для компенсации.

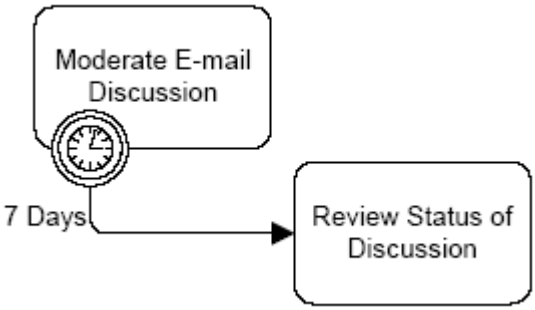
Промежуточное событие изображается в виде круга со свободным центром, как и Стартовый и Конечный типы Событий. Свободный центр предназначен для добавления маркеров, определяющих вид События.

- Промежуточное событие представляет собой круг, выполненный двойной, тонкой, черной линией (см. фигуру 9.3).
- Текст, цвет, размер, а также линии, используемые для изображения Промежуточного события, ДОЛЖНЫ соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм». Однако следует учитывать следующее исключение:
    - Линия ДОЛЖНА быть двойной для того, чтобы без труда можно было отличить Промежуточное событие от Стартового или Конечного.

Фигура 9.3 – Промежуточное событие

Некоторые разработчики моделей используют Промежуточное событие для отображения исключений или компенсации, входящих в Процесс. Для этого Промежуточное событие изображают на границе необходимой Задачи Подпроцесса (как свернутого, так и развернутого). Фигура 9.4. является примером того, каким образом Промежуточное событие может быть соединено с Задачей. Промежуточное событие может соединяться с любой точкой границы действия, а Исходящий Поток операций может быть направлен в

любом направлении. Однако для большей прозрачности диаграмм разработчикам рекомендуется выбирать определенную точку соединения Промежуточного события и действия. Например, в случае, если диаграмма располагается горизонтально, то Промежуточное событие может быть соединено с нижней границей действия, а Поток операций направлен строго вниз, а затем направо. Если же диаграмма располагается вертикально, то Промежуточное событие может быть соединено как с правой, так и с левой границами действия, а Поток операций направлен либо вправо, либо влево, а затем вниз.



Фигура 9.4 – Промежуточное событие, соединенное с Задачей

Триггеры промежуточного события

BPMN выделяет восемь типов Промежуточных событий: Сообщение, Таймер, Ошибка, Компенсация, Отмена, Правило, Связь, Множественный. Данные типы Промежуточных событий указывают различные способы прерывания или торможения хода Процесса после его начала. Разные типы Промежуточных событий изображаются при помощи разных иконок (маркеров), помещенных в свободный центр круга отдельно взятого Промежуточного события для определения его вида.

Таблица 9.8 содержит информацию о типах Триггеров и графических маркерах, используемых для изображения каждого из них:

Таблица 9.8 – Типы Промежуточных Событий

Триггер	Описание	Маркер
Не определенный	Является верным лишь для Промежуточных событий, представляющих собой основную часть Процесса. На модели не отображается данный тип События. Используется в методиках моделирования, где События являются индикаторами изменения статуса Процесса.	
Сообщение	Сообщение поступает от участника Процесса и запускает Событие, что приводит к возобновлению хода Процесса в случае, если он был приостановлен в ожидании получения сообщения. Это также меняет ход действия для исключения. В Стандартном потоке операций Промежуточное событие типа «Сообщение» используется для отправки сообщения другому участнику Процесса. В случае, если оно используется для исключения, то Поток операций из Стандартного становится Исключающим.	

Таймер	Определенный временной интервал или цикл (например, еженедельно по понедельникам в 9 00 утра), инициирующий возникновение События. В случае, если данный тип Промежуточного события используется внутри основного потока, то ход Процесса приостанавливается. В случае, если Таймер используется для исключения, то Поток операций из Стандартного становится Исключающим.	
Ошибка	Используется для управления ошибками: необходим для генерации (выброса) и обработки исключительных ситуаций. Генерирует (выбрасывает) ошибку в случае, если Событие является частью Стандартного потока операций, а при присоединении к границам действия обрабатывает названную ошибку или любую другую ошибку, имя которой не определено.	
Отмена	Используется в транзакционном Подпроцессе. Данный тип Промежуточного события ДОЛЖЕН быть присоединен к границе Подпроцесса. Он также ДОЛЖЕН быть запущен в случае, если достигнуто Конечное событие «Отмена» данного транзакционного Подпроцесса, а также в случае, если сообщение транзакционного протокола «Отмена» было получено в момент, когда транзакция ещё выполнялась.	
Компенсация	Используется для произведения компенсации: необходим для установки и выполнения компенсации. Инициирует компенсацию в случае, если Событие является частью Стандартного потока операций. При присоединении к границам действия реагирует на название компенсационного запроса.	
Правило	Используется для исключений, т.е. в случае, когда Правило становится верным для данной ситуации. Правило является выражением, определяющим данные Процесса.	
Связь	Представляет собой механизм для соединения Конечного события (результата) одного Процесса с Промежуточным событием (триггером) другого. Парные Промежуточные события также могут использоваться в качестве Объектов перехода (“Go To” objects) внутри Процесса.	
Множественный	Подразумевает наличие множества способов запуска События, однако, использован будет лишь один из них. Атрибуты Промежуточного события указывают на то, какие ещё виды Триггеров будут	

	использованы.	

Атрибуты промежуточного события

Таблица 9.9 содержит информацию об атрибутах Промежуточного события, продолжающих список общих атрибутов События:

Таблица 9.9 – Атрибуты Промежуточного События

Атрибут	Описание
Trigger (None | Message | Timer | Error | Cancel | Link | Compensation | Rule | Multiple) Message : String	Атрибут, указывающий тип триггера, используемого для данного Промежуточного события. По умолчанию имеет значение «Message». Триггеры со значениями «None» и «Link» НЕ ДОЛЖНЫ использоваться в случае, если Промежуточное событие соединяется с границами Действия. Триггеры со значениями «Multiple», «Rule», а также «Cancel» НЕ ДОЛЖНЫ использоваться в случае, если данное Промежуточное событие является частью Стандартного потока операций. Триггер со значением «Cancel» НЕ ДОЛЖЕН использоваться в том случае, если Промежуточное событие соединяется с границами Транзакции, или если данное Событие не включено в транзакционный Процесс. Перечень Триггеров МОЖЕТ БЫТЬ расширен в связи с добавление новых их типов. Внутри События разработчиком программы моделирования могут добавляться новые Триггеры.
Target (0-1) : Object	Данный атрибут МОЖЕТ БЫТЬ использован для Промежуточного события. Он также ДОЖЕН являться Действием (Подпроцессом или Задачей). Это означает, что Промежуточное событие присоединено к границам Действия и используется в качестве признака исключения или компенсации данного Действия.
[Message Trigger only] **Message** : Message	В случае, если Триггер имеет тип «Message», то ДОЛЖЕН БЫТЬ определен атрибут «Message». Дополнительную информацию см. в части B, 11.4, «Message», стр. 269.
[Message Trigger only] **Implementation** (Web Service | Other | Unspecified) : Web Service	Данный атрибут указывает на то, какая технология будет использоваться для отправки и получения сообщений. Технологией по умолчанию является Веб-сервис (Web service).
[Timer Trigger only] **Timedate** (0-1) : Date	В случае, если Триггер имеет тип «Timer», то МОЖЕТ БЫТЬ введено значение TimeDate. Если значение TimeDate не введено, то ДОЛЖНО БЫТЬ введено значение TimeCycle (см. описание атрибута ниже).

[Timer Trigger only] **TimeCycle** (0-1) : String	В случае, если Триггер имеет тип «Timer», то МОЖЕТ БЫТЬ введено значение TimeCycle. Если значение TimeCycle не введено, то ДОЛЖНО БЫТЬ введено значение TimeDate (см. описание атрибута выше).
[Error Trigger only] **ErrorCode** : String	В случае, Промежуточное событие включено в Стандартный поток операций, то: Если Триггер имеет тип «Error», то ДОЛЖНО БЫТЬ введено ErrorCode, генерирующее ошибку. В случае, если Промежуточное событие присоединено к границам Действия, то: Если Триггер имеет тип «Error», то МОЖЕТ БЫТЬ введен код ошибки. В данном случае Промежуточное событие будет направлено на обработку ошибки. Если код ошибки введен не был, то Промежуточное событие будет инициировано любой ошибкой. Если же код ошибки был введен, то лишь указанная ошибка, соответствующая введенному коду, будет инициировать возникновение Промежуточного события.
[Compensation Trigger only] **Activity** : Object	В случае, Промежуточное событие включено в Стандартный поток операций, то: Если Триггер имеет тип «Compensation», то ДОЛЖЕН БЫТЬ использован объект Действия, нуждающийся в компенсации, что и сгенерирует компенсацию. В случае, если Промежуточное событие присоединено к границам Действия, то: Данное Промежуточное событие обрабатывает компенсацию. Нет необходимости в добавлении какой-либо другой дополнительной информации. Объект Действия, к которому присоединено данное Промежуточное событие, укажет идентификатор, необходимый для установления соответствия между компенсацией и событием, сгенерировавшем данную компенсацию.
[Rule Trigger only] **RuleName** : Rule	В случае, если Триггер имеет тип «Rule», то ДОЛЖНО БЫТЬ введено значение «Rule». Дополнительную информацию см. в части B, 11.9, «Rule», стр. 271.
[Link Trigger only] **LinkId** : String	В случае, если Триггер имеет тип «Link», то ДОЛЖНО БЫТЬ введено значение «LinkId».
[Link Trigger only] **ProcessRef** : Process	В случае, если Триггер имеет тип «Link», то ДОЛЖНО БЫТЬ введено значение «ProcessRef». Указанный Процесс МОЖЕТ являться Процессом События «Связь».
[Multiple Trigger only] **Triggers (2-n)** : Trigger	В случае, если Триггер имеет тип «Multiple», то каждый из Триггеров, включенных в список, ДОЛЖЕН иметь соответствующие данные, указанные выше. Триггер НЕ ДОЛЖЕН иметь значение «None» или «Multiple» .

Соединение с границами действия

Существуют следующие особенности присоединения Промежуточного события к границе какого-либо действия согласно следующим правила:

- Одно или несколько Промежуточных событий МОГУТ быть присоединены непосредственно к границе действия.
    - Для того, чтобы быть присоединенным к границе действия, Промежуточное событие ДОЛЖНО иметь один из следующих типов: Сообщение, Таймер, Ошибка, Отмена, Компенсация, Правило или Множественный.
        - Промежуточное событие «Отмена» МОЖЕТ быть присоединено к границам Подпроцесса, однако, лишь в случае, если атрибут транзакции данного Подпроцесса имеет значение «TRUE».

Соединение потока операций

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций».

- Промежуточные события следующих типов МОГУТ быть присоединены к границе действия: Сообщение, Таймер, Исключение, Отмена (в случае, если Подпроцесс является транзакцией), Компенсация, Правило, а также Множественный. Следовательно, Промежуточные события типа «Неопределенный» или «Связь» присоединены к границе действия быть НЕ МОГУТ.
    - В случае, если Промежуточное событие присоединено к границе действия, то:
        - Промежуточное событие НЕ МОЖЕТ являться целью Потока операций, следовательно, не может иметь Входящих потоков операций.
        - Промежуточное событие ДОЛЖНО являться источником Потока операций, следовательно, может быть присоединено к одному Исходящему потоку операций.
            - Исключение: Промежуточное событие типа «Компенсация» НЕ ДОЛЖНО иметь каких-либо Исходящих потоков операций (однако МОЖЕТ БЫТЬ соединено с исходящей Ассоциацией).
- Промежуточные события следующих типов МОГУТ быть включены в Стандартный поток операций: Неопределенный, Сообщение, Таймер, Исключение, Компенсация, Правило, а также Связь. Следовательно, Промежуточные события типа «Отмена» или «Множественный» включены в Стандартный поток операций быть НЕ МОГУТ.
    - В случае, если Промежуточное событие включено в Стандартный поток операций, то:
        - Промежуточные события следующих типов ДОЛЖНЫ быть целью Потока операций: Неопределенный, Ошибка, а также Компенсация. В данном случае Промежуточное событие ДОЛЖНО БЫТЬ соединено лишь с одним Входящим потоком операций.
        - Промежуточные события следующих типов МОГУТ быть целью Потока операций: Сообщение, Таймер, Правило, а также Связь. В данном случае Промежуточное событие МОЖЕТ БЫТЬ соединено лишь с одним Входящим потоком операций.

Примечание: Данные типы Промежуточных событий всегда смогут принять необходимые триггеры, пока Процесс, в который они включены, не является завершенным.

- Промежуточное событие ДОЛЖНО являться источником Потока операций и соединяться с одним Исходящим потоком операций.
    - Исключением является ситуация, когда источником Потока операций является Промежуточное событие типа «Связь» (см. ниже). В данном случае нет необходимости соединять Промежуточное событие с Исходящим потоком операций.
- Промежуточное событие типа «Связь» НЕ ДОЛЖНО являться и целью, и источником Потока операций одновременно. Однако это возможно в ситуациях, когда данное Промежуточное событие является частью Эксклюзивного шлюза, основанного на событиях.

Правила использования Промежуточного события типа «Связь» в качестве Соединителя страниц или Объекта перехода:

- Промежуточного события типа «Связь» МОЖЕТ являться либо целью (Target Link – «Связь - цель»), либо источником Потока операций (Source Link – «Источник связи»), однако, НЕ ДОЛЖНО быть целью и источником Потока операций одновременно.
    - В случае, если в Процесс включено Промежуточное событие «Связь», являющееся источником Потока операций, то ДОЛЖНО быть и соответствующее ему Промежуточное событие «Связь», представляющее собой источник Потока операций (данные типы Target Link и Source Link обладают одинаковым атрибутом LinkId). Примечание: Промежуточное событие «Связь», являющееся источником, не следует использовать в качестве соединяющего звена с другим Процессом в рамках одного Пула. Для этих целей используется Конечное событие.
        - В Процессе МОГУТ содержаться несколько Промежуточных событий «Источник связи» для одного Промежуточного События «Связь - цель».
        - Процесс НЕ МОЖЕТ содержать несколько Промежуточных событий «Связь - цель» для одного Промежуточного События «Источник связи».
    - Промежуточное событие «Связь - цель» МОЖЕТ быть использовано без соответствующего Промежуточного События «Источник связи». Это означает, что Промежуточное Событие «Источник связи» (Конечное событие) используется в другом Процессе, однако, в рамках того же самого Пула.

Соединение потока сообщений

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока сообщений, обратитесь к пункту 8.4.2 «Правила Соединения Потоков Сообщений».

Примечание: Все Потоки сообщений должны соединять два разных Пула. Они могут быть присоединены к границам Пулов или Элементам потока, находящимся внутри

Пулов. Потоки сообщений не могут соединять два графических элемента внутри одного Пула.

- Промежуточное событие типа «Сообщение» МОЖЕТ являться целью Потока сообщений и иметь один Входящий поток сообщений.
- Промежуточное событие НЕ ДОЛЖНО являться источником Потока сообщений и иметь исходящий Поток сообщений.

§ 9.4. Действия (Activities)

Действие представляет собой деятельность, выполняемую внутри бизнес-процесса. Действие может быть как элементарным, так и неэлементарным (составным). Диаграмма бизнес-процесса может содержать все существующие виды действий: Процесс, Подпроцесс, а также Задачу. Для изображения Процесса используется несколько графических элементов, а не один. Следующая часть данного документа содержит подробное описание графических элементов Подпроцесса и Задачи. Для более полной информации обратитесь к разделу 8.6 «Процесс».

§§ 9.4.1. Общие атрибуты действия

Таблица 9.10 содержит информацию об атрибутах, относящихся как к Подпроцессу, так и к Задаче и продолжающих список атрибутов Элементов потока (см. таблицу 9.3). Обратите внимание, что таблицы 9.11 и 9.12 содержат информацию о дополнительных атрибутах, которые должны быть включены в данный перечень.

Атрибут	Описание
ActivityType (Task | Sub-Process) Task : String	Значение ActivityType ДОЛЖНО иметь либо тип «Task», либо «Sub-Process».
Status (None | Ready | Active | Cancelled | Aborting | Aborted | Completing | Completed) None : String	Статус действия определяется тогда, когда данное действие выполнять сервером Процессов. Статус действия может использоваться Выражениями Назначения (Assignment Expressions).
Properties (0-n) : Property	К Действию МОГУТ БЫТЬ добавлены Свойства (Properties), определенные инструментом моделирования. По отношению к Действию, такого рода Свойства являются локальными и могут быть использованы лишь в ходе действия. Полное изображаемое имя таких Свойств выглядит следующим образом: “<process name>.<activity name>.<property name>” (например, “ Add Customer.Review Credit.Status ”). Дополнительную информацию см. в части B.11.7, «Property», стр. 270.
InputSets (0-n) : Input	Атрибуты InputSets определяют требования к данным, которые должны быть на входе действия. МОЖЕТ БЫТЬ определено 0 или более атрибутов InputSets. Каждый InputSets является достаточным для того, чтобы было выполнено действие (при условии, что оно было инициировано раньше соответствующим импульсом, полученным от Входящего потока операций).

[Input: for InputSets only] **Inputs** (1-n) : Artifact	Для каждого атрибута InputSet ДОЛЖЕН БЫТЬ определен один или несколько атрибутов Inputs. Атрибут Input является Артефактом, обычно представляющим собой Объект Документа. Обратите внимание, что Артефакты также МОГУТ БЫТЬ отображены на диаграмме и соединены с Действием посредством Ассоциации, однако, их изображение не является обязательным.
OutputSets (0-n) : Output	Атрибуты OutputSets определяют требования к данным, которые должны быть на выходе действия. МОЖЕТ БЫТЬ определено 0 или более атрибутов OutputSets. При завершении действия может быть создан лишь один из атрибутов OutputSets. От выполнения действия зависит то, какие атрибуты OutputSets будут созданы. Однако атрибут IORules МОЖЕТ указывать на отношения между OutputSet и InputSet, запускающим действие.
[Output: for OutputSets only] **Outputs** (1-n) : Artifact	Для каждого атрибута OutputSet МОГУТ БЫТЬ определены один или несколько атрибутов Output. Атрибут Output является Артефактом, обычно представляющим собой Объект Документа. Обратите внимание, что Артефакты также МОГУТ БЫТЬ отображены на диаграмме и соединены с Действием посредством Ассоциации, однако, их изображение не является обязательным.
IORules (0-n) : Expression	Является выражением, определяющим отношения между InputSet и OutputSet. Это означает, что действие инициируется с указанным атрибутом InputSet, после чего выход действия ДОЛЖЕН создать указанный OutputSet. МОГУТ БЫТЬ определены несколько атрибутов IORules, либо не определено ни одного из них.
StartQuantity 1 : Integer	Значение данного атрибута по умолчанию равно «1», однако, оно НЕ ДОЛЖНО БЫТЬ меньше «1». Данный атрибут определяет то, какое количество Токенов должно поступить из Потока операций для того, чтобы действие могло начаться.
LoopType (None | Standard | MultiInstance) None : String	LoopType также является атрибутом. Значение данного атрибута по умолчанию равно «None», однако может изменяться на «Standard» или «MultiInstance». В таких случаях маркер Цикличности ДОЛЖЕН БЫТЬ помещен в центр нижней части формы Действия (см. фигуры 9.8 и 9.13). Задача типа «Receive», имеющая свой атрибут «Instantiate» и значение «True», ДОЛЖНА иметь типы Цикличности «Standard» или «MultiInstance».

Атрибуты стандартного цикла

Действие, называемое Стандартным циклом (Standard Loop), имеет логическое выражение, значение которого определяется после каждого цикла. В случае, если выражение имеет значение «True», то цикл продолжается. Существуют две разновидности циклов, являющиеся эквивалентами программных понятий «пока» (while) и «до» (until). Таким образом, цикл «пока» определяет значение выражения **до того**, как действие начнет выполняться; это означает то, что фактически действие может быть не выполнено. Цикл «до» определяет значение выражения **после того**, как действие было выполнено; это означает, что данное действие будет выполнено по меньшей мере один раз.

Таблица 9.11 содержит информацию об атрибутах Стандартной цикличности (где атрибут LoopType меняется на «Standard»), продолжающих список общих атрибутов Действия (см. таблицу 9.10):

Таблица 9.11 – Атрибуты Стандартного Цикла

Атрибут	Описание
LoopCondition : Expression	Стандартный Цикл ДОЛЖЕН содержать логическое выражение, значение которого необходимо определить, а также расчет по времени того, когда ДОЛЖНО БЫТЬ определено значение данного выражения. Дополнительную информацию об атрибутах выражения см. в части B.11.3, «Expression», стр. 269.
Lo**opCounter** : Integer	Атрибут LoopCounter используется во время выполнения действия для того, чтобы определить количество циклов. Автоматически обновляется сервером Процессов. Данный атрибут ДОЛЖЕН БЫТЬ расширен в самом начале цикла. Программа моделирования может использовать атрибут LoopCounter в выражении LoopCondition.
LoopMaximum (0-1) : Integer	Атрибут Maximum является опциональным атрибутом, позволяющим без труда ограничивать количество циклов. Данный атрибут ДОЛЖЕН БЫТЬ добавлен к выражению, определенному в LoopCondition.
TestTime (Before | After) After : String	Данные атрибуты, значения которых определяются до начала действия, являются эквивалентами программной функции «пока». Атрибуты, значения которых определяются после окончания действия, являются эквивалентами программной функции «до».

Атрибуты многоэкземплярного цикла

Многоэкземплярный цикл является эквивалентом программного понятия «для каждого». Выражение цикличности такого рода представляет собой числовое выражение, значение которого определяется лишь однажды – до начала выполнения действия. Результатом

определения значения данного выражения является целое число, указывающее на то, сколько раз будет повторяться действие.

Многоэкземплярная цикличность также имеет две разновидности, указывающие на то, каким образом будут выполнены экземпляры циклов: последовательно или параллельно.

Таблица 9.12 содержит информацию об атрибутах Многоэкземплярной цикличности (где атрибут LoopType меняется на «MultiInstance»), продолжающих список общих атрибутов Действия (см. таблицу 9.10):

Таблица 9.12 – Атрибуты Многоэкземплярного Цикла

Атрибут	Описание
MI_Condition : Expression	Многоэкземплярная цикличность ДОЛЖНА содержать числовое выражение, значение которого необходимо определить. Данное выражение ДОЛЖНО сводиться к целому числу. Дополнительную информацию об атрибутах выражения см. в части B.11.3, «Expression», стр. 269.
LoopCounter : Integer	Атрибут LoopCounter применяется лишь для последовательных Многоэкземплярных циклов и Процессов, выполняемых сервером Процессов. Значение данного атрибута автоматически обновляется сервером Процессов с целью подсчета количества циклов по мере их появления. Атрибут LoopCounter ДОЛЖЕН БЫТЬ увеличен на «1» в самом начале цикла. В отличие от Стандартного цикла, программа моделирования не использует данный атрибут для выражения MI_Condition, который, однако, может использоваться для отслеживания изменений статуса цикла.
MI_Ordering (Sequential | Parallel) Sequential : String	Данный атрибут применяется лишь для Многоэкземплярных циклов. Атрибут MI_Ordering указывает на то, будут ли экземпляры циклов выполнены последовательно или параллельно. Атрибут MI_Ordering со значением «Sequential» представляет собой чаще используемый цикл. Атрибут MI_Ordering со значением «Parallel» является эквивалентом многоэкземплярной спецификации, использующей другие нотации, например, UML Activity Diagrams. В случае, если выбрано значение Parallel, то маркер цикла, расположенный в центре нижней границы фигуры Действия (см. Фигуру 9.8, а также табл. 9.10), ДОЛЖЕН БЫТЬ заменен маркером параллельности.
[Parallel MI_Ordering only] **MI_FlowCondition** (None | One | All | Complex) All : String	Данный атрибут является равнозначным использованию Шлюза для осуществления контроля над Потоком операций, проходящим за пределами параллельных маршрутов. Атрибут MI_FlowCondition со значением «None»

	является тем же самым, что и Неконтролируемый поток операций (Поток операций, не содержащий Шлюзов), и указывает на то, что все экземпляры действия ДОЛЖНЫ создавать Токены, продолжающиеся даже после завершения существования экземпляров. Атрибут MI_FlowCondition со значением «One» является тем же самым, что и Эксклюзивный шлюз, и указывает на то, что Токены ДОЛЖНЫ продолжаться за пределами Действия после того, как завершится один из экземпляров Действия. Оставшиеся экземпляры Действия будут продолжаться, однако, дополнительные Токены НЕ ДОЛЖНЫ выходить за пределы данного Действия. Атрибут MI_FlowCondition со значением «All» является тем же самым, что и Параллельный шлюз и указывает на то, что Токены ДОЛЖНЫ продолжаться за пределами Действия после того, как все экземпляры действия завершатся. Атрибут MI_FlowCondition со значением «Complex» является тем же самым, что и Составной шлюз и определяет направление Токенов.
[Complex MI_FlowCondition only] **ComplexMI_FlowCondition** (0-1) : Expression	В случае, если атрибут MI_FlowCondition имеет значение «Complex», то ДОЛЖНО БЫТЬ введено выражение, которое МОЖЕТ указывать на данные о Процессе. Также выражение ДОЛЖНО определять, когда и каким образом многочисленные Токены будут продолжаться за пределами Действия. Дополнительную информацию об атрибутах выражения см. в части B.11.3, «Expression», стр. 269.

§§ 9.4.2. Подпроцесс (Sub-Process)

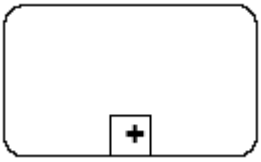
Подпроцесс представляет собой составное действие, заключающее в себе Поток операций. Графически Подпроцесс изображается в качестве элемента Потока операций Процесса. Подпроцесс также может изображаться «открытым» в случае, если необходимо показать другой Процесс (либо Встроенный, либо Независимый) внутри данного Подпроцесса. Подпроцесс, как и Задача, изображается в виде прямоугольника с закругленным углами,.

- Подпроцесс представляет собой прямоугольник с закругленным углами, выполненный одиночной, тонкой, черной линией.
- Текст, цвет, размер, а также линии, используемые для изображения Подпроцесса, ДОЛЖНЫ соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм». Однако следует учитывать следующее исключение:
    - Двойная линия прямоугольника ДОЛЖНА указывать на Подпроцесс, имеющий атрибут «IsATransaction» со значением «True».

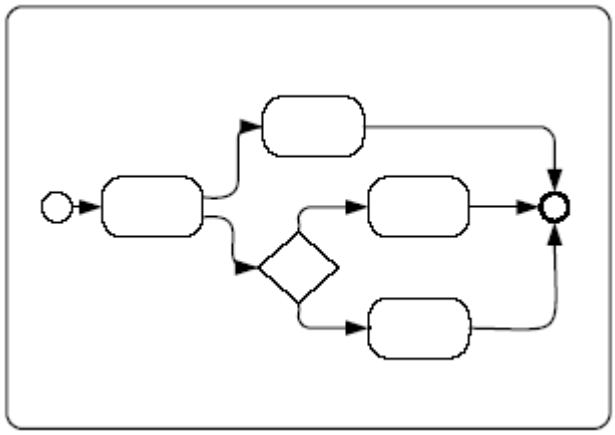
Подпроцесс может быть свернутым (Collapsed Sub-Process), при этом его детали скрыты (см. фигуру 9.5). Подпроцесс также может быть развернутым (Expanded Sub-Process), при этом его детали отображаются внутри Процесса, в котором данный Подпроцесс

содержится (см. фигуру 9.6). В случае, если Подпроцесс является свернутым, то используется маркер, позволяющий отличить Подпроцесс от Задачи.

- Маркер Подпроцесса ДОЛЖЕН изображаться в виде небольшого квадрата, расположенного в центре нижней части графического элемента и заключающего в себе знак «+».



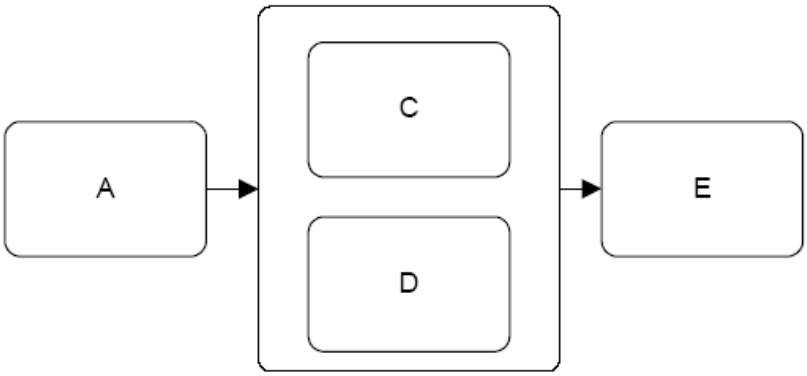
Фигура 9.5 – Свернутый Подпроцесс



Фигура 9.6 – Развернутый Подпроцесс

Развернутый Подпроцесс используется для разных целей. Он может быть использован для выравнивания структуры иерархического процесса с целью одновременного отображения деталей данного процесса. Он также может быть использован для создания соответствующего окружения для выполняемого исключения, применяемого для группы действий (дополнительную информацию см. в части 10.2.2, «Exception Flow», стр. 130). Подобным образом может выполняться и Компенсация (дополнительную информацию см. в части 10.3, «Compensation Association», стр. 133).

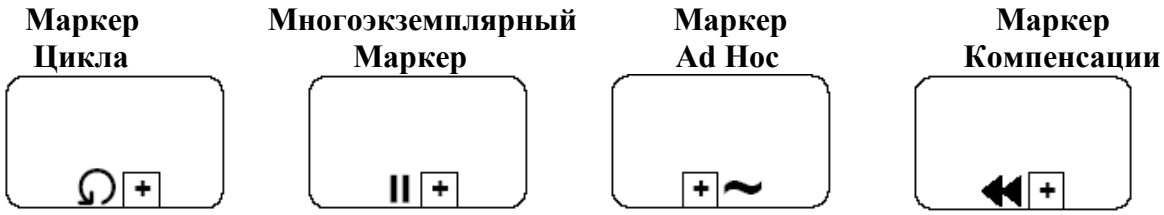
Развернутый Подпроцесс может использоваться в качестве способа компактного отображения группы параллельных действий с использованием минимума деталей. Действия «C» и «D», изображенные на фигуре 9.7, заключены в Развернутый Подпроцесс. Данные действия будут выполняться параллельно. Обратите внимание, что Развернутый Подпроцесс не содержит Стартового или Конечного событий, а также Потока операций, исходящего от данных Событий или направленного к ним. Использование Развернутого Подпроцесса в качестве «параллельных блоков» («parallel boxes») является мотивом для использования Стартового и Конечного событий как опциональных.



Фигура 9.7 – Развернутый Процесс, используемый в качестве «параллельного блока»

BPMN различает пять типов стандартных маркеров Свернутого Подпроцесса. Маркер Подпроцесса, изображенный на фигуре 9.5, может сочетаться с оставшимися четырьмя маркерами: Маркером Цикла (Loop Marker), Параллельным Маркером (Многоэкземплярным) (Parallel/Multiple Instance Marker), Маркером Компенсации (Compensation Marker) и Маркером Ad Hoc (Ad-Hoc Marker). Помимо Маркера Подпроцесса, Свернутый Подпроцесс может содержать от одного до трех вышеуказанных Маркеров. Комбинации Маркеров могут быть любыми, кроме сочетания Маркера Цикличности и Многоэкземплярного, - эти Маркеры не могут изображаться одновременно (см. фигуру 9.8).

- Маркер Цикла ДОЛЖЕН БЫТЬ выполнен в виде небольшой стрелки, острие которой загнуто в направлении, противоположном направлению самой стрелки.
    - Маркер Цикла МОЖЕТ сочетаться с любым другим Маркером Подпроцесса, кроме Многоэкземплярного.
- Многоэкземплярный Маркер ДОЛЖЕН БЫТЬ выполнен в виде двух параллельных вертикальных линий.
    - Многоэкземплярный Маркер МОЖЕТ сочетаться с любым другим Маркером Подпроцесса, кроме Маркера Цикла.
- Маркер Ad Hoc ДОЛЖЕН БЫТЬ выполнен в виде тильды.
    - Маркер Ad Hoc МОЖЕТ сочетаться с любым другим Маркером Подпроцесса.
- Маркер Компенсации ДОЛЖЕН БЫТЬ выполнен в виде двух треугольников, повернутых влево (как кнопка перемотки назад на проигрывателе).
    - Маркер Компенсации МОЖЕТ сочетаться с любым другим Маркером Подпроцесса.
- Все вышеописанные Маркеры ДОЛЖНЫ БЫТЬ сгруппированы и располагаться в центре нижней части графического элемента Подпроцесса.



Фигура 9.8 – Маркеры Свернутого Подпроцесса

Атрибуты подпроцесса

Таблица 9.13 содержит информацию об атрибутах Подпроцесса, продолжающих список стандартных атрибутов Действия:

Таблица 9.13 – Атрибуты Подпроцесса

Атрибут	Описание
SubProcessType (Embedded | Independent | Reference) Embedded : String	Атрибут SubProcessType, значение которого по умолчанию равно Embedded, является атрибутом, определяющим, находятся ли детали Подпроцесса в Процессе верхнего уровня или относятся к другому, повторно используемому Процессу.
IsATransaction False : Boolean	Атрибут IsATransaction указывает на то, будет ли ход Подпроцесса соответствовать Транзакции или нет (дополнительную информацию см. в части «Sub-Process Behavior as a Transaction», стр.59).
Transaction (0-1) : Transaction	В случае, если значение атрибута Transaction равно False, то тип Транзакции указываться НЕ ДОЛЖЕН. Дополнительную информацию об атрибутах Транзакции см. в части B.11.10 «Transaction», стр. 271. Обратите внимание, что Транзакции, расположенные в разных Пулах и соединенные посредством Потока сообщений, ДОЛЖНЫ ИМЕТЬ один и тот же идентификатор TransactionId.

Встроенный подпроцесс (Embedded Sub-Process)

Встроенный (или вложенный (nested)) Подпроцесс представляет собой действие, содержащее другие Действия (Процесс). Процесс, встроенный в другой Процесс, зависит от родительского Процесса и имеет доступ к данным контекста родительского Процесса. В данном случае нет необходимости указывать соответствие между данными Встроенного Подроцесса и данными родительского Процесса..

Будучи зависимыми от родительских элементов, графические элементы, расположенные внутри встроенного Подпроцесса, не имеют всех тех свойств, присущих графическим элементам полной диаграммы бизнес-процесса, какими обладают Пулы и Дорожки. Таким образом, Встроенный Подпроцесс в развернутом виде может содержать лишь Элементы потока, Соединяющие элементы и Артефакты (см. фигуру 9.7).

Атрибуты Встроенного Подпроцесса (атрибут SubProcessType имеет значение Embedded), приведенные в таблице 9.14, являются дополнительными и продолжают список атрибутов Подпроцесса:

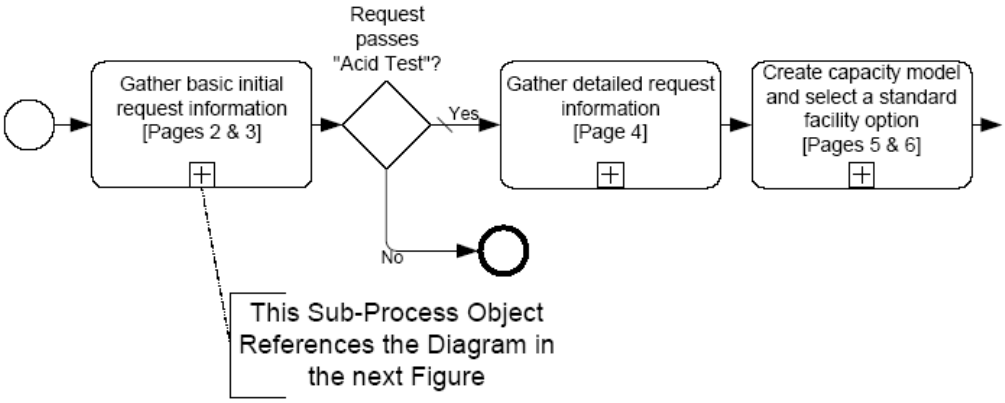
Таблица 9.14 – Атрибуты Встроенного Подпроцесса

Атрибут	Описание
GraphicalElements (0-n) : Object	Атрибут GraphicalElements определяет все графические элементы (например, События, Действия, Шлюзы или Артефакты), включенные во Встроенный Подпроцесс.
AdHoc False : Boolean	Значение логического атрибута AdHoc по умолчанию равно False. Данный атрибут указывает на то, имеет ли

	данный Встроенный Подпроцесс тип AdHoc или нет. Действия внутри Встроенного Подпроцесса типа AdHoc не контролируются и не упорядочиваются. Выполнение действия осуществляется исполнителями действия.
[AdHoc = True only] **AdHocOrdering** (0-1) (Sequential | Parallel) Parallel : String	В случае, если Встроенный Подпроцесс имеет тип AdHoc (значение атрибута AdHoc равно True), то ДОЛЖЕН БЫТЬ добавлен атрибут AdHocOrdering. Данный атрибут указывает, могут ли действия внутри данного Процесса быть выполнены параллельно или должны быть последовательными. По умолчанию значение данного атрибута равно Parallel, а значение Sequential является ограничением выполнения действий, которое бывает необходимо благодаря общим источникам.
[AdHoc = True only] **AdHocCompletionCondition** (0-1) : Expression	В случае, если Встроенный Подпроцесс имеет тип AdHoc (значение атрибута AdHoc равно True), то ДОЛЖЕН БЫТЬ добавлен атрибут CompletionCondition. Данный атрибут определяет условия, при которых завершается Процесс. Маркер AdHoc ДОЛЖЕН быть помещен в центр нижней части графического элемента Процесса AdHoc или Подпроцесса, включенного в Процесс AdHoc.

Независимый подпроцесс

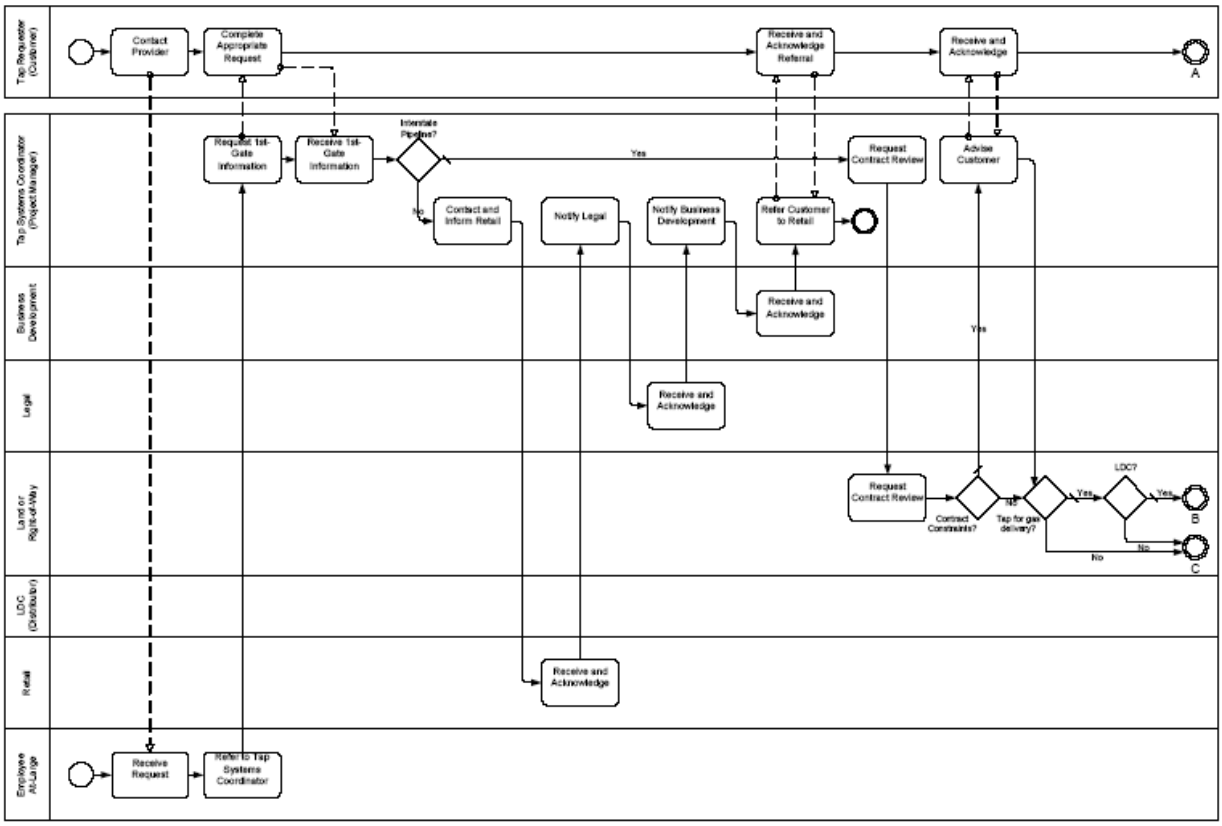
Независимый Подпроцесс является действием в составе Процесса, которое вызывает другой Процесс, включенный в диаграмму бизнес-процесса (см. фигуру 9.9). Вызываемый Процесс не зависит от родительского Процесса при вызове и не зависит от контекста родительского Процесса. Экземпляр Процесса будет создан при вызове, однако, он также может быть создан при вызове Независимого Подпроцесса другой диаграммы или при помощи внешнего источника. Независимый Подпроцесс может передавать данные как от вызываемого Процесса, так и родительскому Процессу.



Фигура 9.9 – Графические Элементы с Детальной Информацией, Изображенной на Следующей Диаграмме

Вызванный Процесс будет изображаться на отдельной диаграмме, которая может содержать несколько Пулов. Любой вид инициируемого Процесса (включая развернутый его вид внутри инициируемого Процесса) отображает полную диаграмму, в которую

включен данный инициируемый Процесс (см. Фигуру 9.10). Однако любые изображаемые данные будут относится лишь к данному Процессу, а не к какому-либо другому Процессу, включенному в данную диаграмму.



Фигура 9.10 – Процесс и Данные Подпроцесса Предыдущей Фигуры

Таблица 9.15 содержит информацию о дополнительных атрибутах Независимого Подпроцесса (где значение атрибута SubProcessType равно Independent), продолжающих список общих атрибутов Подпроцесса:

Таблица 9.15 – Атрибуты Независимого Подроцесса

Атрибут	Описание
DiagramRef : Business Process Diagram	ДОЛЖНА БЫТЬ указана диаграмма бизнес-процесса. Дополнительную информацию см. в части 8.5, «Business Process Diagram Attributes», стр. 28.
ProcessRef : Process	ДОЛЖЕН БЫТЬ указан Процесс. Дополнительную информацию см. в части 8.6, «Processes», стр. 29.
InputPropertyMaps (0-n) : Expression	Между свойствами Независимого Подпроцесса и свойствами родительского Процесса могут быть соответствия по входным значениям в зависимости от Процесса. Они могут быть представлены в виде выражений (однако инструмент моделирования может изобразить их любыми другими способами).
OutputPropertyMaps (0-n) : Expression	Между свойствами Независимого Подпроцесса и свойствами родительского Процесса могут быть соответствия по выходным значениям в зависимости от Процесса. Они могут быть представлены в виде

	выражений (однако инструмент моделирования может изобразить их любыми другими способами).

Ссылочный подпроцесс

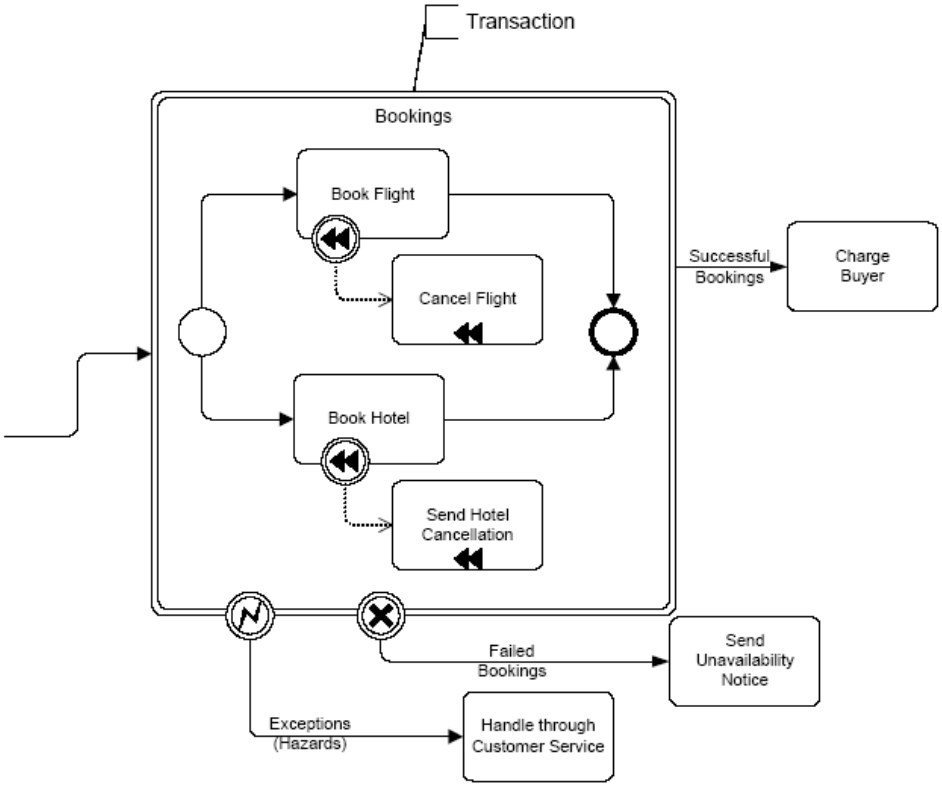
В отдельных случаях моделирования необходимо создание ссылки на Процесс, который был определен ранее. Если два отдельно взятых Подпроцесса имеют одно и то же поведение и атрибуты, то, используя ссылку на Процесс, можно определить атрибуты и поведение Процесса в одном месте и централизованно поддерживать Подпроцесс.

Таблица 9.16 содержит информацию об атрибутах Справочного Подпроцесса (где атрибут SubProcessType имеет значение Reference), продолжающих список общих атрибутов Подпроцесса:

Таблица 9.16 – Атрибуты Справочного Подроцесса

Атрибут	Описание
SubProcessRef : Task	ДОЛЖЕН БЫТЬ определен Подпроцесс, на который ссылаются. Дополнительную информацию см. в таблице 9.13.

Подпроцесс как транзакция



Фигура 9.11 – Пример Транзакционного Развернутого подпроцесса

Как Свернутый, так и развернутый Подпроцессы могут использоваться в качестве Транзакций, которые имеют определенное поведение, контролируемое протоколом Транзакции (таким, как протокол BTP WS-Transaction). Границы действия выполнены двойной линией для того, чтобы подчеркнуть использование Подпроцесса в качестве Транзакции (см. фигуру 9.11).

Существуют три основных результата работы Транзакции:

- Удачное завершение. Выражается в виде Стандартного Потока операций, выходящего из Подпроцесса.
- Неудачное завершение/Отмена (Cancel). В случае, если Транзакция отменена, то Действия, включенные в данную Транзакцию, будут причислены к действиям отмены, что может привести к движению процесса в обратном направлении, а также к Компенсации определенных Действий. Обратите внимание, что механизм прерывания Подпроцесса не запускает Компенсацию (например, Ошибка, Таймер, а также другие действия, не относящиеся к Транзакции). Промежуточное событие «Отмена», присоединенное к границам Действия, управляет Потоком операций после того, как Транзакция направлена в обратном направлении, а все действия компенсации выполнены. Промежуточное событие «Отмена» используется лишь тогда, когда оно присоединено к границам Транзакции и не может быть задействовано в Стандартном потоке операций или присоединено к Действиям, не относящимся к Транзакции. Существуют два механизма отмены Транзакции:
  - Конечное событие «Отмена» используется в лишь Подпроцессе типа Транзакция.
  - Сообщение об отмене может быть получено посредством протокола Транзакции, поддерживающего выполнение Подпроцесса.
- Опасность. Указывает на то, что какое-то действие выполняется крайне неверно, и невозможны удачное окончание или отмена процесса. Для отображения Опасности используется Ошибка. В случае возникновения Опасности действие прерывается (без Компенсации), а Поток операций возобновляется после Промежуточного события «Ошибка».

Ход Подпроцесса Транзакция с удачным завершением на конечном этапе немного отличается от хода стандартного Подпроцесса. Когда все маршруты Подпроцесса Транзакция достигают Конечного события, не имеющего тип «Отмена», Поток операций не направляется мгновенно к Родительскому Процессу верхнего уровня, как это происходит в стандартном Процессе. Прежде всего, Протоколом Транзакции утверждаются все участники, удачно завершившие Транзакцию. В большинстве случаев происходит именно так, и Поток операций перемещается к Процессу верхнего уровня. Однако возможна ситуация, когда у одного из участников возникают проблемы при завершении, что влечет за собой отмену или опасность. В таком случае, Поток операций направляется к Промежуточному событию , даже в случае, если оно завершилось удачно.

Примечание: Вопрос о точном поведении и нотации Транзакции является открытым. Для просмотра полного списка открытых вопросов BPMN обратитесь к Annex D.

Соединение потока операций

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока операций, обратитесь к пункту 8.4.1. «Правила Соединения Потоков Операций».

- Подпроцесс МОЖЕТ являться целью Потока сообщений и быть соединен с несколькими Входящими потоками операций. Входящий поток операций МОЖЕТ исходить как от альтернативных, так и от параллельных маршрутов.

Примечание: В случае, если Подпроцесс имеет несколько Входящих потоков операций, то данный Поток операций является Неконтролируемым. Это означает, что

Токен поступает от одного из маршрутов, и Подпроцесс выполняется, не ожидая появления других Токенов из остальных маршрутов. В случае, если Токен прибывает из того же самого или другого маршрута, то создается отдельный экземпляр Подпроцесса. Если же необходимо осуществлять контроль над Потоком операций, то данный Поток операций должен сойтись в одной точке со Шлюзом, предшествующим Подпроцессу (дополнительную информацию о Шлюзах см. в части 9.5, «Gateways», стр. 68).

- В случае, если Подпроцесс не имеет каких-либо Входящих потоков операций и не содержит в своем составе Стартового события, то при создании экземпляра Процесса ДОЛЖЕН БЫТЬ создан экземпляр данного Подпроцесса.
    - Исключением являются Подпроцессы, считающиеся действиями Компенсациии (имеющие маркер Компенсации). Подпроцесс типа Компенсация не рассматривается в качестве составляющего элемента Стандартного потока операций и НЕ ДОЛЖЕН БЫТЬ продублирован при создании экземпляра Процесса.
- Подпроцесс МОЖЕТ являться источником Потока операций и иметь несколько Исходящих потоков операций. В случае, если Подпроцесс соединен с несколькими Исходящими потоками операций, то для каждого из них должны быть созданы отдельные параллельные маршруты.

Для каждого отдельно взятого Исходящего потока операций, соединенного с Подпроцессом, создаются Токены. Идентификаторы каждого из Токенов устанавливаются таким образом, чтобы указать на то, что все данные Токены берут начало из одного и того же параллельного Раздвоения (И-Разделения), а также количество Токенов, идущих параллельно.

- В случае, если Подпроцесс не соединен ни с одним Исходящим потоком операций и не содержит в своем составе Конечного события, то данный Подпроцесс свидетельствует об окончании одного или более маршрутов Процесса. Когда Подпроцесс завершается при отсутствии других параллельных маршрутов, то Процесс ДОЛЖЕН БЫТЬ закончен.
    - Исключением являются Подпроцессы, считающиеся действиями Компенсациии (имеющие маркер Компенсации). Подпроцесс типа Компенсация не рассматривается в качестве составляющего элемента Стандартного потока операций и НЕ ДОЛЖЕН свидетельствовать об окончании Процесса.

Соединение потока сообщений

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока сообщений, обратитесь к пункту 8.4.2 «Правила Соединения Потоков Сообщений».

Примечание: Все Потоки сообщений должны соединять два разных Пула. Они могут быть присоединены к границам Пулов или Элементам потока, находящимся внутри Пулов. Потоки сообщений не могут соединять два графических элемента внутри одного Пула.

- Подпроцесс МОЖЕТ являться целью Потока сообщений и иметь 0 или более Входящих потоков операций.

- Подпроцесс МОЖЕТ являться источником Потока сообщений и иметь 0 или более Исходящих потоков операций.

§§ 9.4.3. Задача (Task)

Задача является элементарным действием, входящим в Процесс. Задача используется в случае, если данный Процесс не детализируется в данной Модели. Обычно исполнителями Задачи являются конечный пользователь и/или приложение.

Задача, как и Подпроцесс, изображается в виде прямоугольника с закругленными углами (см. фигуру 9.12).

- Задача представляет собой прямоугольник с закругленными углами, выполненный одинарной, тонкой, черной линией.
    - Текст, цвет, размер, а также линии, используемые для изображения Задачи, ДОЛЖНЫ соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм».

Фигура 9.12 – Графический Элемент Задача

BPMN различает три типа маркеров Задачи: Маркер Цикла (Loop Marker), Многоэкземплярный Маркер (Multiple Instance Marker), а также Маркер Компенсации (Compensation Marker). Задача может содержать от одного до двух вышеуказанных Маркеров (см. Фигуру 9.13).

- Маркер Цикла ДОЛЖЕН БЫТЬ выполнен в виде небольшой стрелки, острие которой загнуто в направлении, противоположном направлению самой стрелки.
    - Маркер Цикла МОЖЕТ сочетаться с Маркером Компенсации.
- Многоэкземплярный Маркер ДОЛЖЕН БЫТЬ выполнен в виде двух параллельных вертикальных линий.
    - Многоэкземплярный Маркер МОЖЕТ сочетаться с Маркером Компенсации.
- Маркер Компенсации ДОЛЖЕН БЫТЬ выполнен в виде двух треугольников, повернутых влево (как кнопка перемотки назад на проигрывателе).
    - Маркер Компенсации МОЖЕТ сочетаться как с Маркером Цикла, так и с Многоэкземплярным Маркером.
- Все данные Маркеры ДОЛЖНЫ БЫТЬ сгруппированы и располагаться в центре нижней части графического элемента Задачи.

Все Маркеры Задачи будут сгруппированы и располагаться в центре нижней части графического элемента Задачи.

Маркер Цикла	**Многоэкземплярный Маркер**	**Маркер Компенсации**

Фигура 9.13 – Маркеры Задачи

Помимо указанных типов Задач, в BPMN существуют другие их типы, цель которых заключается в различении поведения, присущего Задачам (см. таблицу 9.2). Однако BPMN не содержит каких-либо графических индикаторов для разных типов Задач. Инструмент моделирования может создавать собственные индикаторы или маркеры для указания типа Задачи, включенной в диаграмму. BPMN позволяет добавлять новые маркеры, однако, графический элемент Задачи, представляющий собой прямоугольник с закругленными углами, изменяться не должен. Перечень типов Задач может быть расширен за счет добавления необходимых индикаторов.

Атрибуты задачи

Таблица 9.17 содержит информацию об атрибутах Задачи, продолжающих список общих атрибутов Действия (см. таблицу 9.10):

Таблица 9.17 – Атрибуты Задачи

Атрибут	Описание
TaskType (Service | Receive | Send | User | Script | Manual | Reference | None) None : String	Атрибут TaskType, значение которого по умолчанию равно Service, МОЖЕТ иметь значения Send, Receive, User, Script, Manual, Reference, а также None. В случае, если в Процессе присутствует Поток сообщений, то данный атрибут зависит от Потока сообщений, направленного либо к Задаче, либо от неё. Атрибут TaskType, имеющий значение Receive, НЕ ДОЛЖЕН БЫТЬ соединен ни с одним Исходящим потоком сообщений. Атрибут TaskType, имеющий значение Send, НЕ ДОЛЖЕН БЫТЬ соединен ни с одним Входящим потоком сообщений. Атрибут TaskType, имеющий значения Script, Manual или None, НЕ ДОЛЖЕН БЫТЬ соединен ни с Входящими, ни с Исходящими потоками сообщений. Перечень значений атрибута TaskType может быть расширен за счет добавления новых. Дополнительную информацию о некоторых значениях атрибута TaskType см. в таблицах 9.18 и 9.24.

Сервисная задача (Service Task)

Сервисная задача представляет собой Задачу, предназначенную для оказания услуги, которая может являться как веб-сервисом (Web service), так и автоматизированным приложением.

Таблица 9.18 содержит информацию об атрибутах Сервисной задачи (где значение атрибута TaskType равно Service), продолжающих список общих атрибутов Задачи (см. таблицу 9.17):

Таблица 9.18 – Атрибуты Сервисной задачи

Атрибут	Описание
InMessage : Message	Для атрибута InMessage ДОЛЖНО БЫТЬ введено значение Message, указывающее на то, что при запуске Задачи, а также при наличии совокупности значений входных данных будет отослано сообщение. При этом на диаграмме МОЖЕТ БЫТЬ изображен соответствующий Исходящий поток сообщений, однако, его присутствие на диаграмме не является обязательным.
OutMessage : Message	Для атрибута OutMessage ДОЛЖНО БЫТЬ введено значение Message. Получение сообщения указывает на завершение Задачи, которое может привести к созданию совокупности значений выходных данных. На диаграмме МОЖЕТ БЫТЬ изображен соответствующий Входящий поток сообщений, однако, его присутствие на диаграмме не является обязательным.
Implementation (Web Service | Other | Unspecified) Web Service : String	Атрибут Implementation указывает технологию, используемую для отправки или получение сообщений. Технологией по умолчанию является веб-сервис (Web service).

Получение сообщений (Receive Task)

Получение сообщений представляет собой Задачу, суть которой заключается в ожидании сообщения, которое должно поступить от внешнего участника Процесса (имеющего отношение к данному бизнес-процессу). Задача считается выполненной в случае, если сообщение было получено хотя бы один раз. Получение сообщения часто используется для запуска процесса. В некотором смысле именно получение сообщения запускает Процесс. Для создания экземпляра Процесса, должно быть выполнено одно из следующих условий:

- Процесс не имеет в своем составе Стартового события, в Получение сообщений не присоединено ни к одному Входящему потоку операций.
- Входящий поток операций для Получения сообщений содержит источник Стартового события.
    - Обратите внимание, что никакой другой Входящий поток операций не может быть соединен с Получением сообщений (в частности, не должно быть соединения с циклом следующего графического элемента).

Таблица 9.19 содержит информацию об атрибутах Получения сообщений (где значение атрибута TaskType равно Receive), продолжающих список общих атрибутов Задачи (таблица 9.17):

Таблица 9.19 – Атрибуты Получения Сообщений

Атрибут	Описание

Message: Message	Для атрибута Message ДОЛЖНО БЫТЬ введено значение Message, указывающее на то, что сообщение будет получено данной Задачей. В данном случае сообщение является эквивалентом образца сообщения *in-only* (Web service). При этом на диаграмме МОЖЕТ БЫТЬ изображен соответствующий Входящий поток сообщений, однако, его присутствие на диаграмме не является обязательным.
Instantiate False : Boolean	Задача типа Получение сообщений может являться механизмом создания экземпляров Процесса, имеющего атрибут Instantiate. Данный атрибут МОЖЕТ иметь значение True в случае, если первое действие следует за Стартовым событием либо за исходной Задачей в случае отсутствия Стартового события (например, Процесс не содержит Входящих потоков операций). Данный атрибут со значением True может использоваться многими Задачами.
Implementation (Web Service | Other | Unspecified) Web Service : String	Атрибут Implementation указывает технологию, используемую для получения сообщений. Технологией по умолчанию является веб-сервис (Web service).

Отправка сообщений (Send Task)

Отправка сообщений представляет собой Задачу, суть которой заключается в отправке сообщения внешнему участнику Процесса (имеющему отношение к данному бизнес-процессу). Задача считается выполненной в случае, если сообщение было отправлено хотя бы один раз.

Таблица 9.20 содержит информацию об атрибутах Отправки сообщений (где значение атрибута TaskType равно Send), продолжающих список общих атрибутов Задачи (таблица 9.17):

Таблица 9.20 – Атрибуты Отправки Сообщений

Атрибут	Описание
Message : Message	Для атрибута Message ДОЛЖНО БЫТЬ введено значение Message, указывающее на то, что сообщение будет отправлено данной Задачей. В данном случае сообщение является эквивалентом образца сообщения *out-only* (Web service). При этом на диаграмме МОЖЕТ БЫТЬ изображен соответствующий Исходящий поток сообщений, однако, его присутствие на диаграмме не является обязательным.
Implementation (Web Service | Other | Unspecified) Web Service : String	Атрибут Implementation указывает технологию, используемую для отправки сообщений. Технологией по умолчанию является веб-сервис (Web service).

Пользовательская задача (User Task)

Пользовательская задача представляет собой задачу, типичную для технологического процесса, где человек участвует в качестве исполнителя и выполняет Задачи с помощью программного обеспечения. Для выполнения Задачи каждый человек назначается каким-либо администратором Задач.

Таблица 9.21 содержит информацию об атрибутах Пользовательской Задачи (где значение атрибута TaskType равно User), продолжающих список общих атрибутов Задачи (таблица 9.17):

Таблица 9.21 – Атрибуты Пользовательской Задачи

Атрибут	Описание
Performers (1-n) : String	МОГУТ быть добавлены один или несколько атрибутов Performers. Атрибут Performers указывает на то, какие человеческие ресурсы будут выполнять Пользовательскую задачу. Исполнителем может выступать как отдельно взятый человек, так и группа людей и организация. Инструмент моделирования может добавлять дополнительные параметры, помогающие определить задание Исполнителя.
InMessage : Message	Для атрибута InMessage ДОЛЖНО БЫТЬ введено значение Message, указывающее на то, что сообщение будет отправлено в начале выполнения Задачи при наличии совокупности значений каких-либо определенных входных данных. При этом на диаграмме МОЖЕТ БЫТЬ изображен соответствующий Исходящий поток сообщений, однако, его присутствие на диаграмме не является обязательным.
OutMessage : Message	Для атрибута OutMessage ДОЛЖНО БЫТЬ введено значение Message. Получение сообщения указывает на завершение выполнения Задачи, что может привести к созданию совокупности значений выходных данных. При этом на диаграмме МОЖЕТ БЫТЬ изображен соответствующий Входящий поток сообщений, однако, его присутствие на диаграмме не является обязательным.
Implementation (Web Service | Other | Unspecified) Other : String	Атрибут Implementation указывает технологию, используемую для отправки сообщений. Технологией по умолчанию является веб-сервис (Web service).

Задача-сценарий (Script Task)

Задача-сценарий выполняется механизмом выполнения бизнес-процессов. Инструмент моделирования определяет сценарий, используя тот язык, который может быть интерпретирован механизмом выполнения бизнес-процессов. Когда Задача готова начаться, данный механизм задействует уже определенный сценарий. При завершении сценария завершается и сама Задача.

Таблица 9.22 содержит информацию об атрибутах Задачи-сценария (где значение атрибута TaskType равно Script), продолжающих список общих атрибутов Задачи (таблица 9.17):

Таблица 9.22 – Атрибуты Задачи-сценария

Атрибут	Описание
Script (0-1) : String	Инструмент моделирования МОЖЕТ использовать сценарий, продолжающийся даже после выполнения Задачи. В случае, если сценарий не используется, то Задача рассматривается в качестве имеющей атрибут TaskType со значением None.

Ручное выполнение (Manual Task)

Ручное выполнение представляет собой задачу, выполнение которой предусматривается без помощи механизма выполнения бизнес-процесса или какого-либо приложения. Примером такого типа задачи может служить монтажник, устанавливающий телефон в местонахождении клиента.

Таблица 9.23 содержит информацию об атрибутах Ручного выполнения (где значение атрибута TaskType равно Manual), продолжающих список общих атрибутов Задачи (таблица 9.17):

Таблица 9.23 – Атрибуты Ручного Выполнения

Атрибут	Описание
Performers (0-n) : String	МОГУТ БЫТЬ введены одно или несколько значений Performers. Данный атрибут Performers указывает на человеческие ресурсы, направленные на выполнение Задачи типа Ручное выполнение. Исполнителем может выступать как отдельно взятый человек, так и группа людей и организация, а также роль или позиция организации.

Ссылочная задача (Reference Task)

В отдельных случаях необходимо создание ссылки на другое, ранее определенное Действие. Если два (или более) отдельно взятых Действия имеют одно и то же поведение и атрибуты, то, используя ссылку на Действие, можно определить атрибуты и поведение Действий в одном месте и централизованно их поддерживать.

Таблица 9.24 содержит информацию об атрибутах Ссылочной задачи (где значение атрибута TaskType равно Reference), продолжающих список общих атрибутов Задачи (таблица 9.17):

Таблица 9.24 – Атрибуты Ссылочной Задачи

Атрибут	Описание
TaskRef : Task	ДОЛЖНА БЫТЬ определена Задача, на которую ссылаются. Дополнительную информацию см. в таблице 9.17.

Соединение потока операций

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций».

- Задача МОЖЕТ являться целью Потока операций и быть соединена с несколькими Входящими потоками операций. Входящий поток операций МОЖЕТ исходить как от альтернативных, так и от параллельных маршрутов.

Примечание: В случае, если Задача имеет несколько Входящих потоков операций, то данный Поток операций является Неконтролируемым. Это означает, что Токен поступает от одного из маршрутов, и Задача выполняется, не ожидая появления других Токенов из остальных маршрутов. В случае, если Токен прибывает из того же самого или другого маршрута, то создается отдельный экземпляр Задачи. Если же необходимо осуществлять контроль над Потоком операций, то данный Поток операций должен сойтись в одной точке со Шлюзом, предшествующим Задаче (дополнительную информацию о Шлюзах см. в части 9.5, «Gateways», стр. 68).

- В случае, если Задача не имеет каких-либо Входящих потоков операций, а Процесс не содержит в своем составе Стартового события, то при создании экземпляра Процесса ДОЛЖЕН БЫТЬ создан экземпляр данной Задачи.
    - Исключением являются Задачи, считающиеся действиями Компенсациии (имеющие маркер Компенсации). Задача типа Компенсация не рассматривается в качестве составляющего элемента Стандартного потока операций и НЕ ДОЛЖНА БЫТЬ продублирована при создании экземпляра Процесса.
- Задача МОЖЕТ являться источником Потока операций и иметь несколько Исходящих потоков операций. В случае, если Задача соединена с несколькими Исходящими потоками операций, то для каждого из них должны быть созданы отдельные параллельные маршруты.

Для каждого отдельно взятого Исходящего потока операций, соединенного с Задачей, создаются Токены. Идентификаторы каждого из Токенов устанавливаются таким образом, чтобы указать на то, что все данные Токены берут начало из одного и того же параллельного Раздвоения (И-Разделения), а также количество Токенов, идущих параллельно.

- В случае, если Задача не соединена ни с одним Исходящим потоком операций, а Процесс не содержит в своем составе Конечного события, то данная Задача свидетельствует об окончании одного или более маршрутов Процесса. Когда Задача завершается при отсутствии других параллельных маршрутов, то Процесс ДОЛЖЕН БЫТЬ закончен.
    - Исключением являются Задачи, считающиеся действиями Компенсациии (имеющие маркер Компенсации). Задача типа Компенсация не рассматривается в качестве составляющего элемента Стандартного потока операций и НЕ ДОЛЖНА свидетельствовать об окончании Процесса.

Соединение потока сообщений

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока сообщений, обратитесь к пункту 8.4.2 «Правила Соединения Потоков Сообщений».

Примечание: Все Потоки сообщений должны соединять два разных Пула. Они могут быть присоединены к границам Пулов или Элементам потока, находящимся внутри Пулов. Потоки сообщений не могут соединять два графических элемента внутри одного Пула.

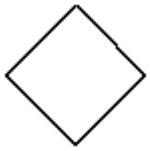
- Задача МОЖЕТ являться целью Потока сообщений и иметь 0 или более Входящих потоков операций.
- Задача МОЖЕТ являться источником Потока сообщений и иметь 0 или более Исходящих потоков операций.

§ 9.5. Шлюз (Gateway)

Шлюзы используются для контроля схождений и расхождений множественных Потоков операций. Необходимость в использовании Шлюзов отпадает в случае, если нет надобности контролировать Поток операций. Термин «Шлюз» подразумевает пропускное устройство, которое либо позволяет осуществлять переход через Шлюз, либо нет. Таким образом, как только Токены подходят к Шлюзу, то при его активизации они могут объединиться у входа в Шлюз и/или разделиться при выходе из Шлюза. При детальном рассмотрении Шлюз представляет собой совокупность «Входов» и «Выходов» (Gates). Существует несколько видов Шлюзов (см. описание ниже), поведение каждого из которых определяет, как много Выходов будет использовано для продолжения хода Потока операций. Для каждого Исходящего потока операций, относящегося к Шлюзу, будет задействован один Вход и один Выход.

Графический элемент Шлюза представляет собой небольшой ромб (см. фигуру 9.14), используемый во многих нотациях схем производственных процессов для изображения ветвления и знакомый большинству инструментов моделирования.

- Шлюз представляет собой небольшой ромб, который ДОЛЖЕН БЫТЬ выполнен одиночной, тонкой, черной линией.
    - Текст, цвет, размер, а также линии, используемые для изображения Шлюза, ДОЛЖНЫ соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм».



Фигура 9.14 - Шлюз

Примечание: Несмотря на то, что Шлюз представляет собой ромб, Входящие и Исходящие потоки операций могут присоединяться к любой точке границы ромба, а не только к его углам.

При помощи Шлюзов можно указать все типы поведения Потоков операций бизнес-процесса: Условие/ветвление (ИЛИ-Разделение; Эксклюзивный ИЛИ; Включающий ИЛИ; Комплексный), Слияние (ИЛИ-Соединение), Раздвоение (И-Разделение), Соединение (И-Соединение). Таким образом, BPMN расширил использование ромба для указания любого типа Шлюзов Эксклюзивных Исключающих шлюзов. Все Шлюзы имеют индикаторы или маркеры, расположенные внутри графического элемента, и указывающие на то, какой тип имеет тот или иной используемый Шлюз (см. фигуру 9.15).

Эксклюзивные Условия/ Объединения (ИЛИ)

Основанные на данных	*или*
Основанные на событиях	
Включающие Условия/Объединения (ИЛИ)	
Комплексные Условия/Объединени	
Параллельное Раздвоение/Соединение (И)	

Фигура 9.15 – Различные типа Шлюзов

- Внутренний маркер, указывающий на тип Шлюза, ДОЛЖЕН располагаться внутри графического элемента Шлюза. Размер и расположение маркера внутри ромба могут быть любыми и зависят от требований инструмента моделирования. Эксклюзивные шлюзы, основанные на данных, не нуждаются в добавлении маркера.

Шлюзы контролируют ход как сходящихся, так и расходящихся Потоков операций. Это означает, что отдельно взятый Шлюз может быть соединен с несколькими Входящими и Исходящими потоками операций одновременно. Тип Шлюза указывает на соответствующее поведение как сходящихся, так и расходящихся Потоков операций. Инструменты моделирования могут использовать два последовательных Шлюза для соединения, а затем и для разделения Потоков операций, т.е. выполнять лишь одну функцию.

§§ 9.5.1. Общие свойства шлюзов

Рассматриваются атрибуты шлюзов, а также правила соединения потоков операций и сообщений в шлюзе.

Общие атрибуты шлюзов

Таблица 9.25 содержит информацию об атрибутах Шлюзов, продолжающих список общих атрибутов Элементов потока (таблица 9.2):

Таблица 9.25 – Общие Атрибуты Шлюзов

Атрибут	Описание
GatewayType (XOR | OR | Complex | AND) XOR : String	Значение атрибута GatewayType по умолчанию равно XOR, однако, оно может быть изменено на OR, Complex или AND. Данный атрибут определяет поведение Шлюза

	как для Входящих, так и для Исходящих потоков операций, а также внутренний индикатор Шлюза (см. фигуру 9.15).

Соединение потока операций в шлюзе

Данные правила относятся к использованию всех типов Шлюзов. Ниже описаны дополнительные правила соединения Потока операций для каждого типа Шлюзов. Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций».

- Шлюз МОЖЕТ БЫТЬ целью Потока сообщений и быть соединен с несколькими Входящими потоками операций. Входящий поток операций МОЖЕТ исходить как от альтернативных, так и от параллельных маршрутов.
    - В случае, если Шлюз не имеет каких-либо Входящих потоков операций, а Процесс не содержит в своем составе Стартового события, то расхождение Шлюза, зависящее от атрибута GatewayType (см. ниже), ДОЛЖНО быть выполнено при создании экземпляра Процесса.
- Шлюз МОЖЕТ являться источником Процесса и иметь 0 или несколько Исходящих потоков операций.
- Шлюз МОЖЕТ быть соединен как с Входящими, так и с Исходящими потоками операций.

Примечание: Входящие и Исходящие потоки операций могут присоединяться к любой точке границы ромба, а не только к его углам.

Соединение потока сообщений

Данные правила относятся к использованию всех типов Шлюзов. Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока сообщений, обратитесь к пункту 8.4.2 «Правила Соединения Потоков Сообщений».

- Шлюз НЕ МОЖЕТ являться целью Потока сообщений.
- Шлюз НЕ МОЖЕТ являться источником Потока сообщений.

§§ 9.5.2. Эксклюзивные шлюзы (ИЛИ) (Exclusive Gateways (XOR))

Эксклюзивные Шлюзы (Условия) включаются в состав бизнес-процесса, в котором Поток операций может идти по двум или более альтернативным маршрутам, что в действительности является «разделением» хода Процесса. Для данного экземпляра Процесса может быть выбран лишь один из предложенных маршрутов (не следует путать с раздвоением маршрута, относящимся к «Раздвоению Потока операций», стр. 110). Условие не является Действием с точки зрения бизнес-процесса, однако, представляет собой один из видов Шлюзов, осуществляющих контроль над Потоком операций на отрезке, ограниченном Действиями. Примером может послужить вопрос, задаваемый в данный момент Процесса и предполагающий несколько вариантов ответов (Шлюзов). Каждый Шлюз типа Условие взаимодействует с условным выражением Исходящего потока операций. Во время выполнения Процесса выбирается Шлюз и соответствующий ему Поток операций. Токен, подходящий к Шлюзу, направляется вниз по определенному маршруту, соответствующему выбранному Шлюзу.

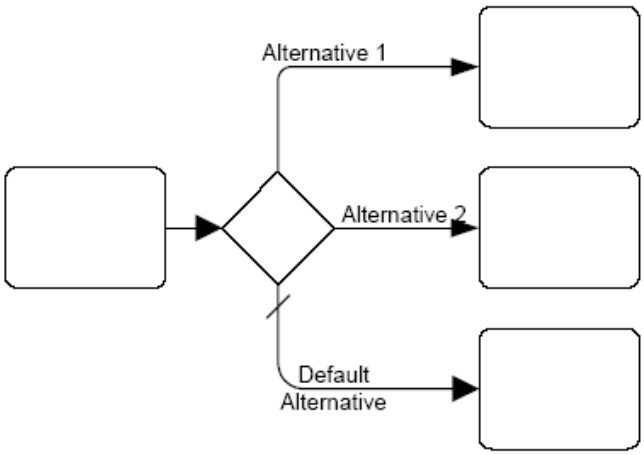
Эксклюзивное Условие может быть соединено с двумя или более Исходящими потоками операций, однако в ходе выполнения Процесса может быть выбран лишь один из них. Таким образом, Эксклюзивное Условие определяет набор альтернативных маршрутов Токена при пересечении им Потока операций. Существуют два вида Эксклюзивных Условий: Эксклюзивные Условия, основанные на Данных, и Эксклюзивные Условия, основанные на Событиях.

Эксклюзивные условия, основанные на данных (Data-Based Exclusive Decisions)

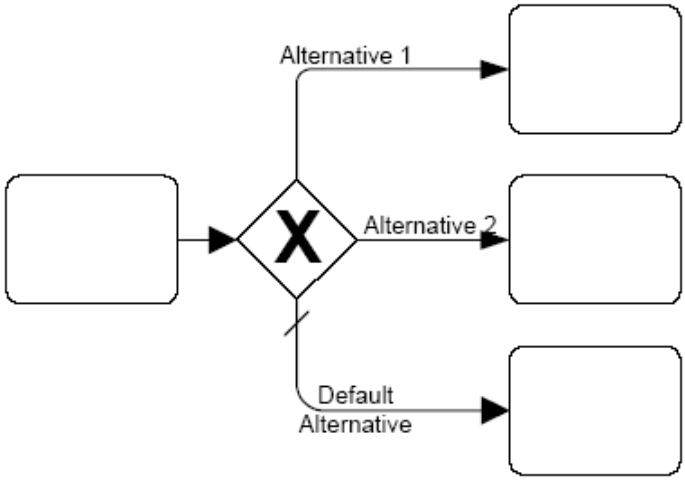
Эксклюзивные Условия, основанные на Данных, являются наиболее часто встречающимися Шлюзами. Набор Выходов для данного Условия основан на логическом выражении, содержащемся в атрибуте ConditionExpression Исходящего потока операций данного Шлюза. Для определения необходимого маршрута такое выражение использует значения данных Процесса (о чем свидетельствует и название данного вида Шлюзов – «основанные на Данных»).

Примечание: BPMN не указывает на формат выражений, используемых в Шлюзах или каких-либо других элементах BPMN, использующих выражения.

- Эксклюзивные Условия, основанные на Данных, МОГУТ содержать маркер, изображаемый как «X» и помещенный в ромб, символизирующий Шлюз (см. фигуру 9.17), для того, чтобы без труда можно было отличить данный вид Шлюзов от других. Данный маркер не является обязательным (см. фигуру 9.16).
    - Использование внутреннего маркера «X» ДОЛЖНО БЫТЬ согласованным. Это означает, что при использовании данного маркера в одних Шлюзах, необходимо его использование и в остальных Шлюзах.



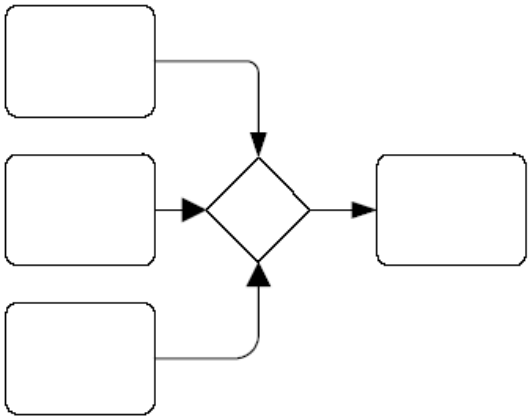
Фигура 9.16 – Пример Эксклюзивных Условий, Основанных на Данных, не Содержащих Внутреннего Маркера



Фигура 9.17 – Пример Эксклюзивных Условий, Основанных на Данных, Содержащих внутренний маркер

Условия для альтернативных Выходов должны быть вычислены в определенном порядке. Первое условие имеет значение TRUE и указывает на то, какой Поток операций будет выбран. Т.к. Шлюз является эксклюзивным, то все условия, имеющие значения True, будут игнорироваться. При этом выбирается лишь один Выход. Один из Выходов может быть маркирован по умолчанию (или иначе) и является в таком случае последними. Это означает, что если не выбран ни один из Выходов, то Поток операций пойдет по Выходу, установленному по умолчанию. Вместе с соответствующим Потоком операций выбираются лишь Входы и Выходы, установленные по умолчанию.

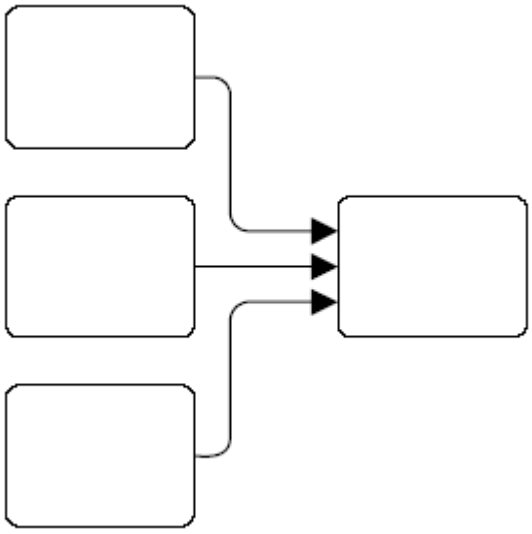
Определение Выхода по умолчанию не являются обязательным для данного Шлюза. Это означает, что в случае, если Выходы по умолчанию не определены, то ответственность, что по меньшей мере одно из условий вернет значение True во время выполнения Процесса, ложится на инструмент моделирования. Спецификация BPMN не указывает на последствия отсутствия действительных Выходов. Однако данная спецификация все-таки определяет то, что скрытых Потоков операций в Процессе БЫТЬ НЕ ДОЛЖНО, а все Стандартные потоки операций данного Процесса должны быть выполнены согласно Потоку операций. Это означает, что модель Процесса, включающая Шлюз, имеющий Выходы, потенциально недействительные в момент выполнения Процесса, считается неверной.



Фигура 9.18 – Эксклюзивное Слияние (Шлюз), не Имеющее Внутреннего Маркера

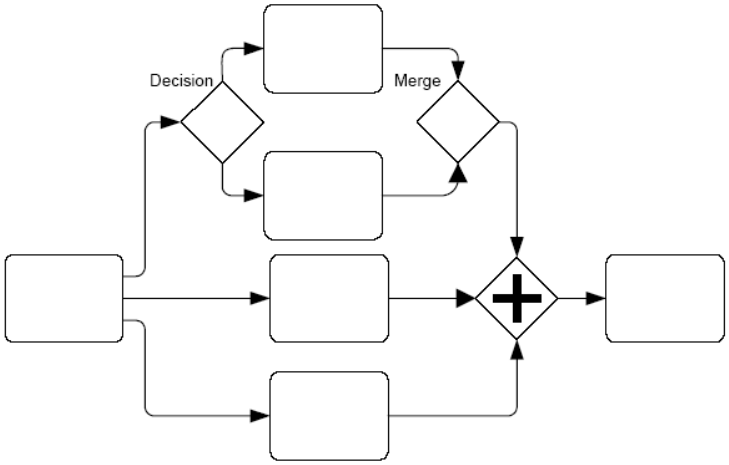
Эксклюзивные Шлюзы могут также использоваться для объединения альтернативных Потоков операций, несмотря на то, что инструменты моделирования используют данные Шлюзы в таком качестве достаточно редко. Объединяющее поведение Шлюза может быть

смоделировано наподобие фигуры 9.19. Фигуры 9.18 и 9.19 являются одинаковыми при наличии альтернативных Входящих потоков операций.



Фигура 9.19 – Неконтролируемое Слияние Потоков операций

В отдельных случаях Эксклюзивные Шлюзы включены в Процесс в качестве соединяющих объектов. На фигуре 9.20 (в оригинале спецификации ошибочно указана фигура 9.21) изображен Эксклюзивный Шлюз (отмеченный как «Merge»), соединяющий два альтернативных Потока операций, созданных более ранним Условием. Альтернативные Потоки операций сливаются перед Параллельным Шлюзом, координирующим несколько параллельных Потоков операций, даже созданных позднее. В случае, если объединяющий Шлюз в данный Процесс не включен, то Параллельный Шлюз объединит четыре Входящих потока операций. Однако одновременно провести Токен смогут лишь три из них. Таким образом, Шлюз будет находится в ожидании четвертого Токена, который никогда не появится. Следовательно, выполнение Процесса приостановится в точке, где располагается данный Параллельный Шлюз.



Фигура 9.20 – Эксклюзивный Шлюз, Объединяющий Потоки Операций, Предшествующие Параллельному Шлюзу

Обычно несложные ситуации не требуют использования Эксклюзивных Шлюзов для объединения Потоков операций, однако, в более сложных ситуациях их использование необходимо. Таким образом, инструмент моделирования должен представлять развитие ситуации, где Потоки операций не контролируется. В действительности, согласно

передовому мировому опыту, инструментам моделирования необходимо использовать Эксклюзивные шлюзы в любых ситуациях.

Атрибуты эксклюзивных шлюзов, основанных на данных

Таблица 9.26 содержит информацию об атрибутах Эксклюзивных Шлюзов, основанных на Данных. Данные атрибуты могут использоваться лишь в тех случаях, когда атрибут GatewayType имеет значение XOR. Нижеописанные атрибуты продолжают список общих атрибутов Шлюзов (см. таблицу 9.30):

9.26 - Атрибуты Эксклюзивных Шлюзов, основанных на Данных

Атрибуты	Описание
XORType (Data | Event) Data : String	Значение атрибута XORType по умолчанию равно Data, однако, оно МОЖЕТ БЫТЬ изменено на Event. Для атрибутов и поведения, определенных для применения Шлюзов, данный атрибут ДОЛЖЕН иметь значение Data до тех пор, пока Эксклюзивный Шлюз (ИЛИ), основанный на Данных, является объектом данной части.
MarkerVisible False : Boolean	Атрибут MarkerVisible указывает на то, будет ли Маркер ИЛИ («X») отображаться в центре графического элемента Шлюза. В случае, если данный атрибут имеет значение True, то маркер отображается в ромбе, если же атрибут имеет значение False, то ромб не содержит данный маркер. По умолчанию маркер не отображается.
Gates (0-n) : Gate	В Шлюзе МОЖЕТ БЫТЬ 0 или более Выходов. Отсутствие Выходов допустимо в том случае, если Шлюз является конечным элементом Процесса, а сам Процесс не содержит в своем составе Стартового или Конечного Событий. В случае, если Процесс содержит 0 либо только 1 Входящий поток операций (т.е. Шлюз представляет собой Условие), то данный Процесс ДОЛЖЕН содержать по меньшей мере один Выход. Если же Процесс не содержит Выходов, определенных по умолчанию, то ДОЛЖНО быть по меньшей мере по два Выхода.
[Gate] **OutgoingSequenceFlow**: SequenceFlow	Каждый Выход ДОЛЖЕН БЫТЬ соединен с Потоком операций. Данный Поток операций ДОЛЖЕН иметь атрибут Condition со значением Expression, а также действительное значение атрибута ConditionExpression. Дополнительную информацию об Атрибутах Потока операций см. в части 10.1.2, «Sequence Flow», стр. 100. В случае, если в Процессе содержится лишь один Выход, то атрибут Потока операций Condition ДОЛЖЕН иметь значение None.
[Gate] **Assigments** (0-n) : Assignment	Одно или более выражений назначения МОГУТ БЫТЬ выполнены для каждого Выхода. Назначение ДОЛЖНО БЫТЬ выполнено тогда, когда выбран данный Выход. Дополнительную информацию см. в части B.11.1,

	«Assignment», стр. 268.
DefaultGate (0-1) : Gate	Выход МОЖЕТ БЫТЬ определен по умолчанию.
[Gate] **OutgoingSequenceFlow**: SequenceFlow	В случае, если Шлюз имеет атрибут DefaultGate, то данный Шлюз ДОЛЖЕН БЫТЬ соединен с Потоком операций. Поток операций ДОЛЖЕН иметь индикатор, определенный по умолчанию (см. фигуру 9.16), а также атрибут Condition со значением Default. Дополнительную информацию об атрибутах Потока операций см. в части 10.1.2, «Sequence Flow», стр. 100.
[Gate] **Assigments** (0-n) : Assignment	Для атрибута DefaultGate МОЖЕТ БЫТЬ определено одно или более выражений назначения. Назначение ДОЛЖНО БЫТЬ выполнено тогда, когда выбран атрибут DefaultGate. Дополнительную информацию см. в части B.11.1, «Assignment», стр. 268.

Соединение потока операций

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций».

Исключающее поведение данного типа Шлюзов, направленное на объединение Потоков операций, заключается в следующем:

- В случае, если в Процессе содержится несколько Входящих потоков операций, то все они предназначены для продолжения хода Процесса (как если бы процесс не содержал каких-либо Шлюзов). Это означает, что:
    - Ход Процесса ДОЛЖЕН продолжиться в момент поступления импульса (Токена) от какого-либо Потока операций.
        - Импульсы от других Потоков операций, среди которых находится тот, от которого поступил вышеописанный импульс, могут поступить в другой раз, и Процесс также, без промедления и синхронизации данных импульсов, будет продолжен при их поступлении.

Исключающее поведение данного типа Шлюзов, направленное на разделение Потоков операций, заключается в следующем:

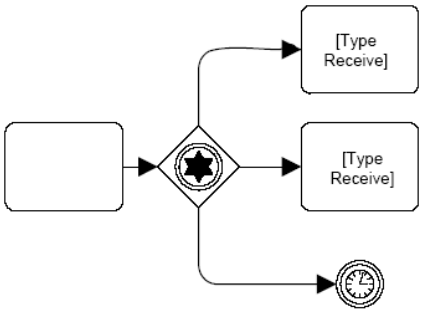
- В случае, если из Шлюза выходит несколько Исходящих потоков операций, то в ходе выполнения Процесса МОЖЕТ БЫТЬ выбран лишь один Выход.
    - Вход и Выход ДОЛЖНЫ быть определены в соответствие с результатом вычисления атрибута ConditionExpression, который относится к Потоку операций, ассоциируемому с данным Выходом.
        - Условие, относящееся к Выходу, ДОЛЖНО вычисляться в том порядке, в каком Выходы присутствуют в списке Выходов.
        - В случае, если атрибут ConditionExpression имеет значение TRUE, то ДОЛЖЕН быть выбран данный Выход, а значения для других Выходов, оставшихся в списке, вычисляться НЕ ДОЛЖНЫ .
        - В случае, если ни для одного из Выходов не указано значение условного выражения, равное TRUE, то ДОЛЖЕН БЫТЬ использован Выход по умолчанию.

Примечание: В случае, если Шлюз не имеет атрибута DefaultGate, и ни одно из условных выражений, относящихся к Выходу, не равно True, то считается, что такая модель Процесса не верна.

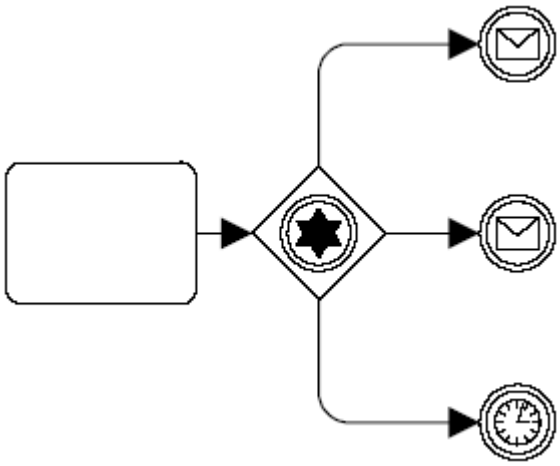
Эксклюзивные условия, основанные на событиях (Event-Based Exclusive Decisions)

Наличие в Процессе Эксклюзивных Условий, основанных на Событиях, стало возможным благодаря результатам новейших разработок в использовании распределенных систем (например, пи-вычисление). С точки зрения входов, данные Условия являются аналогичными Эксклюзивным Шлюзам, основанным на Данных (информацию об Эксклюзивных Условиях, основанных на Событиях, см. выше). А с точки зрения выходов, такой тип Условий в Процессе представляет собой точку ветвления, в которой направления альтернативных маршрутов зависят от событий, происходящих в данной точке Процесса, а не от определения значений выражений, использующих данные Процесса. Определенное событие, чаще всего – получение сообщения, определяет, какой из маршрутов будет выбран в данном случае. Например, в зависимости от ответа заказчика компания выполняет либо один набор действий, соответствующий положительному ответу, либо другой, соответствующий отрицательному ответу. Ответ заказчика – Сообщение определенного рода - определяет необходимый маршрут. Таким образом, сообщения «Да» и «Нет» являются разными, а не просто одинаковыми сообщениями с разными значениями атрибута Message. Получение сообщения может быть выполнено с помощью Задачи с атрибутом TaskTypeReceive либо при помощи Промежуточного события с Триггером Message. Кроме Триггера Message могут использоваться и другие Триггеры Промежуточного события, такие как Timer и Errors.

- Эксклюзивные Шлюзы, основанные на Событиях, ДОЛЖНЫ иметь маркер Множественного промежуточного события, расположенный внутри графического элемента Шлюза (см. фигуры 9.21 и 9.22), для того, чтобы без труда можно было отличить данный видов Шлюзов от других их видов.
- Эксклюзивные Шлюзы, основанные на Событиях, изменяются за счет Исходящих потоков операций, в качестве цели использующих Задачу с атрибутом TaskTypeReceive или Промежуточное событие (см. фигуру 9.21 и 9.22).
    - Все Исходящие потоки операций должны иметь этот тип целей. В данном случае не может быть смешения условных выражений с Промежуточным событием для данного условия.



Фигура 9.21 – Пример Эксклюзивного Условия, основанного на Событиях, Использующего Получение Сообщений



Фигура 9.22 - Пример Эксклюзивного Условия, основанного на Событиях, Использующего События о Получении Сообщений

Для соотнесения Эксклюзивного Условия, основанного на Событиях, с BPEL4WS, необходимо отметить, что графический элемент данного вида Шлюзов указывает на расположение BPEL4WS *pick*, а Промежуточные события, следующие за данным Условием, становятся обработчиками событий, относящихся к *pick* и *choice*. Действия. Следующие за Промежуточными событиями, становятся содержимым набора действий (*activity sets*) обработчиков событий. Границы действий определяются конфигурацией процесса. Это означает, что границы действий простираются до того места, где альтернативные маршруты в конце концов сливаются (данная точка может являться точкой завершения Процесса).

Объединяющая особенность Эксклюзивных Шлюзов, основанных на Событиях, является аналогичной объединяющему свойству Эксклюзивных Шлюзов, основанных на Данных, благодаря тому, что данный тип Шлюзов является Эксклюзивным (см. выше).

Шлюзы могут использоваться для запуска Процесса, а получение сообщения является в какой-то мере триггером для запуска Процесса. Получение любого из предварительно указанных в конфигурации Шлюза сообщений запускает Процесс. Таким образом, Шлюз обеспечивает наличие альтернативных способов инициирования Процесса.

Для того, чтобы Шлюз был способен инициировать Процесс, должны быть удовлетворены следующие требования:

- Процесс не имеет в своем составе Стартового события, а Шлюз не соединен ни с одним Входящим потоком операций.
- Источником Входящего потока операций для данного Шлюза является Стартовое событие.
    - Обратите внимание, что для данного Шлюза не допускается наличие какого-либо другого Входящего потока операций (например, присоединение цикла, относящегося к следующему объекту).
- Промежуточное событие типа Таймер НЕ ДОЛЖНО являться целью Исходящего потока операций данного Шлюза.

Атрибуты эксклюзивных шлюзов, основанных на событиях

Таблица 9.27 содержит информацию об атрибутах Эксклюзивных Шлюзов, основанных на Событиях. Данные атрибуты могут использоваться лишь в тех случаях, когда атрибут GatewayType имеет значение XOR. Нижеописанные атрибуты продолжают список общих атрибутов Шлюзов (см. таблицу 9.30):

9.27 - Атрибуты Эксклюзивных Шлюзов, основанных на Событиях

Атрибуты	Описание
XORType (Data | Event) Event : String	Значение атрибута XORType по умолчанию равно Data, однако, оно МОЖЕТ БЫТЬ изменено на Event. Для атрибутов и поведения, определенных для применения Шлюзов, данный атрибут ДОЛЖЕН иметь значение Event до тех пор, пока Эксклюзивный Шлюз (ИЛИ), основанный на Событиях, является объектом данной части.
Instantiate False : Boolean	Эксклюзивный Шлюз (ИЛИ), основанный на Событиях, и имеющий атрибут Instantiate, может являться механизмом создания экземпляра Процесса. В случае, если Шлюз является первым элементом Процесса, расположенным после Стартового события или исходного Шлюза, если Процесс не содержит в своем составе Стартового события (например, Процесс не содержит Входящих потоков операций), то значение данного атрибута ДОЛЖНО БЫТЬ изменено на True.
Gates (2-n) : Gate	В Шлюзе МОГУТ БЫТЬ 2 или более Выходов. Обратите внимание, что функция данного типа Шлюзов заключается не только в объединении, но ещё и в Условии.
[Gate] **OutgoingSequenceFlow** : SequenceFlow	Каждый Выход ДОЛЖЕН БЫТЬ соединен с Потоком операций. Данный Поток операций ДОЛЖЕН иметь атрибут Condition со значением None (в данном случае не происходит определение значения условного выражения). Дополнительную информацию об атрибутах Потока операций см. в части 10.1.2, «Sequence Flow», стр. 100.
[Gate] **Assigments** (0-n) : Assignment	Одно или более выражений назначения МОГУТ БЫТЬ выполнены для каждого Выхода. Назначение ДОЛЖНО БЫТЬ выполнено тогда, когда выбран данный Выход. Дополнительную информацию см. в части B.11.1, «Assignment», стр. 268.

Соединение потока операций

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций».

Исключающее поведение данного типа Шлюзов, направленное на объединение Потоков операций, заключается в следующем:

- В случае, если из Шлюза выходят несколько Входящих потоков операций, то все они могут быть использованы для продолжения хода Процесса (как если бы процесс не содержал каких-либо Шлюзов). Это означает, что:
    - Ход Процесса ДОЛЖЕН продолжиться в момент поступления сигнала (Токена) от какого-либо Потока операций.
        - Импульсы от других Потоков операций, среди которых находится тот, от которого поступил вышеописанный импульс, могут поступить в другой раз, и Процесс также, без промедления и синхронизации данных импульсов, будет продолжен при их поступлении.

Исключающее поведение данного типа Шлюзов, направленное на разделение Потоков операций, заключается в следующем:

- В ходе выполнения Процесса ДОЛЖЕН БЫТЬ выбран лишь один Выход.
    - Выход ДОЛЖЕН БЫТЬ выбран в соответствии целью Потока операций данного Выхода.
        - В случае, если был создан экземпляр цели Потока сообщений(например, было получено сообщение либо истекло указанное время), то ДОЛЖЕН БЫТЬ выбран Выход, а остальные Выходы НЕ ДОЛЖНЫ быть указаны (т.е. их цели будут недействительными).
- Атрибут Condition Исходящего потока операций ДОЛЖЕН иметь значение None.
- Цель Исходящего потока операций Шлюза ДОЛЖНА являться одним из следующих элементов Процесса:
    - Задача, значение атрибута TaskType которой равно Receive.
    - Промежуточное событие, значение атрибута Trigger которого равно Message, Timer, Rule или Link.
        - В случае, если целью Выхода является Задача, то целью для другого Выхода НЕ ДОЛЖНО являться Промежуточное событие, значение атрибута Trigger которого равно Message. Это означает, что сообщения ДОЛЖНЫ поступать только благодаря Задачам типа Получение сообщений или Событиям типа Сообщение, однако, невозможно использования двух указанных элементов Процесса для одного Выхода.

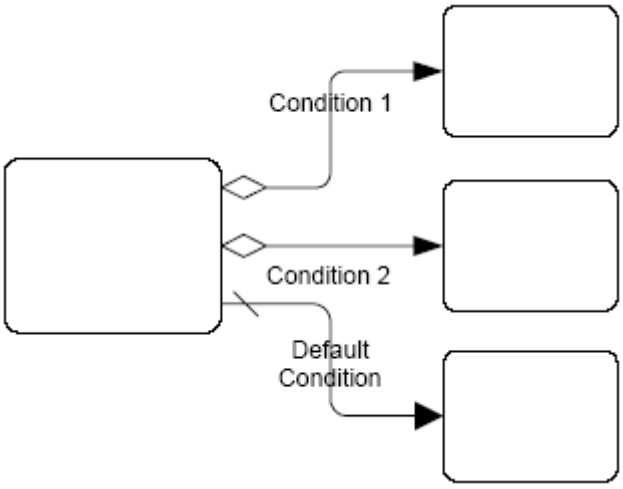
§§ 9.5.3. Неэксклюзивный шлюз (ИЛИ) (Inclusive Gateways (OR))

Данный вид Условий представляет собой точку ветвления, альтернативные маршруты которой зависят от условных выражений, содержащихся в Исходящем потоке операций. Однако в данном случае значение одного из условных выражений, равное True, не исключает определения значения любого другого условного выражения. Токен проходит через все Потоки операций, определенные как True, что напоминает группировку связанных между собой независимых Двойных (Да/Нет) Условий, и может быть смоделировано подобным образом. До тех пор, пока каждый отдельно взятый маршрут является независимым, могут быть задействованы любые комбинации маршрутов, от 0 до бесконечности, однако, комбинация должна содержать не менее одного маршрута.

Примечание: Если ни один из атрибутов ConditionExpressions, относящихся к Неэксклюзивным Шлюзам, не имеет значения True, то модель такого Процесса считается неверной.

Существуют два типа механизмов моделирования ситуаций, задействующих данный тип Шлюзов:

Первый тип механизмов моделирования ситуаций, включающих Неэксклюзивные Условия, не задействует данный тип Шлюзов напрямую, а заключается в создании набора Условных потоков операций и изображается при помощи маленького ромба, представляющего собой совокупность Входов и Выходов, не принадлежащих Шлюзу (см. фигуру 9.23). Условные потоки операций обладают собственными атрибутами: атрибутом Condition, значения которого равно Expression, а также атрибутом ConditionExpression, значение которого представляет собой логическое математическое выражение, основанное на информации, доступной Процессу. Эти Потоки операций отмечаются мини- ромбом в основании линий своих маршрутов.



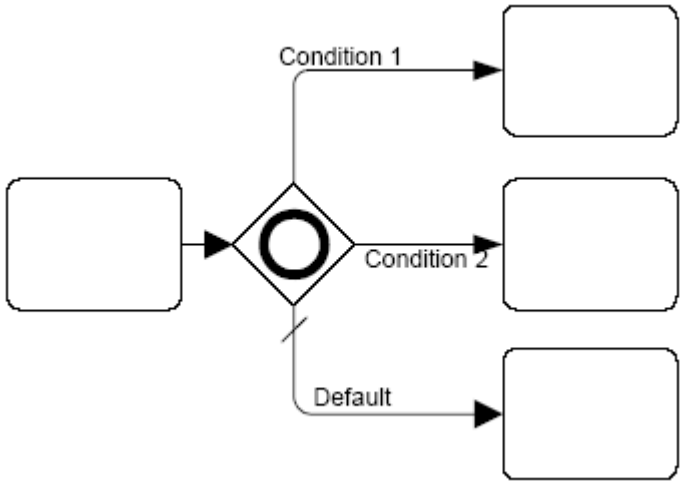
Фигура 9.23 – Неэксклюзивное Условие, Использующее Условные Потоки Операций

Существует набор ограничений использования Условных Потоков операций (в сочетании с мини-ромбами):

- Источником НЕ МОЖЕТ являться Событие, но им МОЖЕТ БЫТЬ Шлюз, однако, на диаграмме НЕ ДОЛЖНЫ БЫТЬ изображены мини-ромбы. Источником Условного потока сообщений МОЖЕТ являться Действие (Задача или Подпроцесс); в данном случае на диаграмме ДОЛЖНЫ изображаться мини-ромбы.
    - Шлюз, использующийся в качестве источника Условного потока операций, НЕ ДОЛЖЕН иметь тип И (Параллельный шлюз).
- В случае, если Условный поток операций начинает свой маршрут от действия, являющегося источником, то к данному действию ДОЛЖЕН БЫТЬ присоединен по меньшей мере один Исходящий поток операций.
    - Дополнительные Потоки операций также МОГУТ быть условными, однако, это условие не является обязательным .

Второй тип механизмов моделирования ситуаций, включающих Неэксклюзивные Условия, основан на использовании Шлюзов ИЛИ (см. фигуру 9.24), которые иногда сочетаются с другими типами Шлюзов. В данном случае маркер помещается в центр графического элемента Шлюза для того, чтобы указать на включающее поведение Шлюза.

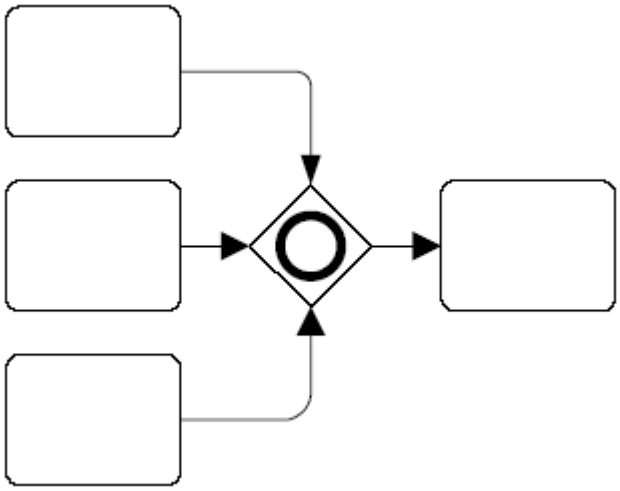
- Неэксклюзивный Шлюз ДОЛЖЕН иметь маркер, изображающийся в виде круга или овала и находящийся внутри графического элемента такого Шлюза (см. фигуру 9.24), для того, чтобы без труда отличить Неэксклюзивный Шлюз от других типов Шлюзов.



Фигура 9.24 - Неэксклюзивное Условие, Использующее Шлюз ИЛИ

Поведение модели, изображенной на фигуре 9.23, является аналогичным поведению модели, изображенной на фигуре 9.24. От разработчика модели зависит то, будет ли хотя бы одно условие иметь значение True при завершении Процесса.

Неэксклюзивный Шлюз, использующийся в качестве Объединителя, будет ждать (синхронизировать) все Токены, созданные ранее. Нет необходимости в том, чтобы все Входящие потоки операций создавали Токены (как это происходит при использовании Параллельного Шлюза), однако, все Потоки операций должны быть созданы предшествующими элементами (например, Эксклюзивным шлюзом ИЛИ). В случае, если Неэксклюзивный Шлюз ИЛИ, созданный ранее, создает два Токена их трех возможных, то другой Неэксклюзивный Шлюз ИЛИ, располагающийся далее, синхронизирует два созданных ранее Токена и не ждет оставшегося Токена, даже если существует три Входящих потока операций (см. фигуру 9.25).



Фигура 9.25 – Неэксклюзивный Шлюз, Объединяющий Потоки Операций

Атрибуты неэксклюзивного шлюза

Таблица 9.28 содержит информацию об атрибутах Неэксклюзивных Шлюзов. Данные атрибуты могут использоваться лишь в тех случаях, когда атрибут GatewayType имеет значение OR. Нижеописанные атрибуты продолжают список общих атрибутов Шлюзов (см. таблицу 9.30):

9.28 - Атрибуты Неэксклюзивных Шлюзов

Атрибуты	Описание
Gates (0-n) : Gate	Неэксклюзивный Шлюз может содержать 0 или более Выходов. Отсутствие Выходов возможно в случае, если Шлюз является конечным объектом в Процессе, не содержащем в своем составе Стартового или Конечного событий. В случае, если Шлюз соединен с 0 или лишь одним Входящим потоком операций (т.е. Шлюз используется как Условие), то в Шлюзе должно быть по меньшей мере два Выхода.
[Gate] **OutgoingSequenceFlow** : SequenceFlow	Каждый Выход ДОЛЖЕН БЫТЬ ассоциирован с Потоком сообщений. Поток сообщений ДОЛЖЕН иметь атрибут Condition, значение которого равно Expression, а также действительное условное выражение. Условное выражение ДОЛЖНО, в свою очередь, быть индивидуальным для каждого Выхода данного Шлюза. Дополнительную информацию об атрибутах Потока операций см. в части 10.1.2, «Sequence Flow», стр. 100. В случае, если Шлюз имеет лишь один Выход (т.е. данный Шлюз выступает в качестве объединяющего элемента), то Поток операций ДОЛЖЕН иметь атрибут Condition, значение которого равно None.
[Gate] **Assigments** (0-n) : Assignment	Для каждого Выхода МОЖЕТ БЫТЬ создано одно или более выражений назначения. Назначение ДОЛЖНО БЫТЬ выполнено тогда, когда выбран Выход. Дополнительную информацию о Назначениях см. в части B.11.1, «Assignment», стр. 268.
DefaultGate (0-1) : Gate	МОЖЕТ БЫТЬ определен Выход по умолчанию.
[Gate] **OutgoingSequenceFlow** : SequenceFlow	В случае, если в Шлюзе определен Выход по умолчанию, то ДОЛЖЕН БЫТЬ и ассоциированный с ним Поток операций. В данном случае Поток операций ДОЛЖЕН иметь индикатор, установленный по умолчанию (см. фигуру 9.24), а также атрибут Condition, значение которого равно Default. Дополнительную информацию об атрибутах Потока операций см. в части 10.1.2, «Sequence Flow», стр. 100.
[Gate] **Assigments** (0-n) : Assignment	Для Выхода, определенного по умолчанию, МОЖЕТ БЫТЬ создано 0 или более назначений. Назначение ДОЛЖНО БЫТЬ выполнено тогда, когда выбран Выход по умолчанию. Дополнительную информацию о Назначениях см. в части B.11.1, «Assignment», стр. 268.

Соединение потока сообщений

Данная часть содержит продолжение описания правил соединения Потока операций основных видов Шлюзов, помещенного в части «Соединение Потока Операций Основных Видов Шлюзов» (стр. 70 в оригинале спецификации). Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций».

Неэксклюзивное поведение данного типа Шлюзов, направленное на объединение Потоков операций, заключается в следующем:

- В случае, если Шлюз соединен с несколькими Входящими потоками операций, то один или более из них могут быть использованы для продолжения хода Процесса (как если бы процесс не содержал каких-либо Шлюзов). Таким образом:
    - Ход Процесса ДОЛЖЕН продолжиться в момент поступления сигналов (Токенов) от всех Входящих потоков операций, ожидающих появления сигнала, основанного на предшествующей структуре Процесса (например, на предшествующем Неэксклюзивном Условии).
        - Некоторые Входящие потоки операций не получат сигналы, а модель, от которой Потоком операций будет получен сигнал, может меняться для разных экземпляров Процесса.

Примечание: Входящие потоки операций, использующие в качестве источников действия, расположенные далее и являющиеся частью цикла, будут обработаны по-другому, нежели те Потоки операций, которые в качестве источников используют предшествующие элементы Процесса. Такие Входящие потоки операций относятся к другому типу Потоков операций, нежели использующие в качестве источников предшествующие действия.

Неэксклюзивное поведение данного типа Шлюзов, направленное на разделение Потоков операций, заключается в следующем:

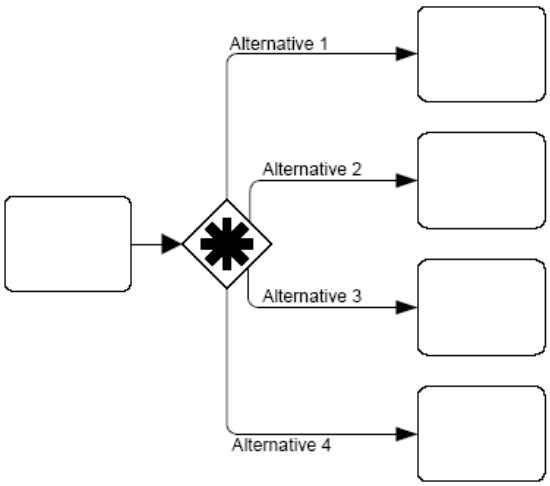
- В ходе выполнения Процесса ДОЛЖНЫ БЫТЬ выбраны один или более Выходов.
    - Выход ДОЛЖЕН БЫТЬ выбран в соответствии с условным выражением, определенным для данного Потока операций, ассоциируемого с данным Выходом.
        - ДОЛЖНО БЫТЬ вычислено значение условия, ассоциируемого со всеми Выходами данного Шлюза.
        - В случае, если значение условия определено как True, то, независимо от того, были ли выбраны или не выбраны какие-либо другие Выходы, ДОЛЖЕН БЫТЬ выбран данный Выход.
        - Если для данного Выхода ни одно из значений условных выражений не определено как True, то ДОЛЖЕН БЫТЬ выбран Выход, установленный по умолчанию.

§§ 9.5.4. Комплексный шлюз (Complex Gateway)

Комплексные Шлюзы используются в BPMN для управления ситуациями, которые с трудом поддаются управлению другими видами Шлюзов. Комплексные Шлюзы также могут использоваться для объединения нескольких простых соединенных Шлюзов в один составной. Разработчики моделей могут использовать составные выражения, определяющие объединяющее и/или разделяющее поведение данного Шлюза.

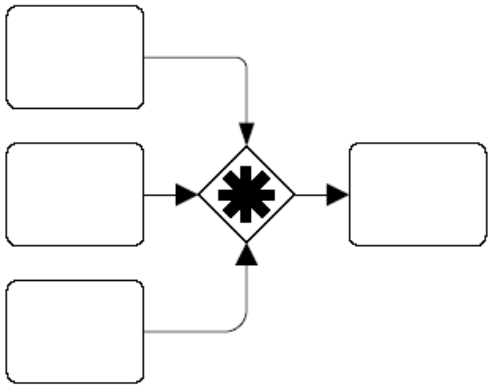
- Составной Шлюз ДОЛЖЕН иметь маркер, изображающийся в виде небольшой звездочки и расположенный внутри графического элемента Шлюза (см. фигуру 9.26), для того, чтобы без труда отличить данный вид Шлюзов от других.

При использовании Шлюза в качестве Условия (см. фигуру 9.26) выражение определяет то, какой из Исходящих потоков операций будет выбран для продолжения данного Процесса. Выражение может ссылать на данные Процесса, а также на статус Входящего потока операций. Например, выражение может оценивать данные Процесса, а затем, согласно результатам оценки, выбирать различные Исходящие потоки операций. Однако выражение должно быть таким, чтобы мог быть выбран по меньшей мере один Исходящий поток операций.



Фигура 9.26 – Комплексное Условие (Шлюз)

В случае, если Шлюз используется в качестве Объединителя (см. фигуру 9.27), то используется выражение, определяющее, какой из Входящих потоков операций понадобится для продолжения Процесса. Данное выражение может ссылаться на данные Процесса, а также на статус Входящего потока операций. Например, выражение может указывать на то, какие три входящих Токена из пяти предлагаемых будут использованы для продолжения Процесса. Выражение также может указывать на то, что для продолжения Процесса требуется Токен Потока операций «а», хотя Токены Потоков операций «б» и «в» также являются приемлемыми. Однако выражение должно быть таким, чтобы на данном отрезке Процесс не приостанавливался.



Фигура 9.27 – Комплексное Объединение (Шлюз)

Атрибуты комплексного шлюза

Таблица 9.29 содержит информацию об атрибутах Комплексных Шлюзов. Данные атрибуты могут использоваться лишь в тех случаях, когда атрибут GatewayType имеет значение Complex. Нижеописанные атрибуты продолжают список общих атрибутов Шлюзов (см. таблицу 9.30):

9.29 - Атрибуты Комплексных Шлюзов

Атрибуты	Описание
Gates (0-n) : Gate	Комплексный Шлюз может содержать 0 или более Выходов. Отсутствие Выходов возможно в случае, если Шлюз является конечным объектом в Процессе, не содержащем в своем составе Стартового или Конечного событий. В случае, если в Процессе содержится 0 или лишь один Входящий поток операций (т.е. Шлюз используется как Условие), то в Шлюзе должно быть по меньшей мере два Выхода.
[Gate] **OutgoingSequenceFlow** : SequenceFlow	Каждый Выход ДОЛЖЕН БЫТЬ ассоциирован с Потоком сообщений. Поток сообщений ДОЛЖЕН иметь атрибут Condition, значение которого равно None. Дополнительную информацию об атрибутах Потока операций см. в части 10.1.2, «Sequence Flow», стр. 100. В случае, если Шлюз имеет лишь один Выход (т.е. данный Шлюз выступает в качестве объединяющего элемента), то Поток операций ДОЛЖЕН иметь атрибут Condition, значение которого равно None.
[Gate] **Assigments** (0-n) : Assignment	Для каждого Выхода МОЖЕТ БЫТЬ создано одно или более выражений назначения. Назначение ДОЛЖНО БЫТЬ выполнено тогда, когда выбран Выход. Дополнительную информацию о Назначениях см. в части B.11.1, «Assignment», стр. 268.
IncomingCondition (0-1) : Expression	В случае, если Шлюз соединен с несколькими Входящими потоками операций, то инструментом моделирования ДОЛЖНО БЫТЬ использовано выражение IncomingCondition, представляющее собой выражение, ссылающееся на названия Потоков операций и/или свойства (данные) Процесса.
OutgoingCondition (0-1) : Expression	В случае, если Шлюз соединен с несколькими Исходящими потоками операций, то инструментом моделирования ДОЛЖНО БЫТЬ использовано выражение OutgoingCondition, представляющее собой выражение, ссылающееся на идентификаторы (исходящих) Потоков операций и/или свойства (данные) Процесса.

Соединение потока операций

Данная часть содержит продолжение описания правил соединения Потока операций основных видов Шлюзов, помещенного в части «Соединение Потока Операций Основных Видов Шлюзов» (стр. 70 в оригинале спецификации). Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций».

Комплексная сущность данного типа Шлюзов, направленная на объединение Потоков операций, заключается в следующем:

- В случае, если Шлюз соединен с несколькими Входящими потоками операций, то все они могут быть использованы для продолжения хода Процесса. Точная комбинация Входящих потоков операций определяется выражением IncomingCondition данного Шлюза.
    - Процесс ДОЛЖЕН быть продолжен тогда, когда от соответствующего Входящего потока операций поступит необходимое количество сигналов (Токенов).
    - От других Потоков операций Процесса также МОГУТ поступать сигналы, однако, они НЕ ДОЛЖНЫ использоваться для продолжения Процесса.

Примечание: Входящие потоки операций, использующие в качестве источников действия, расположенные далее и являющиеся частью цикла, будут обработаны по-другому, нежели те Потоки операций, которые в качестве источников используют предшествующие элементы Процесса. Такие Входящие потоки операций относятся к другому типу Потоков операций, нежели использующие в качестве источников предшествующие действия.

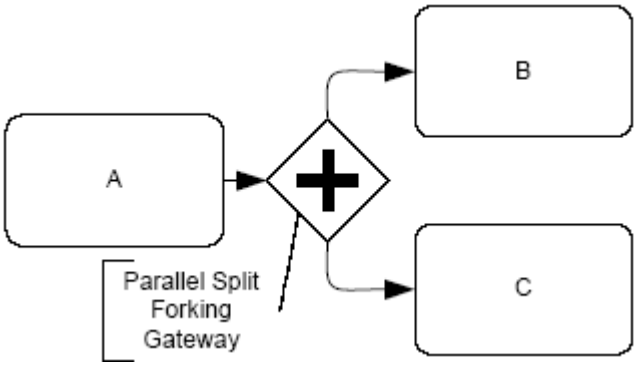
Включающее поведение данного типа Шлюзов, направленное на разделение Потоков операций, заключается в следующем:

- В ходе выполнения Процесса для данного Шлюза ДОЛЖНЫ БЫТЬ выбраны один или несколько Выходов.
- Выходы ДОЛЖНЫ БЫТЬ выбраны в соответствии с выражением OutgoingCondition данного Шлюза.

§§ 9.5.5. Параллельный шлюз (И) (Parallel Gateway(AND))

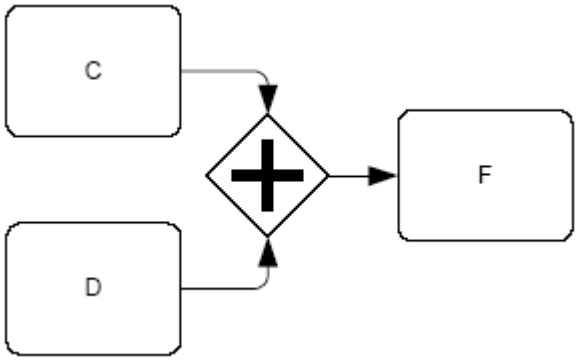
Параллельный Шлюз представляет собой механизм для синхронизации параллельных Потоков операций. Наличие Параллельных Шлюзов не является обязательным условием для создания параллельных Потоков операций, однако, такие Шлюзы помогают более прозрачно изображать развитие сложных ситуаций, где используется несколько Шлюзов и необходимо наличие нескольких параллельных Потоков операций. Разработчики моделей могут пожелать использовать мировой передовой опыт, где Параллельные Шлюзы всегда используются для создания параллельных маршрутов. Это будет способствовать созданию дополнительного составного элемента моделирования, который предоставит сбалансированный метод объединения в пары разделяющихся и соединяющихся элементов.

- Параллельный Шлюз ДОЛЖЕН иметь маркер, изображающийся в виде знака «+» и помещенный в графический элемент Шлюза (см. фигуру 9.28), для того, чтобы без труда отличить данный вид Шлюзов от других.



Фигура 9.28 – Параллельный Шлюз

Параллельные Шлюзы используются для синхронизации параллельных маршрутов.



Фигура 9.29 – Объединение Параллельных Маршрутов

Атрибуты параллельного шлюза (И)

Таблица 9.30 содержит информацию об атрибутах Параллельных Шлюзов. Данные атрибуты могут использоваться лишь в тех случаях, когда атрибут GatewayType имеет значение AND (Parallel). Нижеописанные атрибуты продолжают список общих атрибутов Шлюзов (см. таблицу 9.31):

9.30 - Атрибуты Комплексных Шлюзов

Атрибуты	Описание
Gates (0-n) : Gate	Параллельный Шлюз может содержать 0 или более Выходов. Отсутствие Выходов возможно в случае, если Шлюз является конечным объектом в Процессе, не содержащем в своем составе Стартового или Конечного событий. В случае, если Шлюз соединен с 0 или лишь одним Входящим потоком операций (т.е. Шлюз используется как разделение), то в Шлюзе должно быть по меньшей мере два Выхода.
[Gate] **OutgoingSequenceFlow** : SequenceFlow	Каждый Выход ДОЛЖЕН БЫТЬ ассоциирован с Потоком операций. Поток операций ДОЛЖЕН иметь атрибут Condition, значение которого равно None. Дополнительную информацию об атрибутах Потока операций см. в части 10.1.2, «Sequence Flow», стр. 100.
[Gate] **Assigments** (0-n) : Assignment	Для каждого Выхода МОЖЕТ БЫТЬ создано одно или более выражений назначения. Назначение ДОЛЖНО

	БЫТЬ выполнено тогда, когда выбран Выход. Дополнительную информацию о Назначениях см. в части B.11.1, «Assignment», стр. 268.

Соединение потока сообщений

Данная часть содержит продолжение описания правил соединения Потока операций основных видов Шлюзов, помещенного в части «Соединение Потока Операций Основных Видов Шлюзов» (стр. 70 в оригинале спецификации). Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут служить источниками и целями Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций».

Параллельность данного типа Шлюзов, направленная на объединение Потоков операций, заключается в следующем:

- В случае, если Шлюз соединен с несколькими Входящими потоками операций, то все они могут быть использованы для продолжения хода Процесса, который будет синхронизирован. Таким образом:
    - Процесс ДОЛЖЕН быть продолжен тогда, когда от всех Входящих потоков операций поступят сигналы (Токены), т.е. Процесс будет находиться в ожидании всех сигналов, необходимых для продолжения его хода.

Примечание: Входящие потоки операций, использующие в качестве источников действия, расположенные далее и являющиеся частью цикла, будут обработаны по-другому, нежели те Потоки операций, которые в качестве источников используют предшествующие элементы Процесса. Такие Входящие потоки операций относятся к другому типу Потоков операций, нежели использующие в качестве источников предшествующие действия.

Параллельность данного типа Шлюзов, направленная на разделение Потоков операций, заключается в следующем:

- В ходе выполнения Процесса ДОЛЖНЫ БЫТЬ выбраны все Выходы.

§ 9.6. Зоны Ответственности: пулы и дорожки (Swimlanes: Pools and Lanes)

Язык BPMN предлагает бóльшие возможности, нежели BPEL4WS, и эти возможности могут применяться в разных аспектах. Аспект, о котором поведется речь в данной части, затрагивает бизнес-процессы в корпоративной среде с отношениями «бизнес для бизнеса» (B2B). Спецификация BPMN используется понятие «зона ответственности» для разделения и/или организации действий.

BPEL4WS акцентирует внимание на особых внутренних бизнес-процессах, являющихся внутренними для данного Участника (компании или организации), а также определяет абстрактные процессы с точки зрения отдельно взятого Участника. Что касается диаграмм BPMN, то они могут включать в себя более одного внутреннего бизнес-процесса, а также процессы, указывающие на сотрудничество между внутренними бизнес-процессами и Участниками. В данном случае, любой внутренний бизнес-процесс будет считаться процессом, выполняемым разными Участниками. На схеме каждый Участник отделен от

другого и представляет собой прямоугольник, называемый Пулом (Pool). Пулы могут быть разделены на Дорожки (Lanes).

Часть 7.1.1 «Использование BPMN» (стр. 10 оригинала спецификации) содержит описание вариантов использования BPMN для моделирования внутренних бизнес-процессов, а также взаимодействия между процессами в моделях типа «бизнес для бизнеса». Пулы и Дорожки направлены на реализацию данных вариантов использования BPMN.

§§ 9.6.1. Общие атрибуты зон ответственности

Таблица 9.31 содержит информацию об атрибутах Зон ответственности (Пулов и Дорожек), продолжающих список общих атрибутов графических элементов (таблица 9.1):

Таблица 9.31 – Общие Атрибуты Зон Ответственности

Атрибуты	Описание
Name : String	Атрибут Name представляет собой текстовое описание Зоны ответственности.

§§ 9.6.2. Пул (Pool)

Пул представляет собой Участника Процесса. Такой Участник может быть как одной из заинтересованных сторон (например, компанией), так и играть более общую бизнес-роль (например, покупатель, продавец, производитель). На диаграмме Пул изображается в виде контейнера для отделения Процесса от других Пулов при моделировании ситуаций типа «бизнес для бизнеса», хотя нет необходимости добавления в Пул каких-либо внутренних деталей (т.е. Пул может быть использован как «черный ящик»).

- Пул представляет собой прямоугольник, который ДОЛЖЕН БЫТЬ выполнен жирной, черной, одиночной линией (как показано на фигуре 9.30).
    - Текст, цвет, размер, а также линии, используемые для изображения Конечного события, должны соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм» (стр. 26 оригинала спецификации).

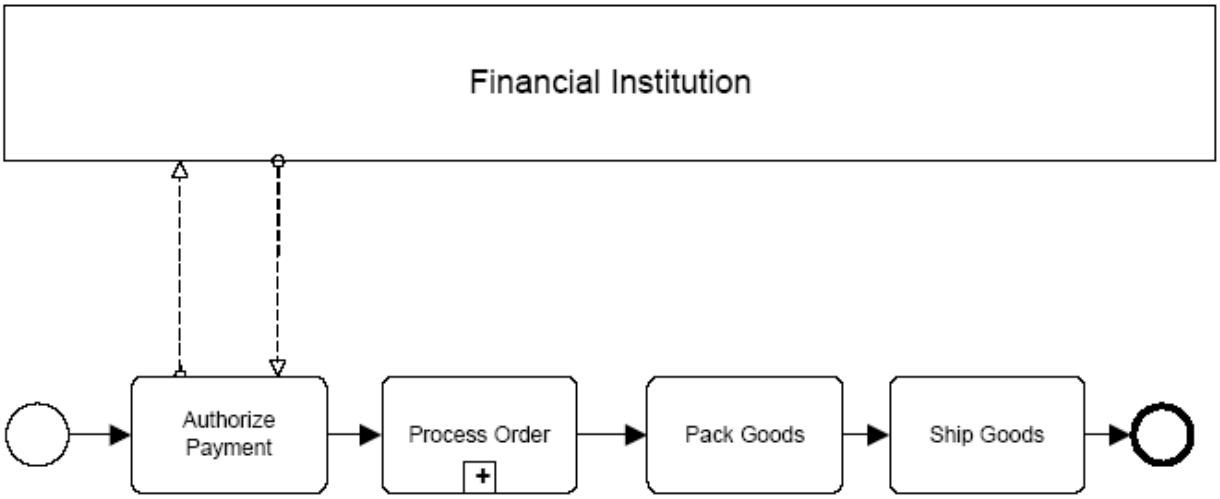
Фигура 9.30 – Пул

Для того, чтобы сделать диаграмму более прозрачной, Пул простирается на всю длину данной Диаграммы как горизонтально, так и вертикально. Однако особых ограничений, касающихся размера и/или расположения Пула, в спецификации BPMN не существует в интересах сохранения «недвижимости» Диаграммы на экране или напечатанной странице. В интересах сохранения «недвижимости» Диаграммы на экране или напечатанной странице использование Пулов (или Дорожек) инструментами моделирования является достаточно гибким.

Пул выступает в роли контейнера Потока операций между действиями. Поток операций может пересекать границы Дорожек, расположенных внутри Пула, однако, не имеет право

пересекать границы самого Пула. Взаимодействие между Пулами, что происходит, например, в ситуациях типа «бизнес для бизнеса», изображается при помощи Потока операций.

Другая сторона использования Пулов зависит от того, содержится ли в Процессе действие, детальные данные которого находятся в Пуле. Таким образом, данный Пул может быть как «прозрачным ящиком», где видны все детали действия, так и «черным ящиком», где все детали скрыты. Пул, представляющий собой «черный ящик», не имеет ассоциированных Потоков операций, однако, к его границе может быть присоединен Поток сообщений (см. таблицу 9.32).



Фигура 9.32 – Основной (Внутренний) Пул без Границ

Атрибуты пула

Таблица 9.33 содержит информацию об атрибутах Пула, продолжающих список общих атрибутов Зон ответственности (см. таблицу 9.34):

Таблица 9.33 - Атрибуты Пула

Атрибут	Описание
Process (0-1) : Process	Атрибут Process указывает на Процесс, содержащийся в Пуле. Каждый Пул МОЖЕТ содержать Процесс. Дополнительную информацию об атрибутах Процесса см. в части 8.6, «Processes», стр. 29 оригинала спецификации.
Participant : Participant	Разработчик модели ДОЛЖЕН указать Участника Пула. Участником может быть как Роль, так и Заинтересованная сторона. Данный атрибут указывает на то, какую роль будет выполнять данная Заинтересованная сторона или Роль в сотрудничестве, включенном в Диаграмму. Дополнительную информацию об атрибутах Участника см. в части B.11.6, «Participant», стр. 270 оригинала спецификации.
Lanes (1-n) : Lane	Внутри Пула ДОЛЖНЫ находиться одна или две Дорожки. В случае, если Пул содержит лишь одну Дорожку, то данная Дорожка имеет то же название,

	что и Пул. В данном случае изображается только название самого Пула. В случае, если Пул содержит больше одной Дорожки, то каждой Дорожке должно быть присвоено имя. В данном случае отображаться должны имена всех Дорожек. Дополнительную информацию об атрибутах Дорожки см. в части 9.6.3, «Lane», стр. 90 оригинала спецификации.
BoundaryVisible True : Boolean	Атрибут BoundaryVisible указывает на то, являются ли границы графического элемента Пула видимыми. Среди всех элементов диаграммы лишь Пул МОЖЕТ иметь атрибут BoundaryVisible со значением False.

§§ 9.6.3. Дорожка (Lane)

Дорожка представляет собой подразделение внутри Пула и простирается на всю длину данного Пула как горизонтально, так и вертикально (см. фигуру 9.33). По желанию разработчика модели текст, относящийся к Дорожке (например, название Дорожки или какой-либо атрибут), может быть помещен в любом месте данной Дорожки и располагаться в любом направлении. В представленном ниже примере название Дорожки располагается в виде баннера в левой части (в горизонтальном Пуле) или вверху (в вертикальном Пуле) рядом с линией, отделяющей название Дорожки от названия Пула. Однако такое размещение текста не является обязательным.

Фигура 9.33 – Пул, Содержащий Две Дорожки

Дорожки используются для организации и категоризации действий, расположенных внутри данного Пула. Содержание Дорожек зависит от разработчика моделей. Спецификация BPMN не указывает на то, как должны быть использованы Дорожки, которые часто используются в качестве внутренних ролей (например, Менеджер, Партнер), систем (например, прикладные системы предприятия), внутренних отделов (например, поставки, бюджет) и т.д. В добавок Дорожки могут быть вложены или установлены в матрице. Например, может существовать внешнее множество Дорожек отделов предприятия, а также внутреннее множество Дорожек для ролей, существующих внутри каждого отдела.

Атрибуты дорожки

Таблица 9.34 содержит информацию об атрибутах Дорожки, продолжающих список общих атрибутов Зон ответственности (см. таблицу 9.34):

Таблица 9.34 - Атрибуты Дорожки

Атрибут	Описание
ParentPool : Pool	ДОЛЖЕН БЫТЬ указан родительский Пул, причем Дорожка может иметь лишь один родительский Пул. Дополнительную информацию об атрибутах Пула см. в части 9.6.2, «Pool», стр. 87 оригинала спецификации.
ParentLane (0-1) : Lane	Атрибут ParentLane является опциональным и используется в случае, если одна Дорожка вложена в другую Дорожку. Такая вложенность может быть многоуровневой, однако, указывается лишь предшествующая родительская Дорожка.

§ 9.7. Артефакты (Artifacts)

Язык BPMN позволяет разработчикам моделей указывать дополнительную информацию о Процессе, не связанную непосредственно с Потоками операций или Потоками сообщений данного Процесса.

BPMN предлагает использование трех стандартных Артефактов: Объект данных, Группа и Аннотация. В более позднюю версию спецификации BPMN могут быть добавлены дополнительные виды Артефактов. Разработчик модели или инструмент моделирования могут расширить диаграмму бизнес-процесса путем добавления новых видов Артефактов в данную диаграмму. Все новые виды Артефактов должны удовлетворять требованиям соединения Потоков операций и Потоков сообщений (см. ниже). Ассоциация может быть использована для соединения Артефактов с Объектами данных (см. часть 10.1.4, «Association», стр. 105 оригинала спецификации).

§§ 9.7.1. Общие понятия артефактов

Данная часть содержит информацию о понятиях, общих для всех видов Артефактов.

Общие атрибуты артефактов

Таблица 9.35 содержит информацию об общих атрибутах Артефактов, продолжающих список общих атрибутов графических элементов BPMN (см. таблицу 9.34):

Таблица 9.35 – Общие Атрибуты Артефактов

Атрибут	Описание
ArtifactType (DataObject | Group | Annotation) DataObject : String	Атрибут ArtifactType МОЖЕТ иметь значения DataObject, Group, а также Annotation. Список значений данного атрибута МОЖЕТ быть расширен путем добавления новых значений.
Pool (0-1) : Pool	Для определения положения Пула МОЖЕТ БЫТЬ добавлен Атрибут Pool. Такой Артефакт, как Аннотация, может находиться за пределами границ Пулов. Группа может простираться на длину нескольких Пулов. Дополнительную информацию об атрибутах Пула см. в части 9.6.2, «Pool», стр. 87 оригинала спецификации.
Lanes (0-n) : Lane	В случае, если указан Пул, содержащий больше одной Дорожки, то ДОЛЖНЫ БЫТЬ добавлены

	названия Дорожек. МОГУТ БЫТЬ перечислены несколько Дорожек. Дополнительную информацию об атрибутах Дорожки см. в части 9.6.3, «Lane», стр. 90 оригинала спецификации.

Соединение потоков операций артефактов

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут являться объектами Потока операций, обратитесь к пункту 8.4.1 «Правила Соединения Потоков Операций» (стр. 27 оригинала спецификации).

- Артефакт НЕ ДОЛЖЕН являться целью Потока операций.
- Артефакт НЕ ДОЛЖЕН являться источником Потока операций.

Соединение потоков сообщений артефактов

Для того, чтобы увидеть полный список графических элементов и узнать, каким образом они могут являться объектами Потока сообщений, обратитесь к пункту 8.4.2 «Правила Соединения Потоков Сообщений» (стр. 28 оригинала спецификации).

- Артефакт НЕ ДОЛЖЕН являться целью Потока сообщений.
- Артефакт НЕ ДОЛЖЕН являться источником Потока сообщений.

§§ 9.7.2. Объект данных (Data Object)

В спецификации BPMN Объект данных относится к Артефактам, а не к Элементам потока. Объект данных относится к Артефактам потому, что он не оказывает непосредственного влияния на Поток операций или Поток сообщений, присутствующих в Процессе, однако, содержит сведения о данном Процессе, заключающиеся в описании того, какие документы, сведения или какие-либо другие объекты используются и дополняются в ходе выполнения Процесса. Т.к. название «Объект данных» может подразумевать электронный документ, то Объекты данных могут быть использованы для представления различных типов объектов, как электронных, так и физических.

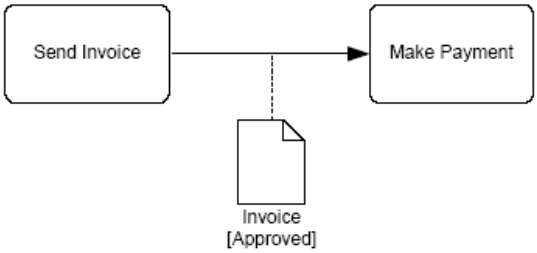
В целом спецификация BPMN не занимается стандартизацией многих Артефактов, использующихся в моделировании диаграмм. Создание Объектов данных главным образом зависит от разработчиков моделей для удовлетворения их требований. Однако эквиваленты Объектов данных, использующихся в BPMN, используются в технологических процессах, ориентированных на управление документооборотом, а также во многих других методологиях моделирования. Таким образом, Объект данных используется настолько часто, что представляется важным стандартизировать его изображение и поведение.

- Объект данных представляет собой вертикально расположенный прямоугольник с отогнутым книзу правым верхним углом. Прямоугольник ДОЛЖЕН БЫТЬ выполнен жирной, черной, одиночной линией (см. фигуру 9.34).
    - Текст, цвет, размер, а также линии, используемые для изображения Конечного события, должны соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм» (стр. 26 оригинала спецификации).

Фигура 9.34 – Объект Данных

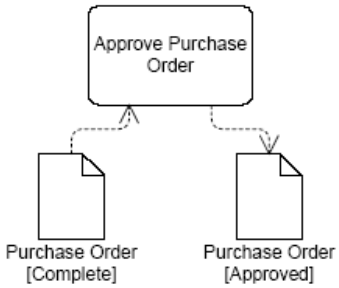
Как все Артефакты, Объект данных, как правило, ассоциируется с Элементами потока. Ассоциация используется для соединения Объекта данных и Элементов потока. Это означает, что разработчики моделей, желающие упростить диаграмму, могут моделировать поведение Процесса без использования Объектов данных. Что касается разработчиков моделей, стремящихся включить в диаграмму как можно больше информации, не прибегая при этом к изменению основного поведения Процесса, то при моделировании данного Процесса они могут использовать Объекты данных.

В отдельных случаях на диаграмме изображается Объект данных, передаваемый от одного действия другому посредством Потока операций (см. фигуру 9.35). Объекты данных также ассоциируются с Потоками сообщений. Они не смешиваются непосредственно с самими сообщениями, однако, могут рассматриваться в качестве дополнительной информации или содержимого некоторых сообщений.



Фигура 9.35 – Объект Данных, Ассоциированный с Потоком Операций

Случается, что один и тот же Объект данных изображается в качестве входа, а затем и выхода Процесса (см. фигуру 9.36). Направленность вместе с Ассоциацией указывают, будет ли являться данный Объект данных входом или выходом. Значение атрибута Объекта данных также может указывать на воздействие Процесса на Объект данных.



Фигура 9.36 – Объект данных, выступающий в качестве входа и выхода

Атрибуты объекта данных

Таблица 9.36 содержит информацию об атрибутах Объектов данных, продолжающих список общих атрибутов Артефактов (см. таблицу 9.35). Данные атрибуты могут использовать лишь в случае, если атрибут ArtifactType имеет значение DataObject:

Таблица 9.36 –Атрибуты Объектов Данных

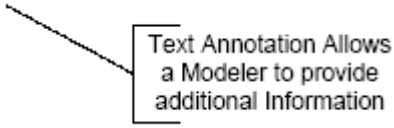
Атрибут	Описание
Name : String	Атрибут Name представляет собой текстовое описание графического элемента.
State (0-1) : String	Атрибут State является опциональным и указывает на влияние, оказанное Процессом на Объект данных. Несколько Объектов данных, имеющих одно и то же название, МОГУТ иметь один и тот же статус в Процессе.
Properties (0-n) : Property	В Объект данных МОГУТ БЫТЬ добавлены свойства, определенные разработчиком модели. Полное изображаемое имя этих свойств выглядит следующим образом: <process name>.<task name>.<property name> (например, «Add Customer.Credit Report.Score»). Дополнительную информацию об определении Свойств см. в части B.11.7, «Property», стр. 270 оригинала спецификации.
RequiredForStart True : Boolean	Значение атрибута RequiredForStart по умолчанию равно True. Это означает, что для запуска действия необходимы входные данные. В случае, если установленное значение равно False, то действие может начаться во входе, однако, оно может иметь вход (более, чем один раз) после того, как действие началось.
ProducedAtCompletion True : Boolean	Значение атрибута ProducedAtCompletion по умолчанию равно True. Это означает, что при завершении действия будут созданы выходные данные. В случае, если установленное значение равно False, то действие МОЖЕТ создать выходные данные (более, чем один раз) до своего завершения.

§§ 9.7.3. Текстовая аннотация (Text Annotation)

Текстовая аннотация представляет собой механизм добавления разработчиком модели дополнительной информации на диаграмму BPMN.

- Текстовая аннотация представляет собой прямоугольник, который ДОЛЖЕН БЫТЬ выполнен жирной, черной, одиночной линией (как показано на фигуре 9.37).
    - Текст, цвет, размер, а также линии, используемые для изображения Конечного события, должны соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм» (стр. 26 оригинала спецификации).

Графический элемент Текстовой аннотации может быть соединен с элементом диаграммы, называемым Ассоциацией (см. фигуру 9.37), однако, не может оказывать влияние на ход Процесса. Текст, ассоциированный с Аннотацией, может быть помещен в пределах открытого прямоугольника.



Фигура 9.37 – Текстовая Аннотация

Атрибуты аннотации

Таблица 9.37 содержит информацию об атрибутах Аннотации, продолжающих список общих атрибутов Артефактов (см. таблицу 9.35). Данные атрибуты могут использовать лишь в случае, если атрибут ArtifactType имеет значение Annotation:

Таблица 9.37 –Атрибуты Аннотации

Атрибут	Описание
Text : String	Атрибут Text представляет собой текст, который разработчик модели желает сообщить читателю диаграммы.

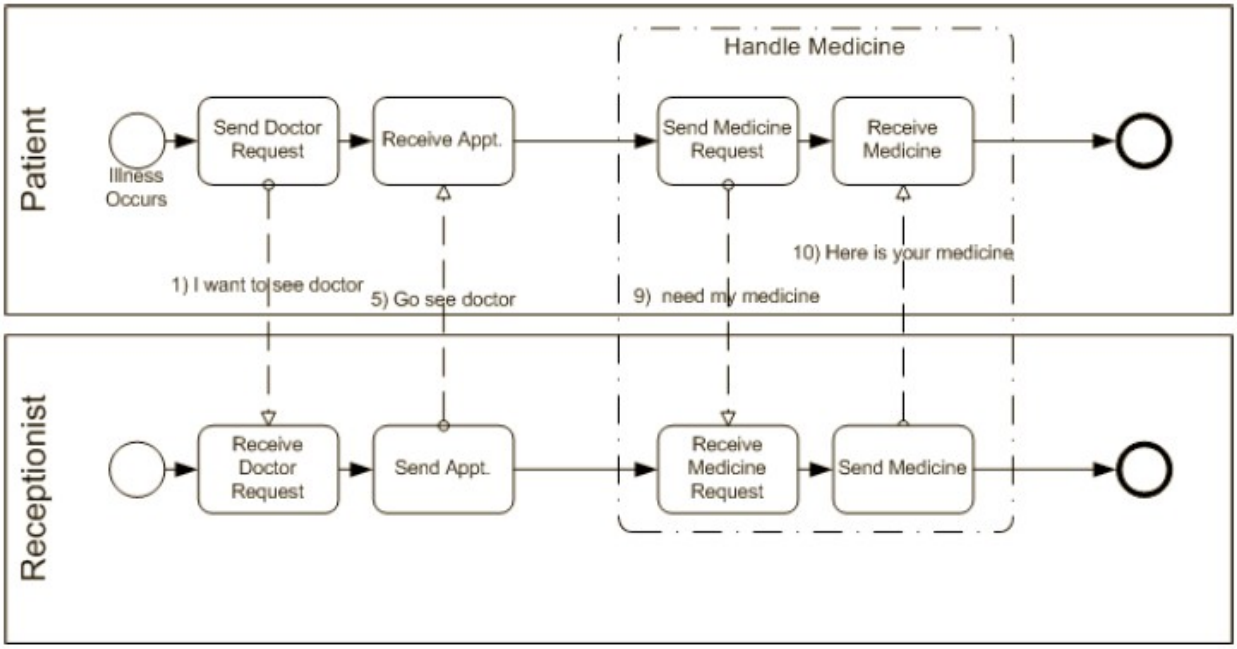
§§ 9.7.4. Группа (Group)

Группа представляет собой Артефакт, предоставляющий визуальный механизм для свободной группировки элементов графических элементов Процесса.

- Группа представляет собой прямоугольник с закругленными углами, который ДОЛЖЕН БЫТЬ выполнен жирной, черной, пунктирной линией (как показано на фигуре 9.38).
    - Текст, цвет, размер, а также линии, используемые для изображения Конечного события, должны соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм» (стр. 26 оригинала спецификации).

Фигура 9.38 – Артефакт «Группа»

Как все Артефакты, Группа не является Действием или Элементом потока, и, следовательно, не может соединяться с Потоком операций или Потоком сообщений. В добавок к этому, Группы не ограничены рамками Пулов и Дорожек (см. фигуру 9.39). Это означает, что Группа может пересекать границы Пула для того, чтобы окружить элементы диаграммы (см. фигуру 9.39), и часто для того, чтобы идентифицировать действия, включенные в распределенную транзакицию типа «бизнес для бизнеса».



Фигура 9.39 – Группа, объединяющая действия разных Пулов

Группы часто используются для выделения определенных частей диаграммы без добавления дополнительных ограничений на выполнение действий, как это было бы в Подпроцессе. Выделенные (сгруппированные) элементы диаграммы могут быть разделены в целях отчетности и анализа. Группы не оказывают влияния на ход Процесса и не соответствуют никаким элементам BPEL4WS.

Атрибуты группы

Таблица 9.38 содержит информацию об атрибутах Группы, продолжающих список общих атрибутов Артефактов (см. таблицу 9.35). Данные атрибуты могут использовать лишь в случае, если атрибут ArtifactType имеет значение Group:

Таблица 9.37 –Атрибуты Аннотации

Атрибут	Описание
Name (0-1) : String	Атрибут Name является опциональным и представляет собой текстовое описание Группы.

Глава 10. Объекты соединения диаграммы бизнес-процесса

Данная часть содержит информацию о графических элементах, используемых для соединения двух объектов (т.е. являющихся соединяющими линиями на диаграмме), а также о том, каким образом эти линии проходят в Процессе (т.е. проходят ли они в соответствии с правильным ходом событий либо разделяются на параллельные или альтернативные маршруты).

§ 10.1. Графические элементы соединения (Graphical Connecting Objects)

Спецификация BPMN выделяет два вида графических элементов соединения: Поток операций/сообщений и Ассоциация. Поток операций и Поток сообщений в некоторой степени затрагивают ортогональный аспект бизнес-процессов, изображаемых на диаграммах, хотя оба этих графических элемента оказывают влияние на выполнение

действий, включенных в Процесс. Поток операций движется, как правило, в одном направлении (либо слева направо, либо сверху вниз), а Поток сообщений - под углом в 90° по отношению к Потоку операций. Такая организация движения помогает изображать отношения между графическими элементами более прозрачно на диаграмме, содержащей Поток операций и Поток сообщений. Однако данная спецификация не настаивает на такого рода отношениях между двумя этими потоками. Разработчик модели может направить потоки в любом направлении к любой точке диаграммы.

Следующие три раздела спецификации содержат описание того, как функционируют эти виды соединения в BPMN.

§§ 10.1.1. Общие атрибуты элементов соединения

Таблица 10.1 содержит информацию об атрибутах, общих для Элементов соединения (Поток операций, Поток сообщений, Ассоциация), продолжающих список общих атрибутов графических элементов (см. таблицу 9.1):

Таблица 9.34 – Общие Атрибуты Элементов Соединения

Атрибут	Описание
Name (0-1) : String	Атрибут Name является опциональным и представляет собой текстовое описание Элемента соединения.
Source : Object	Атрибут Source указывает на то, к какому исходному Элементу потока присоединен Элемент соединения. Примечание: существуют ограничения, касающиеся элементов, к которым могут быть присоединены Поток операций или Поток сообщений. Для Элементов потока, Зон ответственности и Артефактов обратитесь к разделам, посвященным соединению Потоков операций и соединению Потоков сообщений.
Target : Object	Атрибут Target указывает на то, к какому Элементу потока присоединен Элемент соединения. Примечание: существуют ограничения, касающиеся элементов, к которым Поток операций или Поток сообщений присоединены быть не могут. Для Элементов потока, Зон ответственности и Артефактов обратитесь к разделам, посвященным соединению Потоков операций и соединению Потоков сообщений.

§§ 10.1.2. Поток операций (Sequence Flow)

Поток операций служит для отображения того порядка, в котором выполняются действия Процесса. Любой Поток операций имеет лишь один источник и одну цель. Источниками и целями потоков операций могут являться следующие Элементы потока: События (Стартовое, Промежуточное, Конечное), Действия (Задача, Подпроцесс) и Шлюзы. В ходе выполнения (моделирования) Процесса от исходного Элемента потока отходит Токен, который пересекает Поток операций и достигает целевого Элемента потока.

- Поток операций представляет собой стрелку с массивным концом, линия которой ДОЛЖНА БЫТЬ одиночной и жирной (как показано на фигуре 10.1).
- Текст, цвет, размер, а также линии, используемые для изображения Конечного события, должны соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм» (стр. 26 оригинала спецификации).

Фигура 10.1 – Поток операций

Спецификация BPMN не применяет термин «Контрольный поток операций» в отношении линий, относящихся к Потокам операций или Потокам сообщений. Начало действия контролируется не только Потоком операций (порядком выполнения действий), но также и Потоком сообщений (поступающим сообщением), и другими факторами развития Процесса, такими, как запланированные источники. Артефакты могут быть ассоциированы с Действиями для отображения других факторов развития Процесса. Таким образом, спецификация BPMN использует более узкий термин – «Поток операций», т.к. вышеописанные линии главным образом иллюстрируют тот порядок, в котором будут выполнены действия Процесса.

- Поток операций МОЖЕТ содержать атрибут условного выражения, зависящий от элемента, являющегося источником.

Это означает, что значение данного условного выражения должно быть вычислено до того, как будет создан Токен, который впоследствии покинет элемент, являющийся источником, для того, чтобы пересечь Поток операций. Условия обычно бывают ассоциированы со Шлюзами-Условиями, однако, могут быть использованы и для действий.

- В случае, если источником Потока операций является действие, а не Шлюз, то в начале Потока операций ДОЛЖЕН быть изображен Условный Маркер (Conditional Marker), изображаемый в качестве мини-ромба (см. фигуру 10.2).

Такой мини-ромб используется для того, чтобы приблизить поведение Потока операций к поведению Шлюза (также изображаемого в качестве ромба), который осуществляет контроль над Потоками операций, входящими в состав Процесса. Дополнительную информацию об использовании Условных потоков операций см. в разделе «Разделительный Поток Операций», стр. 115 оригинала спецификации.

Фигура 10.2 – Условный Поток Операций

Поток операций, источником которого являются Эксклюзивный Шлюз, основанный на данных, или действие, также может быть определен посредством условного выражения, значение которого равно Default. Такой Поток операций будет изображен с маркером, указывающим на то, что данный Поток операций является Потоком операций по умолчанию.

- Маркер Потока операций по умолчанию ДОЛЖЕН изображаться в виде наклонной косой черты, расположенной у основания линии данного Потока операций (см. фигуру 10.3).

Фигура 10.3 – Поток Операций по Умолчанию

Атрибуты потока операций

Таблица 10.2 содержит информацию об атрибутах Потока операций, продолжающих список общих атрибутов Элементов соединения (см. фигуру 10.44):

Таблица 10.2 - Атрибуты Потоков Операций

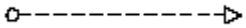
Атрибут	Описание
ConditionType (None | Expression | Default) None : String	Значение атрибута ConditionType Потока операций равно None. Это означает, что во время выполнения действий не будет выполнено вычисление, определяющее, будет ли использован данный Поток операций в Процессе или нет. Если хотя бы однажды Токен готов пересечь Поток операций (т.е. источником Потока операций является действие, которое было завершено), то данный Токен пересечет его. Стандартное, неконтролируемое использование Потока операций в последовательности действий указывает на то, что значение атрибута ConditionType равно None (см. фигуру 10.12). Атрибут ConditionType со значением None НЕ ДОЛЖЕН БЫТЬ использован в случае, если источником Потока операций является Эксклюзивный Шлюз, основанный на Данных или Неэксклюзивный Шлюз. Атрибут ConditionType МОЖЕТ иметь значение Expression в случае, если источником Потока операций является Задача, или Подпроцесс, или Эксклюзивный Шлюз, основанный на данных, или Неэксклюзивный Шлюз. В случае, если Атрибут ConditionType имеет значение Expression, то к линии Потока операций, исходящей от действия, ДОЛЖЕН БЫТЬ добавлен маркер Условного потока операций (см. фигуру 10.2). Однако условный маркер НЕ ДОЛЖЕН БЫТЬ добавлен к линии Потока операций в случае, если данный Поток операций исходит из Шлюза. Выражение ConditionType НЕ ДОЛЖНО БЫТЬ использовано в случае, если источником Потока операций является Эксклюзивный Шлюз, основанный на событиях, или Комплексный Шлюз, или Параллельный Шлюз, или Стартовое событие, или Промежуточное событие. Также выражение ConditionType НЕ ДОЛЖНО БЫТЬ использовано в случае, если Поток операций ассоциирован с Выходом Шлюза, установленным по умолчанию. Атрибут ConditionType МОЖЕТ иметь значение Default лишь в случае, если источником Потока операций является действие или Эксклюзивный Шлюз, основанный на данных. В данном случае ДОЛЖЕН быть изображен маркер Потока операций по умолчанию (фигура 10.3).
[ConditionType is set to Expression only] **ConditionExpression**: Expression	В случае, если значение атрибута ConditionType равно Expression, то атрибут ConditionExpression ДОЛЖЕН являться истинным выражением. Значение данного выражения будет вычислено во время выполнения действий. В случае, если вычисленное значение равно TRUE, то будет создан Токен, который затем пересечет Поток операций в зависимости от ограничений, установленных источником, которым является Шлюз.

Quantity 1 : Integer	Значение атрибута Quantity, установленное по умолчанию, равно 1. Значение НЕ ДОЛЖНО БЫТЬ менее, чем 1. Данный атрибут указывает на то, какое количество Токенов будет создано в ходе Потока операций.

§§ 10.1.3. Поток сообщений

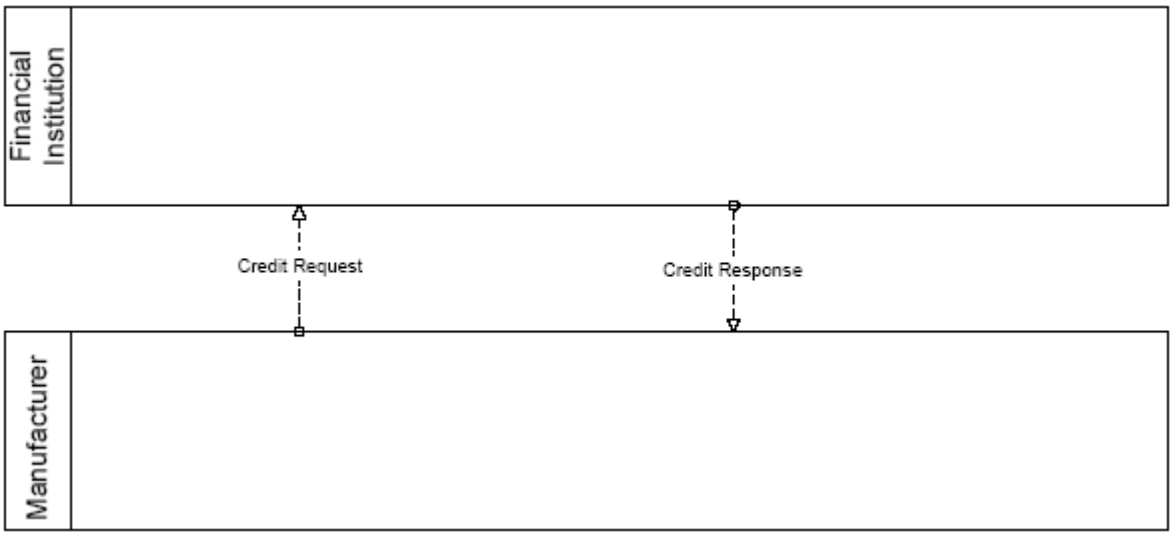
Поток сообщений используется для отображения того порядка, в котором происходит обмен сообщениями между двумя заинтересованными сторонами, готовыми как отсылать, так и получать эти сообщения. В спецификации BPMN заинтересованные стороны представлены Пулами. Таким образом:

- Поток сообщений ДОЛЖЕН соединять два Пула как между собой, так и Элементами потока, расположенными внутри этих Пулов. Однако он не может соединять два элемента, расположенные внутри одного и того же Пула.
- Поток сообщений представляет собой стрелку со свободным концом, линия которой ДОЛЖНА БЫТЬ пунктирной, одиночной и черной (как показано на фигуре 10.4).
    - Текст, цвет, размер, а также линии, используемые для изображения Конечного события, должны соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм» (стр. 26 оригинала спецификации).



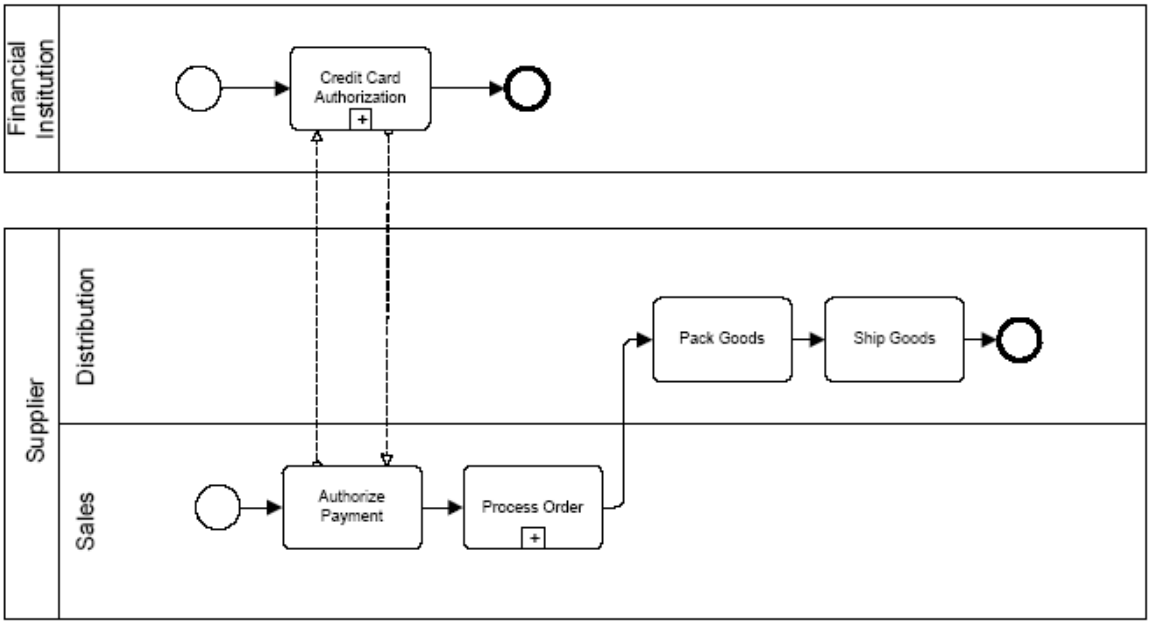
Фигура 10.4 – Поток Сообщений

Поток сообщений может присоединяться непосредственно к границам Пула (см. фигуру 10.5), особенно, если Пул не содержит какой-либо информации о деталях Процесса (например, является «черным ящиком»).



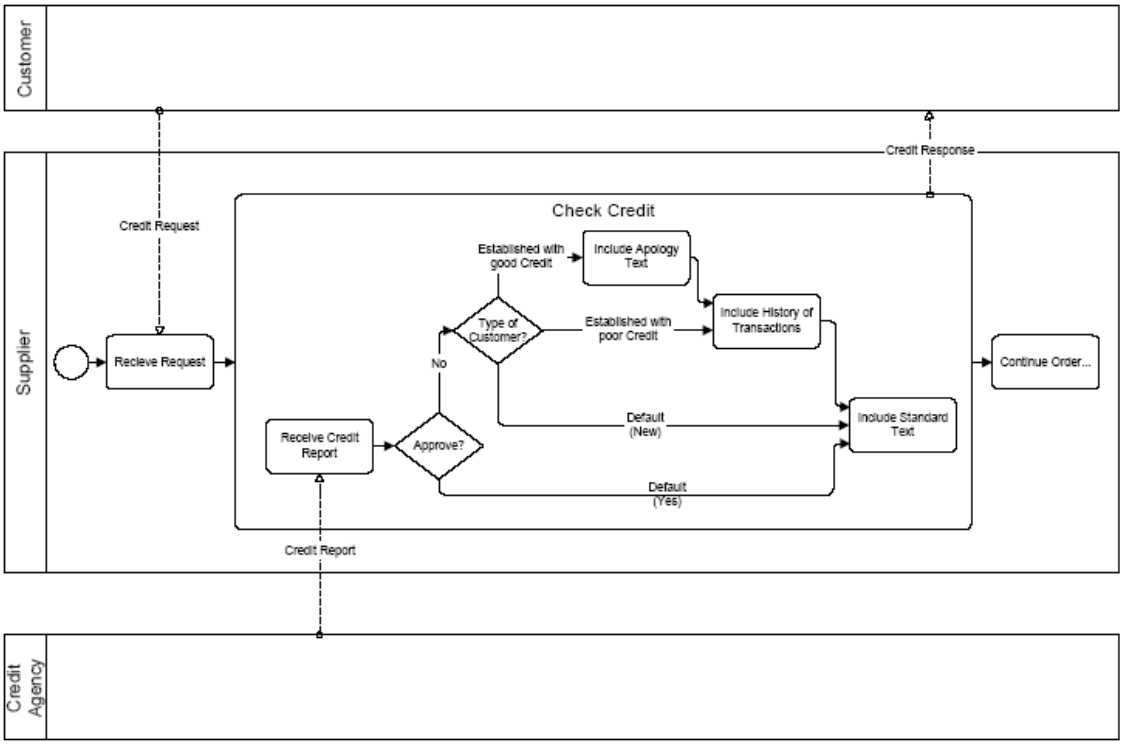
Фигура 10.5 – Поток сообщений, соединенный с границами двух Пулов

Поток сообщений также может пересекать границы Пула и присоединяться к Элементам потока, расположенным в данном Пуле (см. фигуру 10.6).



Фигура 10.6 – Поток сообщений, присоединенный к Элементам потока, расположенным внутри двух Пулов

В случае, если в одном из Пулов содержится Развернутый Подпроцесс, то Поток сообщений может быть присоединен как к границам данного Подпроцесса, так и к каким-либо элементам, находящимся внутри данного Подпроцесса. В случае, если Поток сообщений присоединен к границам Развернутого Подпроцесса, то это является эквивалентом присоединения Входящего потока сообщений к Стартовому событию или Исходящего потока сообщений к Конечному событию.



Фигура 10.7 – Поток сообщений, присоединенный к границам Подпроцесса и внутренним элементам данного Подпроцесса

Атрибуты потока сообщений

Таблица 10.3 содержит информацию об атрибутах Потока сообщений, продолжающих список общих атрибутов Элементов соединения (см. фигуру 10.1):

Таблица 10.3 - Атрибуты Потоков Операций

Атрибут	Описание
Message (0-1) : Message	Атрибут Message является опциональным и указывает на то, какое сообщение отправляется. Дополнительную информацию об атрибутах Сообщения см. в разделе B.11.4, «Message», стр. 269 оригинала спецификации.

§§ 10.1.4. Ассоциация

Ассоциация используется для соединения какой-либо информации или Артефактов с Элементами потока. Текст и графические объекты, не являющиеся Элементами потока, могут быть ассоциированы с Элементами потока или Потоком операций. Ассоциация также применяется для отображения действий, используемых для компенсации другого действия. Дополнительную информацию о компенсации см. в разделе 10.3 «Компенсированная Ассоциация», стр. 133 оригинала спецификации.

- Ассоциация ДОЛЖНА представлять собой пунктирную, одиночную, черную линию (как показано на фигуре 10.8).
    - Текст, цвет, размер, а также линии, используемые для изображения Конечного события, должны соответствовать правилам, указанным в разделе 8.3 «Использование Текста, Цвета и Линий в Моделировании Диаграмм» (стр. 26 оригинала спецификации).

Фигура 10.8 - Ассоциация

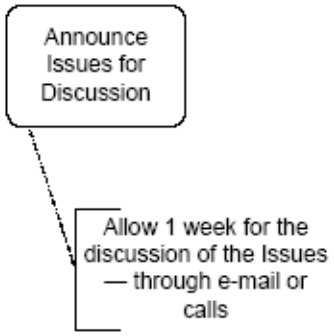
В случае, если необходимо указать направление линии Ассоциации, то:

- К линии Ассоциации МОЖЕТ БЫТЬ добавлена стрелка (см. фигуру 10.9).

Направленная Ассоциация часто используется в сочетании с Объектами данных для того, чтобы показать: по отношению к действию, Объекты данных могут являться как входами, так и выходами.

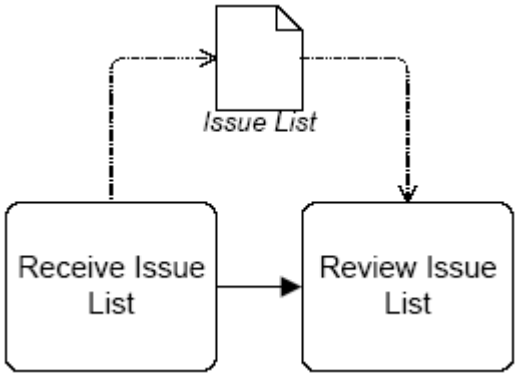
Фигура 10.9 – Направленная Ассоциация

Ассоциация используется для соединения определенного разработчиком текста (Аннотации) с каким-либо Элементом потока (см. фигуру 10.10).



Фигура 10.10 – Ассоциация Текстовой Аннотации

Ассоциация также используется для установления связи между Объектом данных и другими графическими элементами (см. фигуру 10.11). Объект данных используется для отображения того, каким образом в ходе Процесса используются документы. Дополнительную информацию об Объектах данных см. в разделе 9.7.2, «Объект Данных», стр. 93 оригинала спецификации.



Фигура 10.11 – Ассоциация, соединяющая Объект данных с Потоком операций

Атрибуты ассоциации

Таблица 10.4 содержит информацию об атрибутах Ассоциации, продолжающих список общих атрибутов Элементов соединения (см. фигуру 10.1):

Таблица 10.4 - Атрибуты Ассоциации

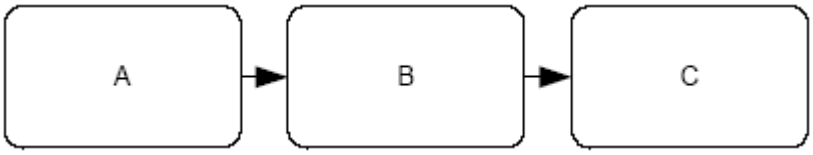
Атрибут	Описание
Direction (None | To | From | Both) None : String	Атрибут Direction определяет, указывает ли Ассоциация на какое-либо направление с помощью стрелки или нет. Значение данного атрибута по умолчанию равно None (стрелка отсутствует). Значение «To» указывает на то, что стрелка ДОЛЖНА быть направлена на исходный объект. Значение «From» указывает на то, что стрелка ДОЛЖНА быть направлена на Артефакт, являющийся целью. Значение «Both» указывает на то, что оба конца линии Ассоциации ДОЛЖНЫ ИМЕТЬ стрелки.

§ 10.2. Типы потоков операций (Sequence Flow Mechanisms)

Существуют следующие типы Потоков операций, описанные ниже: Стандартный поток операций, Поток исключений, События Соединения и Ad Hoc (отсутствие потока). Ввиду разделения Потоков операций на эти четыре типа, спецификацию BPMN можно сравнить с «Моделями последовательности операций» («Workflow Patterns»). Данные модели появились в качестве разработок следующих разработчиков: Wil van der Aalst, Arthur ter Hofstede, Bartek Kiepuszewski и AlistairBarros. В качестве способов поведения документа, которые могут выполняться системой BPM, была определена 21 модель. Поведение этих моделей может быть как очень простым, так и составным (бизнес-поведение). Данные модели могут быть полезными, благодаря тому, что они представляют полный список способов поведения, которые должны учитываться системой BPM. Некоторые модели будут описаны в следующих разделах данной спецификации с целью отображения того, каким образом BPMN может удовлетворять как простым, так и сложным требованиям моделирования бизнес-процессов.

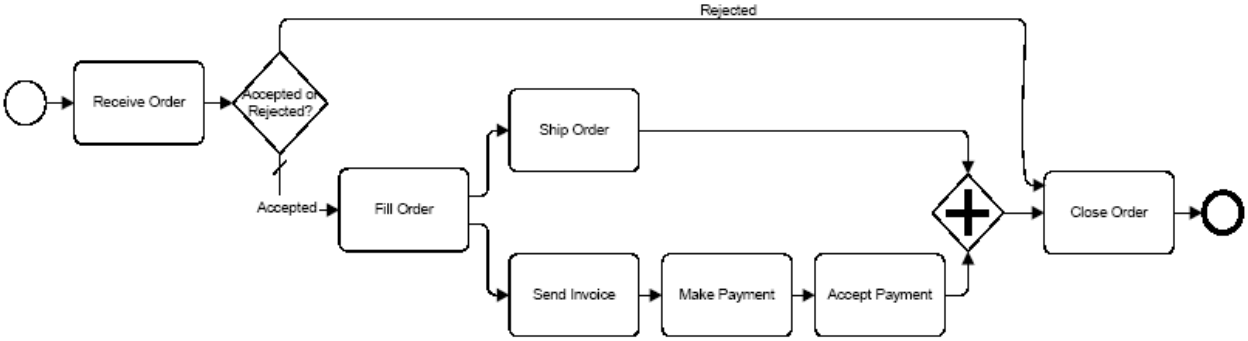
§§ 10.2.1. Стандартный поток операций (Normal Flow)

Стандартный поток операций относится к Потокам операций, берущим начало от Стартового события и проходящим по альтернативным и параллельным маршрутам до своего завершения в Конечном событии. Простейшим примером такого Потока операций в Процесса является последовательность операций, определяющая зависимость заказа для набора действий, которые будут выполнены (последовательно). Последовательность действий также является Моделью последовательности операций №1 – Последовательность (см. фигуру 10.12).



Фигура 10.12 - Модель последовательности операций №1 - Последовательность

Как указано выше, Стандартный поток операций должен быть абсолютно прозрачным; поведение такого Потока операций не должно быть скрытым. Это означает, что на данном уровне Процесса читатель BPMN диаграммы будет способен проследить за ходом Процесса от начала до конца, включая все Элементы потока и Потоки операций, без каких-либо пробелов или скачков (см. фигуру 10.13). Фигура 10.13 отображает присоединение Потока операций ко всем элементам диаграммы: от Стартового до Конечного события. Поведение изображаемого Процесса указывает на соединения Потока операций с Элементами потока, как и показано на рисунке, не пропускает каких-либо действий и не перескакивает к завершению данного Процесса.



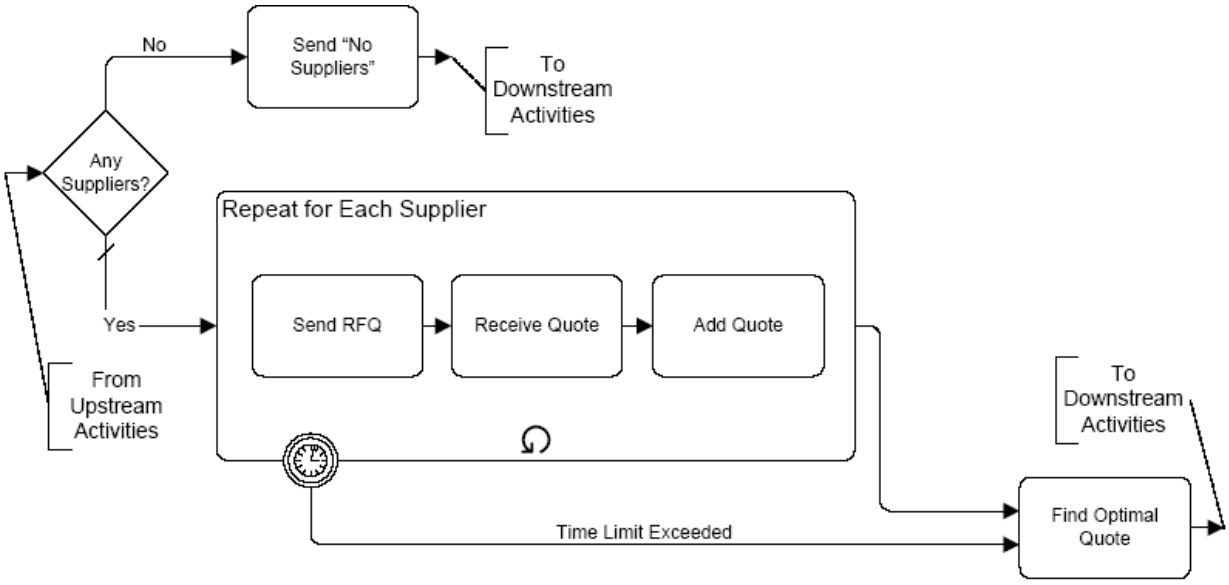
Фигура 10.13 – Процесс, содержащий Стандартный поток операций

Так как Процесс включает последовательность Потоков операций, то механизм управления может разделять или объединять Потоки операций для изображения комплексного поведения Процесса. Существуют механизмы управления для разделения (раздвоение и разделение), а также для объединения (объединение и слияние) Потоков операций. Разделение и объединение Потоков операций осуществляются благодаря использованию Шлюзов и Условных потоков операций. В случае, если Шлюзы и/или Условные потоки операций не сконфигурированы таким образом, чтобы обеспечивать все функциональные возможности, то в Потоке операций могут возникать пробелы. В данном случае модель, нарушающая требование прослеживаемости Потока операций, будет считаться неверной. Возможно, программное обеспечение, направленное на разработку моделей процессов, или средства тестирования моделей бизнес-диаграмм будут в состоянии проверять модель процесса на её адекватность.

Случайная просмотр значений английских терминов, соответствующих данным механизмам (например, forking – раздвоение и splitting - разделение), указывает на то, что оба термина, включенные в пару, по существу означают одно и то же. Однако их

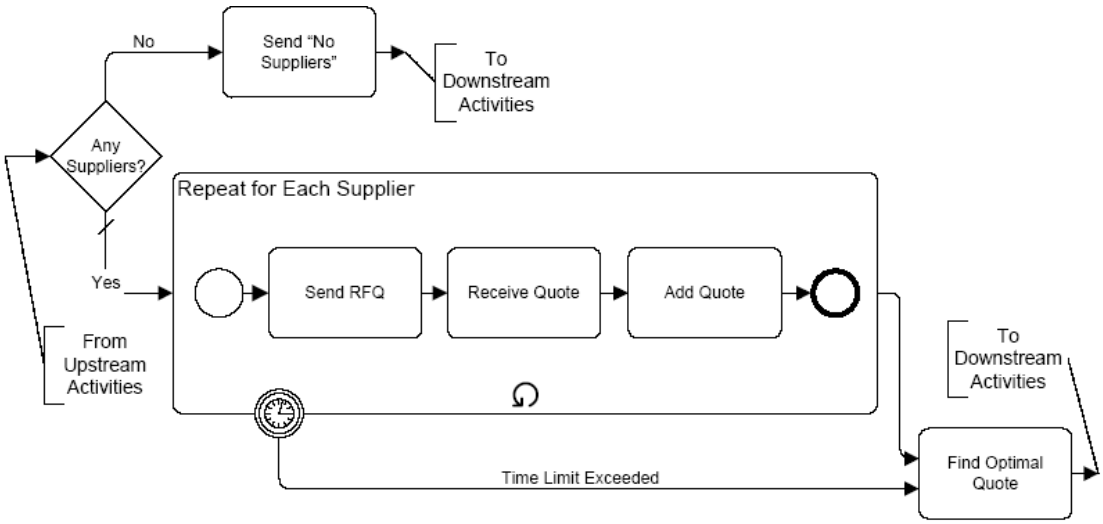
воздействие на поведение Процесса ощутимо различается. Данные английский термины будут использоваться в спецификации и далее, но в следующих нескольких её разделах будут определены точные значения используемых терминов, характеризующие их влияние на ход процесса. Также эти BPMN термины будут сравниваться с таким английскими терминами, как OR-Split (ИЛИ – Разделение; используется для разделения), Or-Join (ИЛИ – Соединение; используется для объединения), AND-Split (И – Разделение; используется для раздвоения), а также AND-Join (И – Соединение; для объединения), утвержденными Коалицией по Управлению Технологическим Потоком (Workflow Management Coalition).

Использование в Процессе Развернутого Подпроцесса (см. фигуру 10.14), являющегося включением одного уровня Процесса в другой его уровень, иногда может нарушать прослеживаемость Потока операций по линиям диаграммы. Подпроцесс не обязательно должен содержать в своем составе Стартовое и Конечное события. Это означает, что данная последовательность Потоков операций будет разорвана на отрезке, расположенном между границей Развернутого Подпроцесса и первым элементом данного Подпроцесса. Ход Потока операций перепрыгнет к первому элементу Развернутого Подпроцесса. Как показано на рисунке, такой Развернутый Подпроцесс часто используется для изображения наличия выполняемого исключения. Требование об обязательном включении в состав Развернутого Подпроцесса Стартового и Конечного событий, предъявляемое к разработчикам моделей, в основном только перегружает диаграмму бизнес-процесса и не содействует созданию прозрачной диаграммы. Таким образом, спецификация BPMN не требует использования Стартового и Конечного событий для удовлетворения требования прослеживаемости многоуровневой диаграммы.



Фигура 10.14 – Развернутый Подпроцесс, не содержащий Стартового и Конечного событий

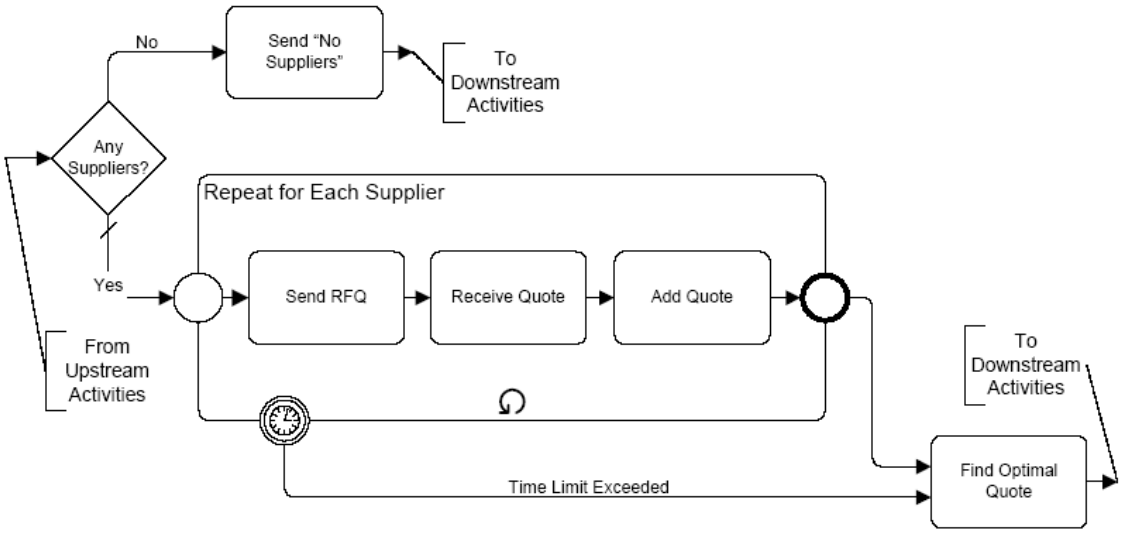
Развернутый Подпроцесс также может содержать в своем составе Стартовое и Конечное события, которые помещаются в пределах данного Подпроцесса (см фигуру 10.15). Такая модель также далека от полной прослеживаемости Потока операций по ходу перехода Процесса с одного уровня на другой.



Фигура 10.15 – Развернутый Подпроцесс со Стартовым и Конечным событиями в составе

Однако разработчик модели может пожелать удостовериться в том, что диаграмма является прослеживаемой, и включить Стартовое и Конечное события в состав Развернутого Подпроцесса. Одним из способов сделать это является присоединение такого типа событий к границам Развернутого Подпроцесса (см. фигуру 10.16). Поток операций, являющийся входящим для данного Подпроцесса, может быть присоединен непосредственно к Стартовому событию, а не к границам Подпроцесса. Подобно этому, Поток операций, исходящий от данного Подпроцесса, может быть присоединен непосредственно к Конечному событию, а не к границам Подпроцесса. Благодаря такому присоединению, ход Стандартного потока операций может прослеживаться даже в многоуровневом Процессе.

С технической точки зрения, Стартовое и Конечное события находятся в Подпроцессе. Использование такой техники моделирования заключается в графическом уменьшении изображения Процесса в более точное (как показано на фигуре 10.15). Следовательно, Поток операций, присоединенный к Стартовому события и отходящий от Конечного события, не нарушает правил соединения Потока операций (см. раздел «Правила соединения Потока операций», стр. 38 оригинала спецификации, а также раздел «Правила соединения Потока операций», стр. 42 оригинала спецификации).



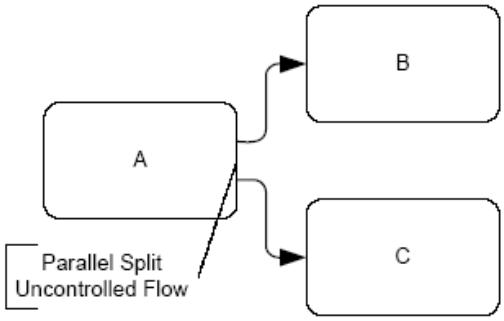
Фигура 10.16 – Развернутый Подпроцесс со Стартовым и Конечным событиями, прикрепленным к границам Подпроцесса

При использовании Исключений и Компенсации требование прослеживаемости также смягчается (см. раздел 10.2.2 «Поток исключений», стр. 130 оригинала спецификации, а также раздел 10.3 «Компенсированная Ассоциация», стр. 133 оригинала спецификации).

Разделение потоков операций (Forking Flow)

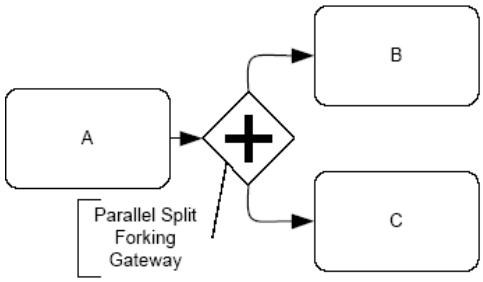
В спецификации BPMN под термином разделение (forking) подразумевается разделение потока на два или более параллельных маршрутов (другое название – И-Разделение). Разделение представляет собой механизм, позволяющий выполнять несколько действий одновременно, а не последовательно, и является Моделью технологического процесса №2 – Параллельное Разделение (Parallel Forking). Спецификация выделяет три механизма создания разделения.

Первый механизм создания разделения маршрутов является простым: Элемент потока соединен с двумя или более Исходящими потоками операций (см. фигуру 10.17). В данном случае для разделения маршрута не используется специальный контролирующий Элемент потока, т.к. такой Поток операций считается неконтролируемым. Таким образом, Поток операций будет идти по каждому из маршрутов вне зависимости от условий, т.к. отсутствует Шлюз, контролирующий данный Поток операций. Разделяющийся Поток операций может исходить от Задачи, Подпроцесса или Стартового события.



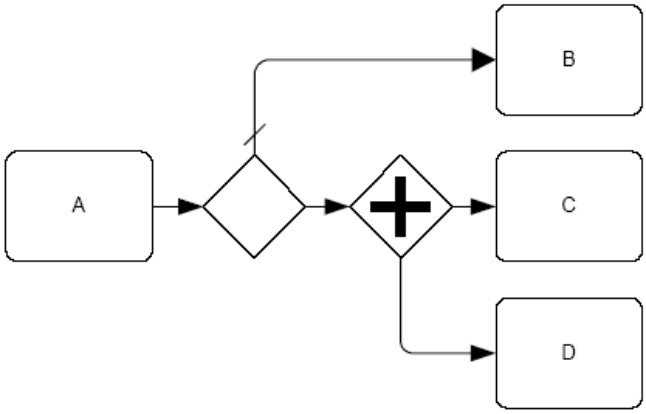
Фигура 10.17 – Модель технологического процесса №2: Параллельное Разделение – Версия 1

Второй механизм создания разделения маршрутов использует Параллельный Шлюз (см. фигуру 10.21). Ситуации, подобные изображенной на рис. 10.18, не требуют использования Шлюзов, т.к. такое же поведение может быть создано при помощи нескольких Исходящих потоков операций (см. фигуру 10.17). Однако, согласно мировому передовому опыту, для разделения маршрутов разработчики моделей и инструменты моделирования могут использовать Шлюз. Дополнительную информацию о Параллельных Шлюзах см. в разделе 9.5.5 «Параллельный Шлюз (И)», стр. 85 оригинала спецификации.



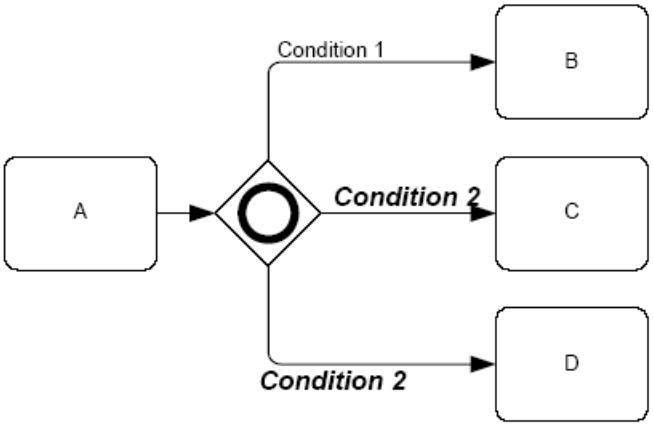
Фигура 10.18 – Модель технологического процесса №2: Параллельное Разделение – Версия 2

Однако, несмотря на мировой передовой опыт, существуют ситуации, в которых Параллельный Шлюз выступает в качестве полезного индикатора поведения Процесса. Фигура 10.19 показывает, каким образом Шлюз, предназначенный для разделения, используется в случае, когда выход Эксклюзивного Условия требует выполнения нескольких действий в соответствии с одним условием (Выходом).



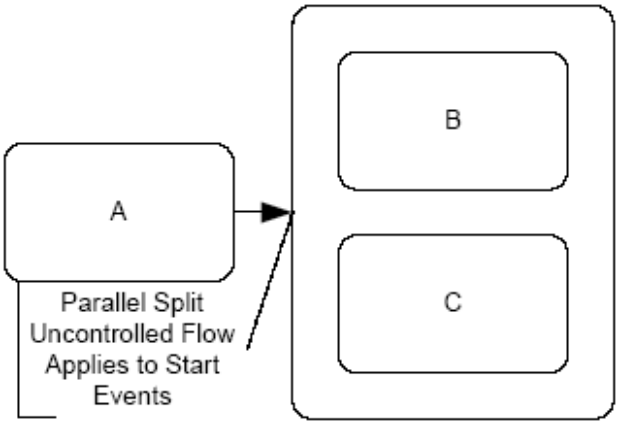
Фигура 10.19 – Создание параллельных маршрутов при помощи Шлюза

До тех пор, пока в Процессе присутствует несколько условных Потоков операций, каждый из них, имеющий то же самое условное выражение (фигура 10.20), может быть использован с Неэксклюзивным Шлюзом для создания необходимого поведения. Использование разделяющего Шлюза делает такого рода поведение гораздо более очевидным.



Фигура 10.20 – Создание параллельных маршрутов с равными условиями

Третий механизм создания разделения маршрутов для группировки действий с последующим их параллельным выполнением использует Развернутый Подпроцесс (си. фигуру 10.21). Такой Подпроцесс не содержит Стартового и Конечного событий и отображает действия, протекающие внутри данного Подпроцесса. Подобного рода конфигурацию, позволяющую изобразить параллелизм, присутствующий в Процессе, более компактно и с меньшим количеством лишних элементов, можно определить как «параллельный ящик». Существование такой конфигурации обусловлено тем, что Стартовое и Конечное события являются в спецификации BPMN опциональными.

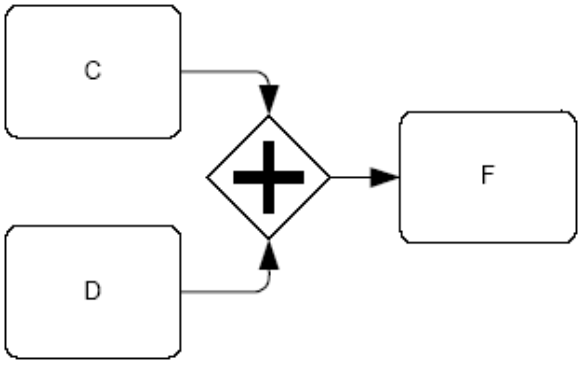


Фигура 10.21 - Модель технологического процесса №2: Параллельное Разделение – Версия 3

Практически всегда ранее разделенные маршруты объединяются снова посредством объединителя (см. следующий раздел) и синхронизируются до того, как продолжится Поток операций. Однако при комплексных ситуациях Процесса данная спецификация предлагает использование более продвинутых методов. Таким образом, точное поведение Процесса определяется конфигурацией используемых Потоков операций и Шлюзов.

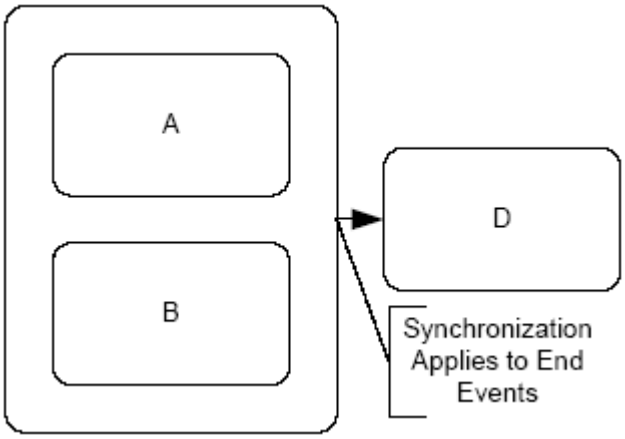
Объединяющий поток операций (Joining Flow)

Термин объединение (joining) используется в спецификации BPMN для объединения двух или более параллельных маршрутов в один (также используется термин И-Соединение). Для синхронизации двух или более Входящих потоков операций используется Параллельный Шлюз (см. фигуру 10.22). В целом, это означает, что Токены, созданные при разделении маршрутов, будут идти по параллельным маршрутам, а затем встретятся у данного Параллельного Шлюза, откуда путь продолжит лишь один из Токенов. Такого рода ситуация представляет собой Модель технологического процесса №3 – Синхронизация. Дополнительную информацию о Параллельных Шлюзах см. в разделе 9.5.5 «Параллельный Шлюз (И)», стр. 85 оригинала спецификации.



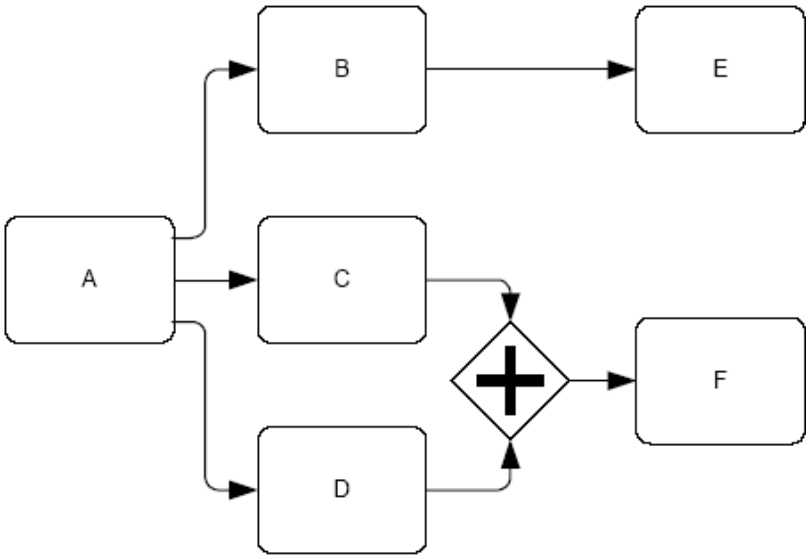
Фигура 10.22 - Модель технологического процесса №3: Синхронизация – Версия 1

Другой механизм синхронизации представляет собой завершение Подпроцесса (см. фигуру 10.23). В случае, если в Подпроцессе содержится несколько параллельных маршрутов, не синхронизированных посредством Параллельного Шлюза, то все они в итоге достигнут Конечного события (даже, если подразумевается Конечное событие). Поведение Подпроцесса, установленное по умолчанию, заключается в ожидании момента, когда все действия, содержащиеся в данном Подпроцессе, завершатся до того, как Поток операций переместится обратно на более высокий уровень Процесса. Таким образом, завершение Подпроцесса является точкой синхронизации.



Фигура 10.23 - Модель технологического процесса №3: Синхронизация – Версия 2

Между объединением нескольких параллельных маршрутов и разделением, создающим эти параллельные маршруты, конкретной взаимосвязи нет. Например, действие может иметь три Исходящих потока операций, создающих разделение на три параллельных маршрута, однако, нет необходимости в объединении трех данных маршрутов посредством одного и того же элемента диаграммы. Фигура 10.24 показывает то, каким образом два из трех существующих параллельных маршрутов объединяются в Задаче «F». Все маршруты будут в итоге объединены, но их объединение может произойти посредством любой комбинации элементов диаграммы, включая отдельные Конечные события. В действительности, каждый маршрут может закончиться отдельно взятым Конечным событием, а затем, как было сказано выше, будет синхронизированы.



Фигура 10.24 – Отношения Разделение-Объединение не зафиксированы

Таким образом, что касается параллельных Потоков операций, то здесь BPMN контрастирует с BPEL4WS, который в целом является блочным. В BPEL4WS поток (flow), соответствующий совокупности параллельных действий в спецификации BPMN, представляет собой специфическую блочную структуру со строго определенными границами. Т.к. параллельные маршруты, созданные при помощи разделителя, не имеют четких границ, то

соответствующие границы могут быть созданы за счет определения конфигурации Потока операций, следующего за разделением. Отрезки Процесса, на которых Токены с одним и тем же идентификатором, а также со всеми соответствующими им идентификаторами Подтокенов объединяются со несколькими сквозными Входящими потоками операций,

определяют границы конкретного блока параллельных действий. В действительности, границы могут представлять собой завершение Процесса. Дополнительную информацию об определении границ элемента BPEL4WS см. в главе 11 «Соответствие с BPEL4WS».

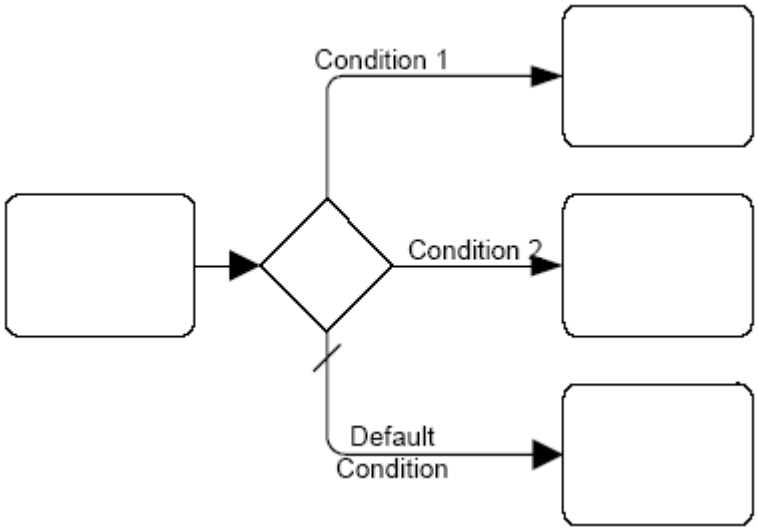
Распределяющий поток операций (Splitting Flow)

Термин распределение (splitting) используется в спецификации BPMN для разбиения маршрута на два или более альтернативных (также используется термин ИЛИ-Разделение). Такая ситуация возникает на отрезке Процесса, где задается вопрос, а ответ определяет, какие из маршрутов будут использованы. Данная ситуация является разделением, где движущийся элемент – в данном случае Токен – может использовать лишь один из разделителей (не путать с разделение (forking), см. ниже).

Общим понятием, на которое ссылается распределение Потока операций, обычно является Условие. В традиционных методологиях изображения схемы Потока операций Условия изображаются в виде ромбов и являются эксклюзивными. Ромб используется в спецификации BPMN, т.к. данный графический элемент является известным, однако, в данной спецификации ромб используется ещё и для управления комплексными ситуациями, возникающими в бизнес-процессах (такими, управление которыми не может быть выполнено посредством традиционных схем изображения процессов). Графический элемент ромба используется как для изображения Шлюзов, так и для изображения начала Условных потоков операций (отходящих от действия). Таким образом, при просмотре диаграммы бизнес-процесса пользователь видит ромб и знает, что Поток операций каким-то образом контролируется, а не просто переходит от одного действия к другому. Расположение мини-ромба и внутренних индикаторов Шлюзов указывает на то, каким образом будет осуществляться контроль над Потоком операций.

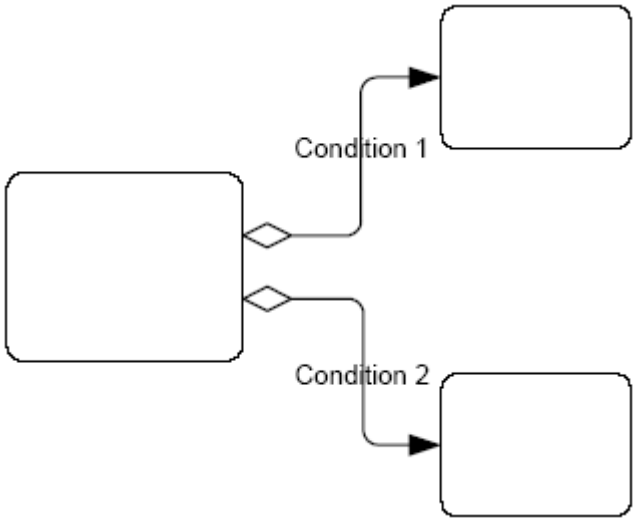
В спецификации BPMN для разбиения Потока операций существует несколько конфигураций, что позволяет создавать разнообразные модели поведения Процесса. Для разделения Потока операций используется Условный поток операций, а также три вида Шлюзов (Эксклюзивный, Неэксклюзивный и Комплексный). Дополнительную информацию об Условном потоке операций см. в разделе 10.1.1 «Поток операций», стр. 100 оригинала спецификации. Дополнительную информацию о Шлюзах см. в разделе 9.5 «Шлюз», стр. 68 оригинала спецификации.

Существуют два основных механизма создания Условия в ходе выполнения Процесса. Первый механизм представляет собой вычисление значения условного выражения. Данный механизм бывает трех разновидностей: Эксклюзивный, Неэксклюзивный и Комплексный. Первая разновидность данного механизма – Эксклюзивное Условие – представляет собой Модель технологического процесса №4 – Эксклюзивный Выбор (см. фигуру 10.25). Дополнительную информацию о Эксклюзивных Шлюзах, основанных на Данных, см. в разделе «Эксклюзивный Шлюз, Основанный на Данных», стр. 71 оригинала спецификации.



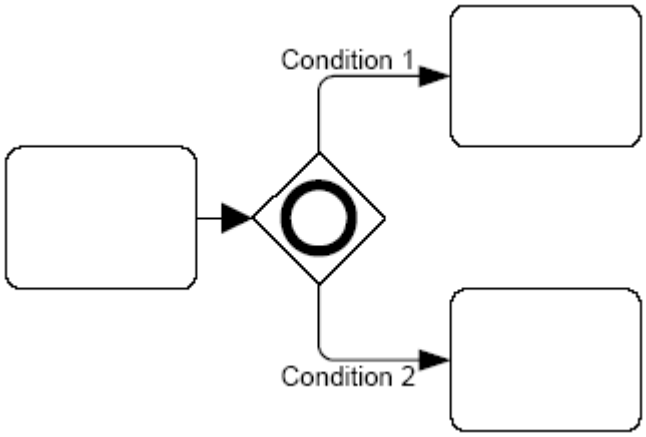
Фигура 10.25 – Пример Условия, основанного на Данных, представляющего собой Модель технологического процесса №4 – Эксклюзивный Выбор

Вторая разновидность данного механизма представляет собой Неэксклюзивное Условие, представляющее собой Модель технологического процесса №6 – Множественный Выбор. Существуют два типа Неэксклюзивных Условий. Первый тип Неэксклюзивного Условия использует Условный поток операций, отходящий от Действия (см. фигуру 10.26).



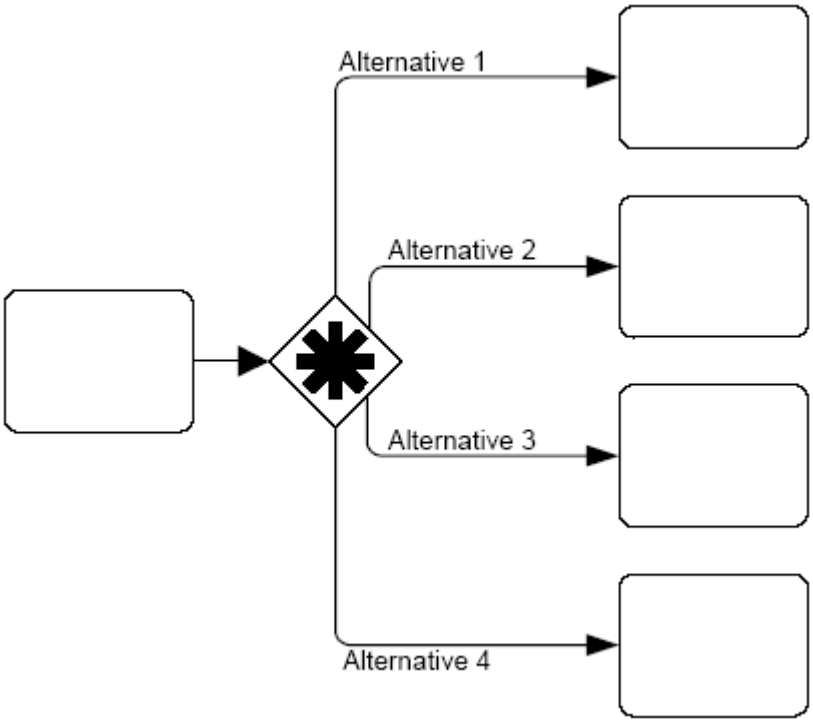
Фигура 10.26 – Модель технологического процесса №6 – Множественный Выбор - Версия 1

Второй тип Неэксклюзивного Условия для осуществления контроля над Потоком операций использует Неэксклюзивный Шлюз (см. фигуру 10.27). Дополнительную информацию об Неэксклюзивных Шлюзах см. в разделе 9.5.3 «Неэксклюзивный Шлюз (ИЛИ)», стр. 78 оригинала спецификации.



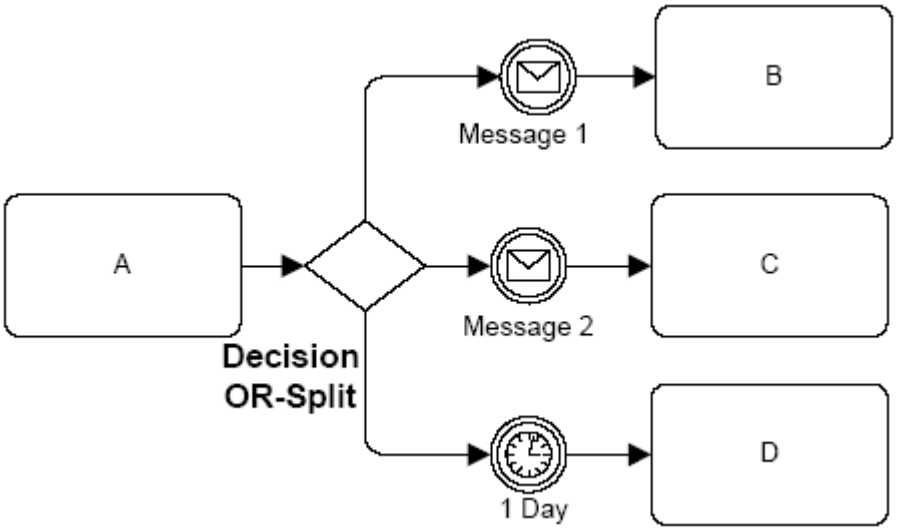
Фигура 10.27 – Модель технологического процесса №6 – Множественный Выбор - Версия 2

Третья разновидность данного механизма представляет собой Комплексное Условие (см. фигуру 10.28). Дополнительную информацию о Комплексных Шлюзах см. в разделе 9.5.4 «Комплексный Шлюз», стр. 82 оригинала спецификации.



Фигура 10.28 – Комплексное Условие (Шлюз)

Второй механизм создания Условия заключается в возникновении определенного события, например, получения сообщения (см. фигуру 10.29). Дополнительную информацию об Эксклюзивных Шлюзах, основанных на Событиях, см. в разделе «Эксклюзивный Шлюз, Основанный на Событиях», стр. 75 оригинала спецификации.



Фигура 10.29 – Пример Условия, основанного на Событиях

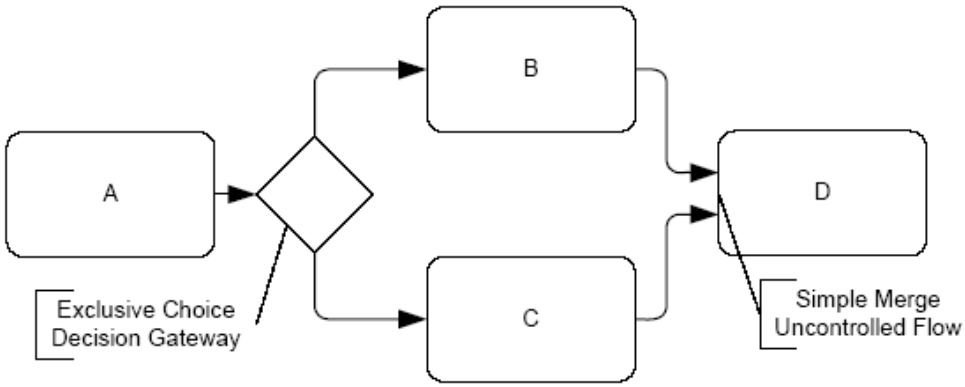
Соединяющий поток операций (Merging Flow)

Термин соединение (merging) используется в спецификации BPMN для объединения двух или более альтернативных маршрутов в один (также используется термин ИЛИ-Соединение). Такая ситуация возникает на отрезке Процесса, где два или более альтернативных маршрутов начинают пересекать действия, общие для всех этих маршрутов. Теоретически, любой альтернативный маршрут может быть смоделирован индивидуально к моменту завершения Процесса (к Конечному событию). Однако соединение позволяет совмещать маршруты и избегает дублирования действий, общих для отдельных маршрутов. В данном примере Процесса Токен лишь наблюдает за последовательностью действий, принадлежащей одному маршруту, как если бы этот маршрут был смоделирован отдельно к моменту завершения Процесса.

Существует несколько способов как разделения и распределения Потоков операций, так и их соединения. Выделяют пять различных Моделей технологического процесса, использующих соединение.

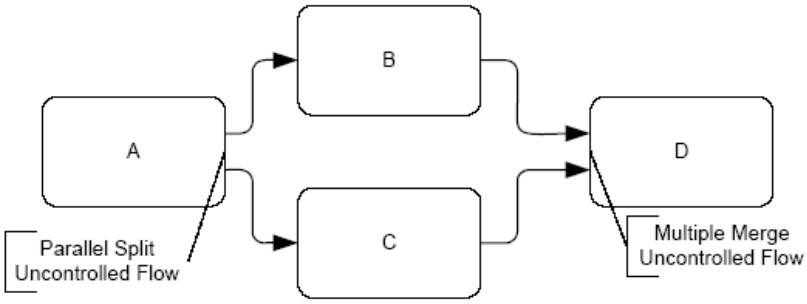
Первая Модель технологического процесса, использующая соединение, представляет собой Простое Соединение (Simple Merge). Графический механизм соединения альтернативных маршрутов является простым: Элемент потока соединяется с двумя или более Входящими потоками операций (см. фигуру 10.30). В целом, это означает, что Токен будет проходить по одному из альтернативных маршрутов (в данном примере Процесса) и продолжать свой ход из этой точки. К примеру, Токены никогда не будут следовать за альтернативными маршрутами. Спецификация BPMN предоставляет две версии Простого Соединения.

Первая версия Простого Соединения представлена на рис.10.30. Два Входящих потока операций для действия «D» являются неконтролируемыми. Т.к. два данных Потока операций находятся на концах двух альтернативных маршрутов, созданных с помощью предшествующего Эксклюзивного Шлюза, то лишь один Токен достигнет действия «D» данного Процесса.



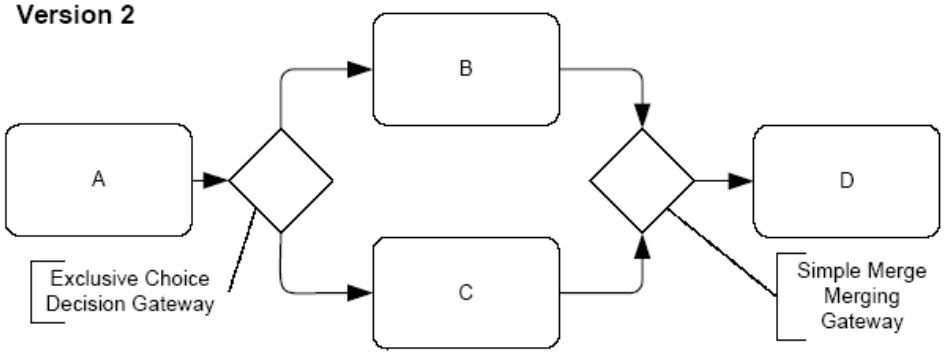
Фигура 10.30 – Модель технологического процесса №5: Простое Соединение – Версия 1

В случае, если несколько Входящих потоков операций в действительности являются параллельными, а не альтернативными, то конечный результат будет иным, даже несмотря на то, что конфигурация соединения осталась той же самой, что и на фигуре 10.30. Фигура 10.31 содержит изображение предшествующего параллельного поведения. Таким образом, в данном случае Процесс будет содержать два Токена, подходящих (в разное время) к действию «D». Т.к. Входящий поток операций, присоединенный к действию «D», является неконтролируемым, то каждый Токен, подходящий к действию «D», будет способствовать появлению нового экземпляра данного действия. Эта концепция очень важна для понимания разработчиков, использующих спецификацию BPMN. Такой вид соединения также является Моделью технологического процесса – Множественное Соединение (Multiple Merge).



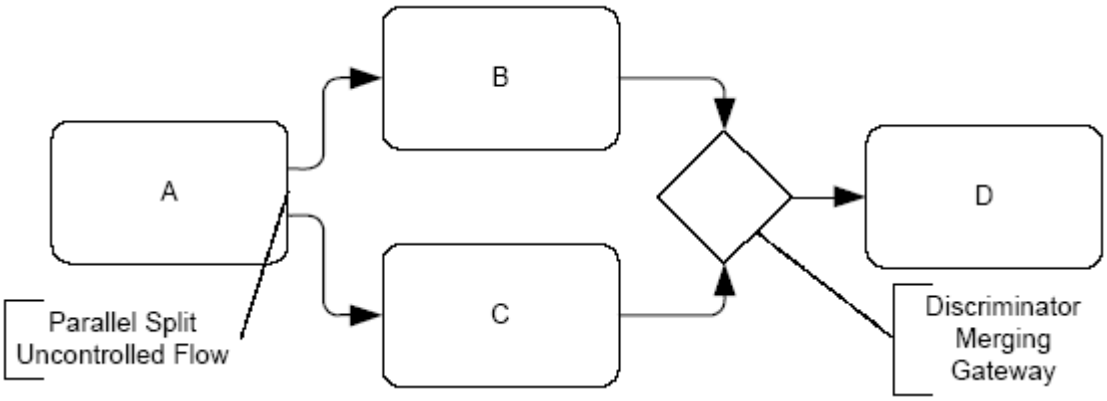
Фигура 10.31 - Модель технологического процесса №7 – Множественное Соединение

Вторая версия Простого Соединения представлена на рис. 10.32. Два Входящих потока операций для действия «D» контролируются Эксклюзивным Шлюзом. Т.к. два данных Входящих потока операций находятся на концах двух альтернативных маршрутов, созданных с помощью предшествующего Эксклюзивного Шлюза, то лишь один Токен достигнет этого Шлюза в любом экземпляре Процесса. Затем Токен немедленно продолжит свое движение по направлению к действию «D».



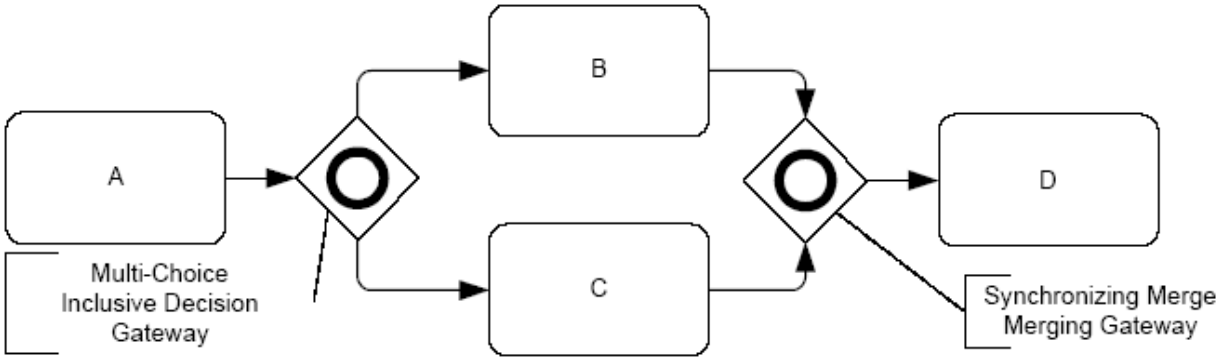
Фигура 10.32 - Модель технологического процесса №5: Простое Соединение – Версия 2

И опять-таки, в случае, если несколько Входящих потоков операций в действительности являются параллельными, а не альтернативными, то конечный результат будет иным, даже несмотря на то, что конфигурация соединения осталась той же самой, что и на фигуре 10.32. Фигура 10.33 содержит изображение модели Процесса, которая включает два Токена, подходящих (в разное время) к Эксклюзивному Шлюзу, предшествующему действию «D». В данном случае Шлюз примет первый Токен и передаст его действию, пропустив через себя. По прибытии, второй Токен будет исключен из окна напоминания Потока операций. Это означает, что второй Токен не пройдет через Шлюз к действию, а будет уничтожен. Такой тип соединения представляет собой Модель технологического процесса - Дискриминатор (Discriminatior).



Фигура 10.33 - Модель технологического процесса №8 - Дискриминатор

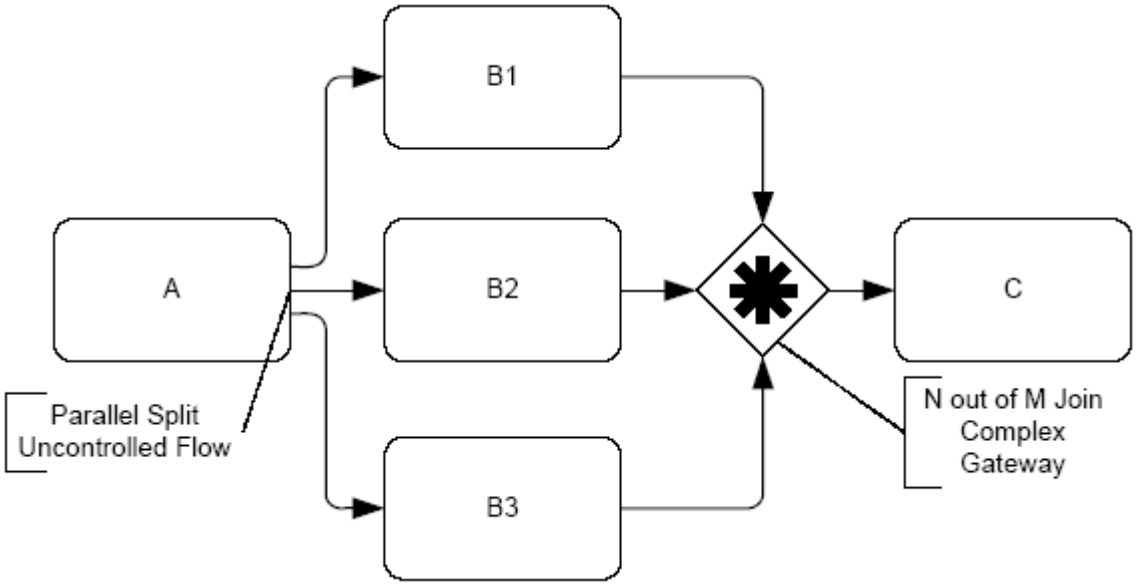
Третий тип Модели технологического процесса, использующей соединение, представляет собой Синхронизирующее Объдинение (Synchronizing Join). Такая ситуация возникает тогда, когда до срока завершения Процесса в месте слияния неизвестно, сколько Токенов прибудут к Шлюзу. Некоторые экземпляры Процесса могут содержать лишь один Токен. Другие его экземпляры могут содержать более одного прибывающего Токена. Такого рода ситуация возникает в случае, когда Неэксклюзивное Условие является частью потока (см. фигуру 10.34). Для управления этой ситуацией и объединения соответствующего количества Токенов каждого экземпляра Процесса может быть использован Неэксклюзивный Шлюз. Шлюз, следующий за Синхронизирующим Соединением, будет ожидать прибытия всех необходимых Токенов до того, как Процесс операций перейдет к следующему действию. Дополнительную информацию о Неэксклюзивных Шлюзах см. в разделе 9.5.3 «Неэксклюзивный Шлюз (ИЛИ)», стр. 78 оригинала спецификации.



Фигура 10.34 - Модель технологического процесса №9 – Синхронизирующее Объединение

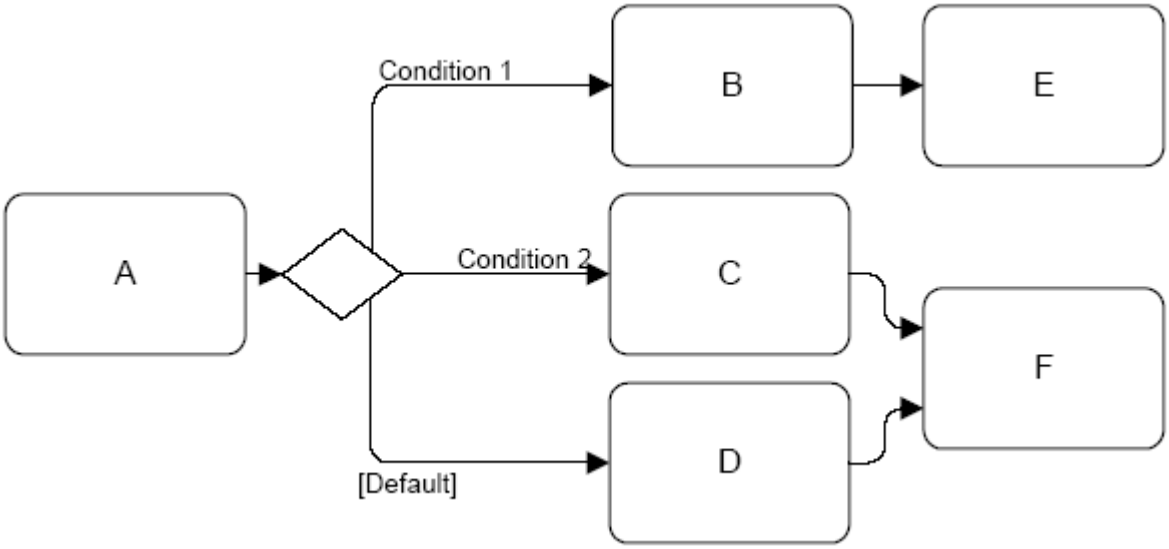
Четвертый тип Модели технологического процесса, использующей соединение, называется Объединение N из M (N out of M Join). Такого рода ситуация считается более сложной и может управляться с помощью Комплексного Шлюза (см. фигуру 10.35).

Комплексный Шлюз получает Токены из Входящего потока операций и вычисляет значение выражения для того, чтобы определить, должен ли данный Поток операций продолжать движение. В случае, если условие было удовлетворено хотя бы один раз, то дополнительные Токены, если они прибыли, будут исключены (данная модель схожа с Моделью «Дискриминатор», изображенной на фигуре 10.33). Дополнительную информацию о Комплексных Шлюзах см. в разделе 9.5.4 «Комплексный Шлюз», стр. 82 оригинала спецификации.



Фигура 10.35 - Модель технологического процесса №8 – Объединение N из M

Между соединением и разделением нескольких маршрутов, осуществляемыми с помощью графического элемента Шлюза, особой взаимосвязи нет. Например, Условие может разделять один маршрут на три отдельных, однако, данные три маршрута не нуждаются в объединении в одном и том же элементе диаграммы. Рис. 10.36 содержит изображение того, как два или три альтернативных маршрута объединяются в Задаче «F». Все маршруты в итоге будут объединены, однако, их объединение может выполнено посредством любой комбинации элементов диаграммы, включая отдельно взятые Конечные события. В действительности, каждый из маршрутов может завершиться в любом Конечном событии.



Фигура 10.36 - Отношения Разделение-Объединение не зафиксированы

Таким образом, что касается альтернативных Потоков операций, то здесь BPMN контрастирует с BPEL4WS, который в целом является блочным. В BPEL4WS switch и pick, соответствующие Эксклюзивному Шлюзу в спецификации BPMN, представляет собой специфическую блочную структуру со строго определенными границами. Т.к. в BPMN альтернативные маршруты, созданные при помощи условия, не имеют четких границ, то соответствующие границы могут быть созданы за счет определения конфигурации Потока операций, следующего за данным условием. Отрезки Процесса, на которых Токены с одним и тем же идентификатором объединяются посредством нескольких Входящих потоков операций, определяют границы конкретного условия. В действительности, границы могут представлять собой завершение Процесса. Дополнительную информацию об определении границ элемента BPEL4WS см. в главе 11 «Соответствие с BPEL4WS».

Цикличность (Looping)

Спецификация BPMN выделяет два типа механизма создания цикличности внутри Процесса. Для определения цикла первый механизм задействует атрибуты действий, а второй – соединение Потока операций с предшествующими элементами диаграммы.

Цикличность действия (Activity Looping)

Атрибуты Задач и Подпроцессов определяют, будет ли данная Задача или Подпроцесс повторяться как цикл. Можно выделить два вида циклов: Стандартный цикл и Многоэкземплярный цикл.

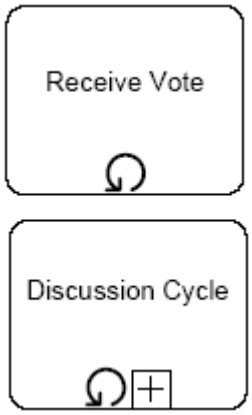
В случае, если цикл является Стандартным, то:

- Если значение условия цикла вычисляется до начала действия, то такой цикл относится к циклам «пока» (while). Это означает, что действия будут повторяться на протяжении всего периода, пока условие имеет значение «true». Действия могут быть либо вообще не выполнены (в случае, если сначала условие имело значение «false»), либо выполнены несколько раз.
- Если значение условия цикла вычисляется после выполнения действия, то такой цикл относится к циклам «до» (until). Это означает, что действия будут повторяться до тех пор, пока значение условия не измениться на «true». Действия будут выполнены по меньшей мере один раз, однако, могут выполняться несколько раз.

В случае, если цикл является Многоэкземплярным, то:

- Если атрибут MI_Ordering является последовательным, то такой цикл скорее относится к циклам «пока» с установленным количеством повторений, включенных в данный цикл. Такие циклы используются в Процессах, где особый тип элементов имеет установленное количество подэлементов или элементов линии. Многоэкземплярные циклы используются для обработки каждого элемента линии.
- Если атрибут MI_Ordering является параллельным, то такой цикл относится к множественным экземплярам действий. Такой цикл может быть использован в Процессе написания книги, где для написания глав используется Подпроцесс. В данном случае Процесс будет содержать количество копий или экземпляров Подпроцесса, равное количеству глав книги. Все экземпляры Подпроцесса могут начинаться в одно и то же время.

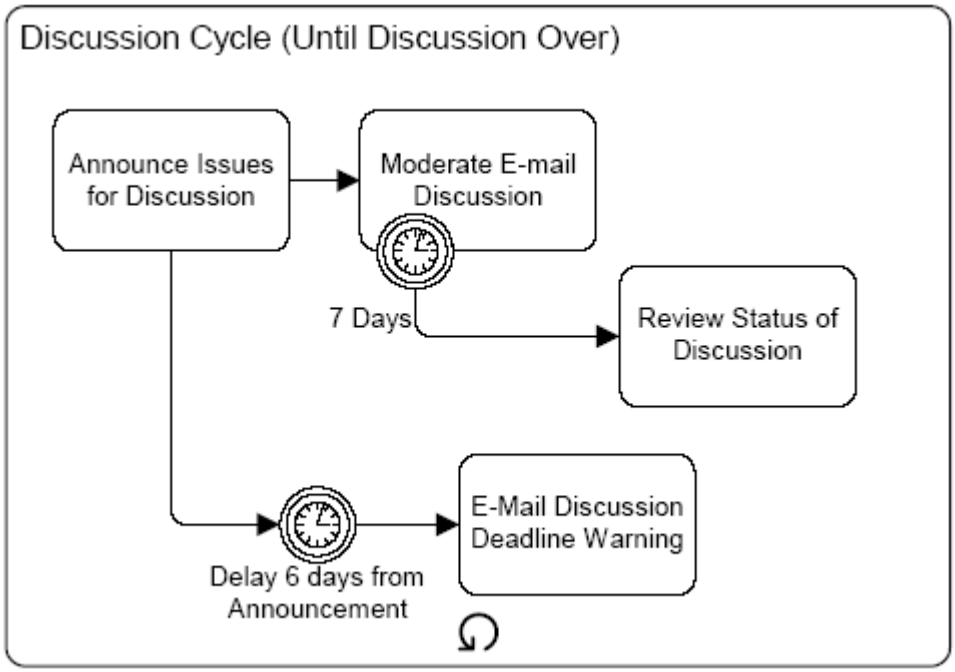
Повторяющиеся (циклами) действия имеют маркер цикла, расположенный в центре нижней части графического элемента Действия (см. фигуру 10.37). Параллельные Многоэкземплярные действия имеют параллельный маркер, расположенный в центре нижней части графического элемента Действия (см. фигуру 10.38).



Фигура 10.37 – Задача и Свернутый Подпроцесс с маркерами цикла

Фигура 10.38 – Задача с параллельным маркером

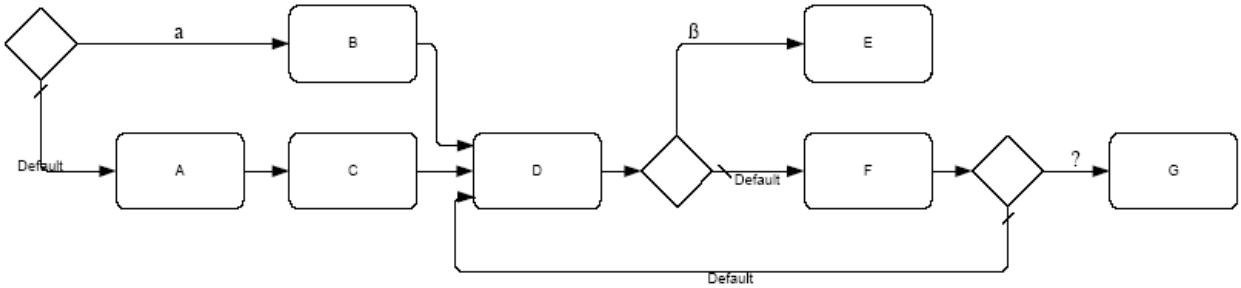
Развернутый Подпроцесс также может содержать маркер цикла, расположенный в центре нижней части графического элемента Подпроцесса (см. фигуру 10.39). Элементы, включенные в такой Подпроцесс, будут повторяться так часто, как это указано в значениях атрибутов.



Фигура 10.39 – Развернутый Подпроцесс с параллельным маркером

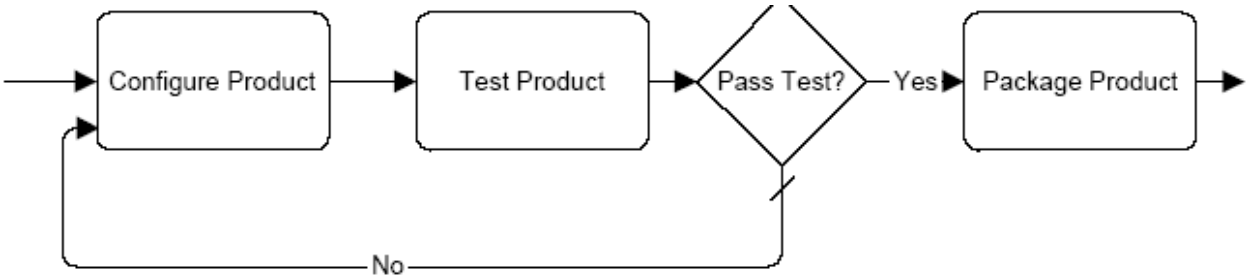
Цикличность потока операций

Циклы можно создавать посредством присоединения Потока операций к предшествующему графическому элементу диаграммы. Графический элемент считается предшествующим в случае, если имеет Исходящий поток операций, ведущий к другим Потокам операций, последний из которых оказывается Входящим потоком операций для исходящего элемента. Это значит, что такой графический элемент создает Токен, который пересекает совокупность Потоков операций и возвращается к создавшему его графическому элементу. Такой цикл Потока операций представляет собой Модель технологического процесса №6 – Случайный Цикл (см. фигуру 10.40).



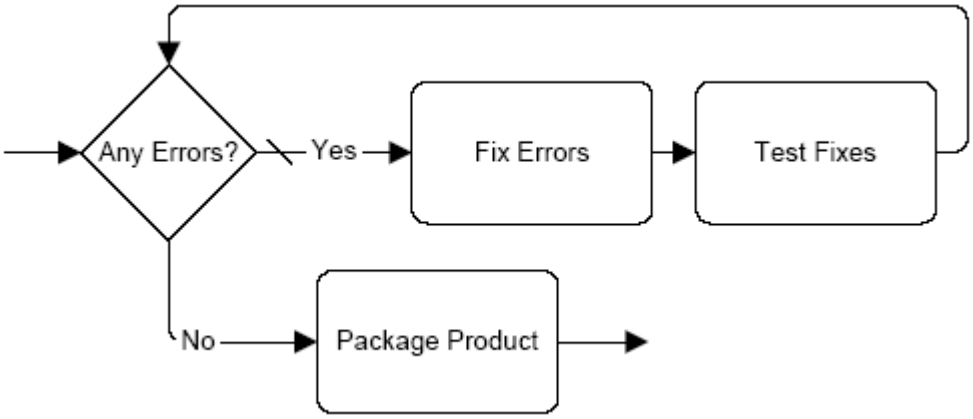
Фигура 10.40 - Модель технологического процесса №6 – Случайный Цикл

Обычно такие соединения Потоков операций следуют за Условиями, благодаря чему циклы не являются бесконечными (см. фигуру 10.41). В случае, если Поток операций направлен непосредственно от Условия к предшествующему графическому элементу, то такой цикл относится к циклам «до». Совокупность цикличных действий будет иметь место в Процессе до тех пор, пока значение определенного условия не будет равно «true».



Фигура 10.41 – Цикл «до»

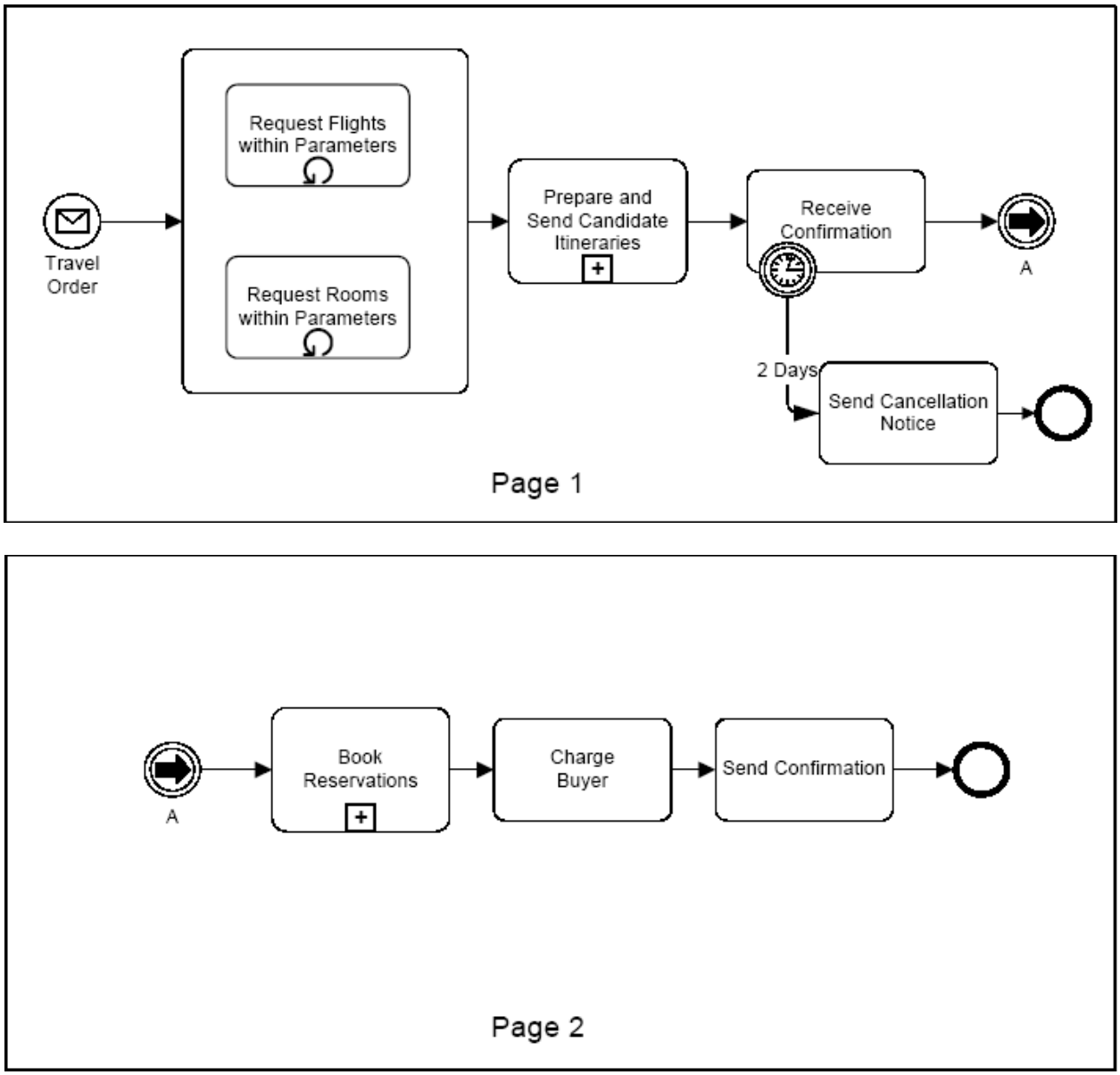
Цикл «пока» создается, во-первых, посредством принятия решения, а затем и выполнения повторяющихся действий или перехода в Процессе (см. фигуру 10.42). Совокупность цикличных действий может либо вообще не встречаться в Процессе, либо встречаться в нем несколько раз.



Фигура 10.42 – Цикл «пока»

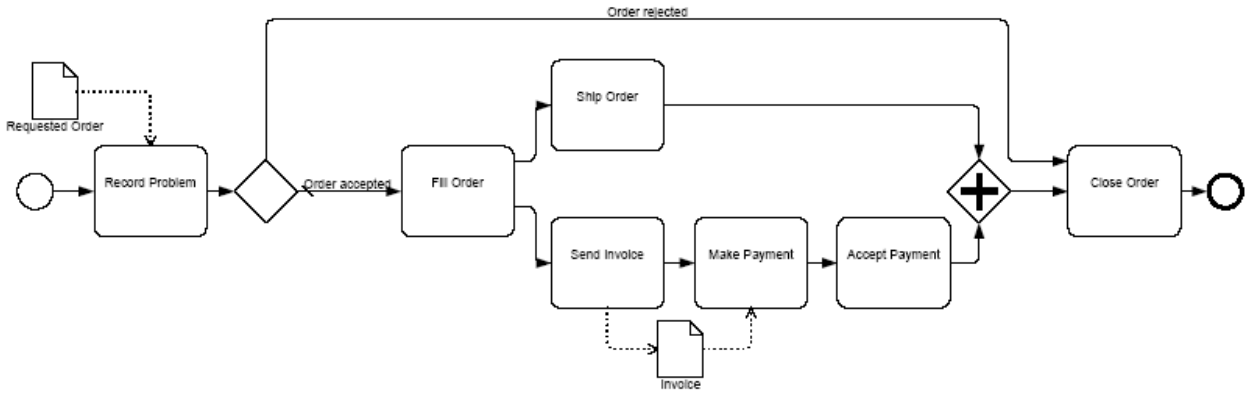
Перемещающийся поток операций. Соединители страниц и элементы «Переход к» (Sequence Flow Jumping. Off-Page Connectors and Go To Objects)

Т.к. модели Процессов часто выходят за пределы одной печатной страницы, то возникает необходимость показать то, каким образом соединения Потоков операций переносятся при переходе на новую страницу. Предоставляемое спецификацией BPMN и часто используемое решение данной проблемы заключается в использовании Соединителя страниц, который указывают на то, где заканчивается одна страница и начинается другая. Данная спецификация в качестве Соединителя страниц предлагает использовать Промежуточное событие типа «Связь» (см. фигуру 10.43. Обратите внимание, что данная фигура содержит изображения двух разных печатных страниц, а не двух Пулов одной диаграммы). Используется пара Промежуточных событий «Связь». Одно из них указывает на конец одной страницы, имеет название, а также соединено с Входящим потоком операций и ни с одним Исходящим. Второе Промежуточное событие находится в начале следующей страницы, имеет то же самое название, что и предыдущее, а также Входящий поток операций и ни одного Исходящего.

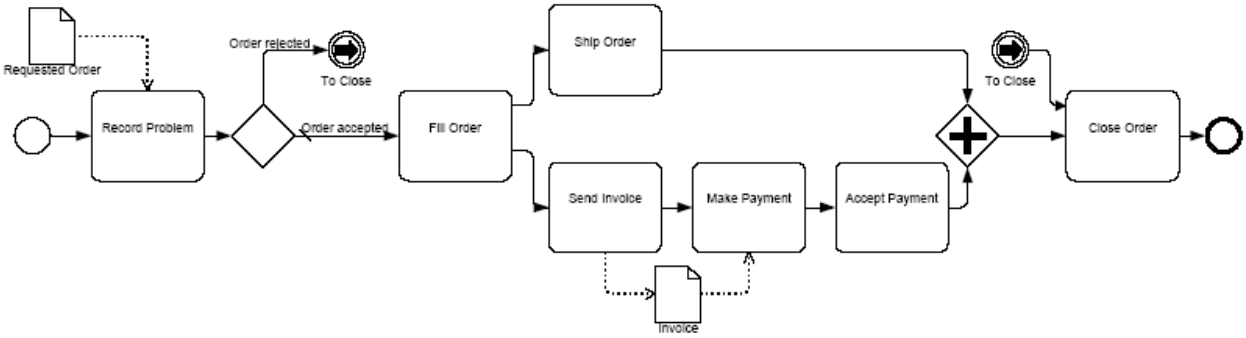


Фигура 10.43 – Промежуточное событие «Связь», используемое в качестве Соединителя страниц

Промежуточные события типа «Связь» также используются в качестве элементов «Переход к». С функциональной точки зрения, они действуют, как Соединителя страниц (см. описание выше), однако, могут быть помещены в любое место на диаграмме: могут располагаться на одной или на нескольких страницах. В целом, элементы «Переход к» предоставляют механизм сокращения длины линий Потока операций. Разработчики моделей могут посчитать длинные линии Потоков операций сложными для отслеживания. Элементы «Переход к» могут быть использованы для избежания использования слишком длинных Потоков операций (см. фигуры 10.44 и 10. 45). Диаграммы, изображенные на обоих рисунках, поведут себя одинаково. Если маршрут «Order Rejected», изображенный на фигуре 10.45, берет начало из Условия, то Токен, пересекающий Поток операций, достигнет исходного Промежуточного События «Связь» а затем «перепрыгнет» к целевому Промежуточному Событию «Связь» и продолжит свое движение в соответствии с направлением Потока операций. Такой Процесс продолжается, как если бы Поток операций был соединен с двумя этими элементами непосредственно.

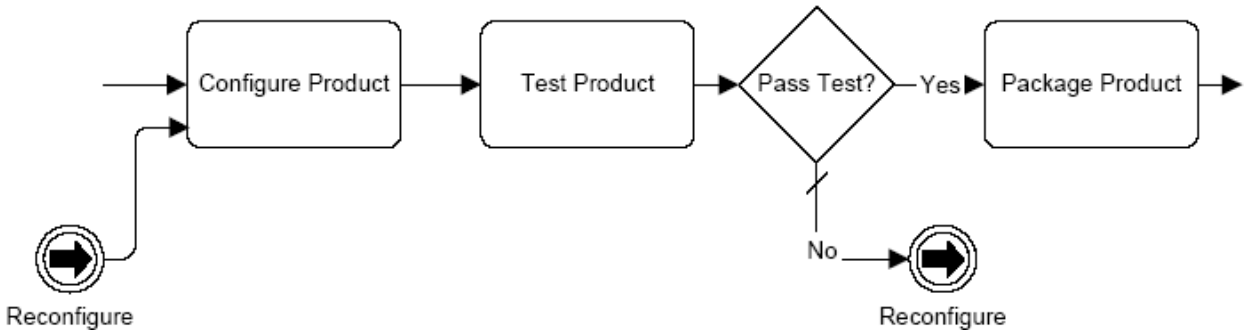


Фигура 10.44 – Процесс, содержащий длинный Поток операций



Фигура 10.45 – Процесс с Промежуточным событием «Связь», используемым в качестве элемента «Переход к»

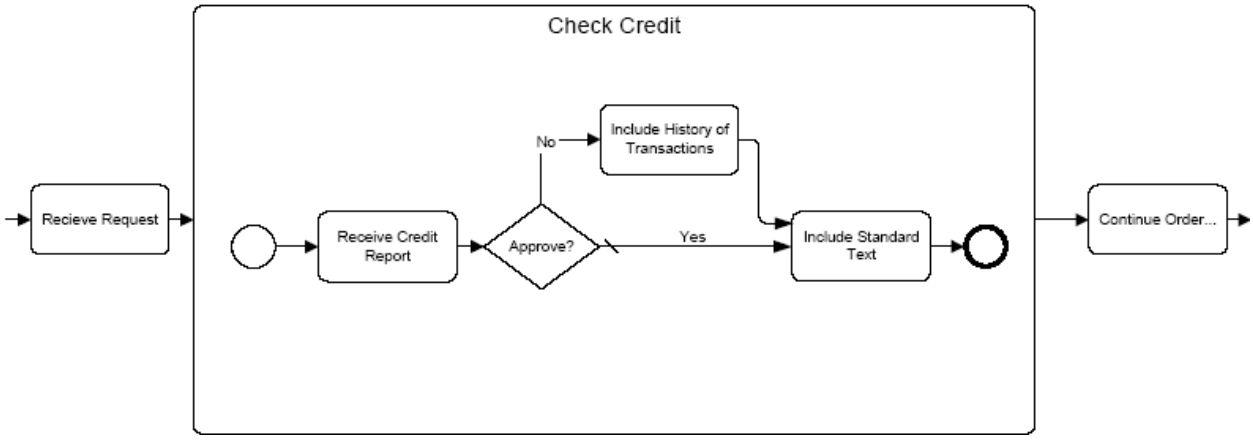
Некоторые методологии содержат утверждение о том, что Потоки операций могут двигаться лишь в одном направлении – вперед, и не позволяют Потокам операций соединяться непосредственно с предшествующими элементами диаграммы. Посредством такой методологии в моделировании достигается определенная последовательность, однако, все меняют ситуации, требующие использование циклов. Промежуточное событие «Связь» может быть использовано для соединения предшествующих элементов диаграммы, а также для создания циклов, без нарушения ограничений, предъявляемых к выбору направления Потока операций (см. фигуру 10.46).



Фигура10.46 – Промежуточное событие «Связь», используемое для создания цикла

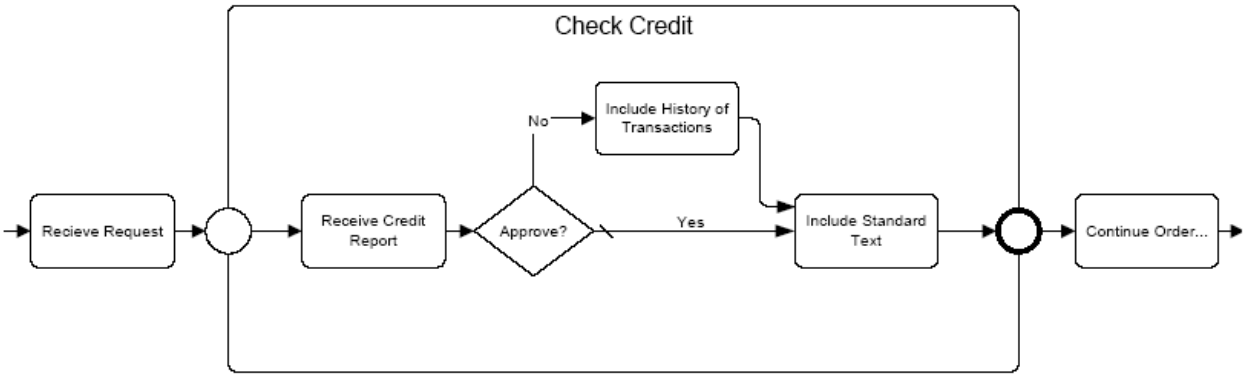
Поток операций, курсирующий к подпроцессу и обратно (Passing Flow to and from Sub-Processes)

Данный раздел содержит информацию о том, каким образом Поток операций курсирует от родительского Процесса к Подпроцессам, входящим в данный Процесс, и обратно. Поток операций (например, Токен) берет начало в родительском Процессе, перемещается в Подпроцесс, а затем возвращается в родительский Процесс (см. фигуру 10.47). Чаще всего Поток операций достигает Подпроцесса, перемещается к Стартовому событию Подпроцесса, пересекает Поток операций данного Подпроцесса, достигает Конечного его события и в итоге перемещается обратно в родительский Процесс для того, чтобы продолжить свой путь в соответствии с направлением Исходящего потока операций Подпроцесса. В случае, если Подпроцесс содержит параллельные Потоки операций, то все Потоки операций должны завершиться до того, как Токен вернется обратно в родительский Процесс. Такая функциональность рассматривает Подпроцесс в качестве независимого «блока» действий.



Фигура 10.47 – Пример Подпроцесса, включающего Стартовое и Конечное события

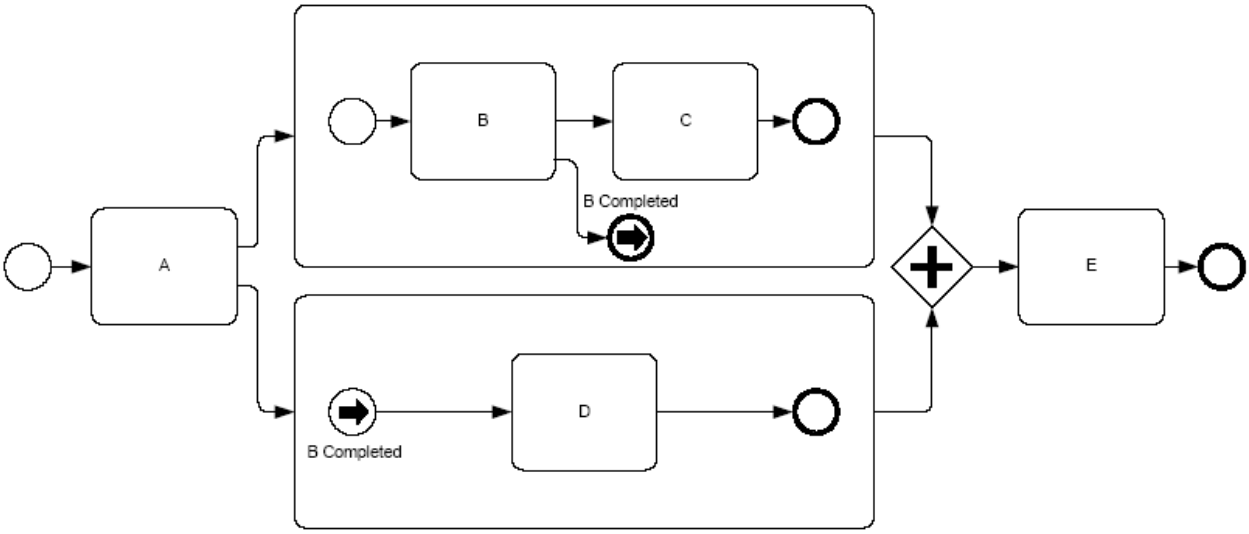
Для того, чтобы сделать Поток операций, находящийся между двумя уровнями Процесса, более заметным, разработчики моделей могут воспользоваться опциями помещения Стартового и Конечного Событий на границах Подпроцесса, а также соединения Потоков операций графических элементов родительского Процесса с этими событиями (см. фигуру 10.48).



Фигура 10.48 - Пример Подпроцесса со Стартовым и Конечным событиями, помещенными на границах данного Подпроцесса

Контролирующий поток операций между процессами (Controlling Flow Across Processes)

Иногда в Процессе возникают ситуации, когда на Поток операций оказывает влияние действие, происходящее в другом Процессе, либо Поток операций зависит от данного действия. Такие ситуации или условия относятся к контрольным точкам (milestone) моделирования Процесса. Модель Процесса должна уметь распознавать такие контрольные точки и управлять ими. Это значит, что начало либо продолжение Процесса могут быть инициированы Событиями типа «Связь», через которые проходит Поток операций (Токены) между Процессами (см. фигуру 10.49). Такой тип Модели технологического процесса называется Контрольная точка.



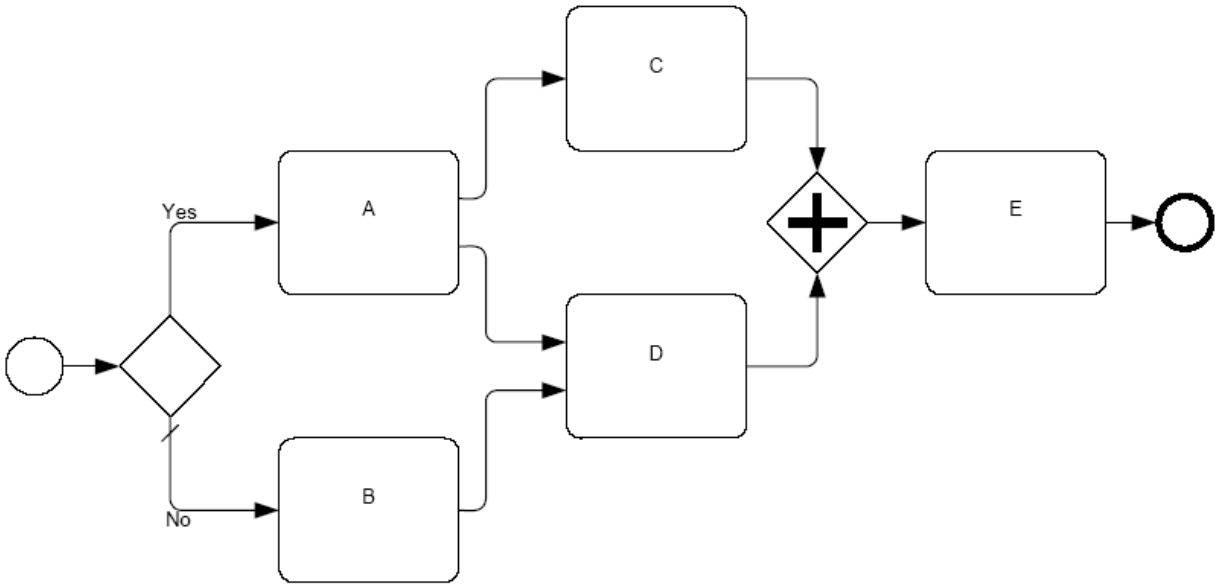
Фигура 10.49 – События типа «Связь», используемые для синхронизации поведения между Процессами

Избежание некорректных моделей и непредсказуемого поведения (Avoiding Illegal Models and Unexpected Behavior)

BPMN, будучи языком с графовой структурой, допускает значительную гибкость в изображении поведения комплексных Процессов в довольно сжатой форме, нежели BPEL4WS, имеющий блочную структуру. Однако эта особенность BPMN может спровоцировать создание таких ситуаций, которые не могут быть выполнены, или тех, которые поведут себя неожиданно для разработчика модели. Такие проблемы моделирования могут возникать из-за отсутствия строго определенных отношений между распределителями (forks) и соединителями (joins) или разделителями (splits) и

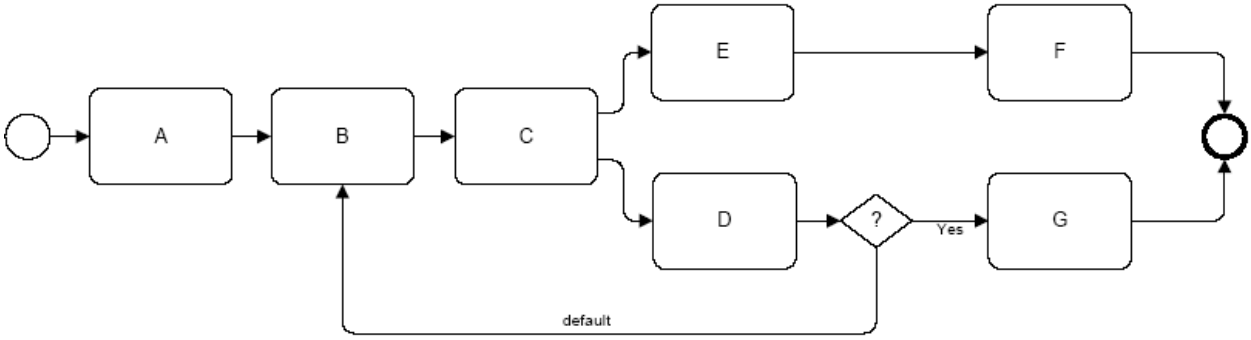
объединителями (merges). Блочная структура обеспечивает наличие таких строго определенных отношений, однако, графовая структура позволяет сочетать и сравнивать эти механизмы контроля Потоков операций, исходя из пожеланий разработчика модели. Некоторые комбинации такого рода управляющих элементов создают Процессы, которые не могут быть выполнены, а также поведение, не ожидаемое разработчиком модели. Ситуации, в которых альтернативные маршруты пересекают скрытую границу группы параллельных маршрутов, могут спровоцировать создание неверной модели.

Фигура 10.50 содержит изображение такого рода модели. Задача «D» представляет собой действие, имеющее два Входящих потока операций, один из которых выходит из разделенного маршрута (после разветвленного маршрута), а другой – из разветвленного маршрута. Такая ситуация создает проблему в Параллельном Шлюзе, предшествующем Задаче «E», которая также имеет несколько Входящих потоков операций. Поток операций, идущий от Задачи «B», пересекает скрытую границу разделения, распложенного после Задачи «A». Таким образом, в случае, если из Условия на диаграмме выбран Поток операций «Yes» (вариант 1), то Задача «E» может ожидать прибытия двух Токенов: одного от Задачи «C», а другого от задачи «D». Однако, в случае, если из Условия на диаграмме выбран Поток операций «No» (вариант 2), то к Параллельному Шлюзу подойдет лишь один Токен, направленный от Задачи «D». А т.к. Шлюз ожидает прибытия двух Токенов, то на данном этапе Процесс зайдет в тупик.



Фигура 10.50 – Потенциально тупиковая модель

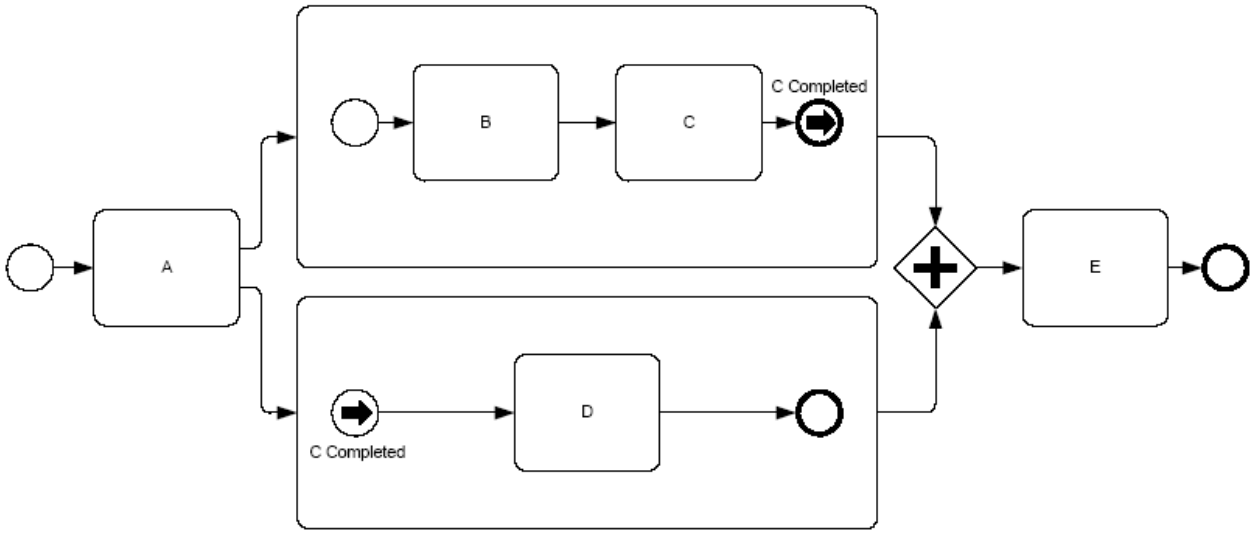
Другая проблема возникает тогда, когда происходит возврат к началу цикла предшествующих действий. В случае, если Условие цикла создано внутри скрытых пределов нескольких параллельных маршрутов, то поведение цикла в данном случае становится неоднозначным (см. фигуру 10.51), т.к. неясно, было ли задумано, что Задача «E» будет повторяться в соответствии с циклом, и что случится, если Задача «E» все ещё будет активной в момент следующего достижения данной Задачи циклом.



Фигура 10.51 – Неверный цикл

Использование Событий типа «Связь» также может спровоцировать создание непредсказуемого поведения. В целом, События типа «Связь», не используемые в качестве Соединителей страниц, должны рассматриваться как передовой технический прием моделирования, а разработчик модели должен использовать их с осторожностью, предполагая, каким может быть получаемое в результате поведение, а также ход Токенов.

Фигура 10.52 является вариантом фигуры 10.49, однако, в данном случае Конечное событие типа «Связь», расположенное в Подпроцессе верхнего уровня, используется неверно. В данном Подпроцессе верхнего уровня может быть создан и доступен лишь один Токен. Когда Токен отходит от Задачи «С» и подходит к Конечному событию «Связь», то оно расходует Токен, который, однако, затем немедленно переходит к целевому Стартовому событию, которое имеет то же самое название (внизу Подпроцесса). Т.к. Токен переходит в другой Подпроцесс, то для передачи родительскому Процессу, а также для продолжения Исходящего потока операций Подпроцесса верхнего уровня не остается ни одного Токена. Таким образом, весь Процесс будет приостановлен у Параллельного Шлюза в ожидании Токена, который, однако, никогда не появится.

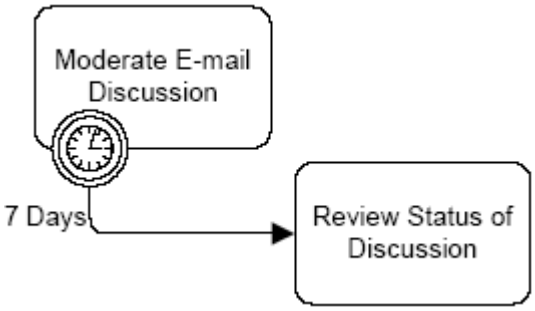


Фигура 10.52 – Неверное использование Конечного события «Связь»

В целом, анализ того, каким образом Токены проходят через модель, поможет выявить те модели, которые не могут быть выполнены верно. Такой анализ движения Токенов используется для создания некоторых соответствий с BPEL4WS. Т.к. BPEL4WS является верно выполняемым, то, если анализ движения Токенов не может создать верный BPEL4WS процесс, значит, модель считается неверно структурированной.

§§ 10.2.2. Поток исключений (Exception Flow)

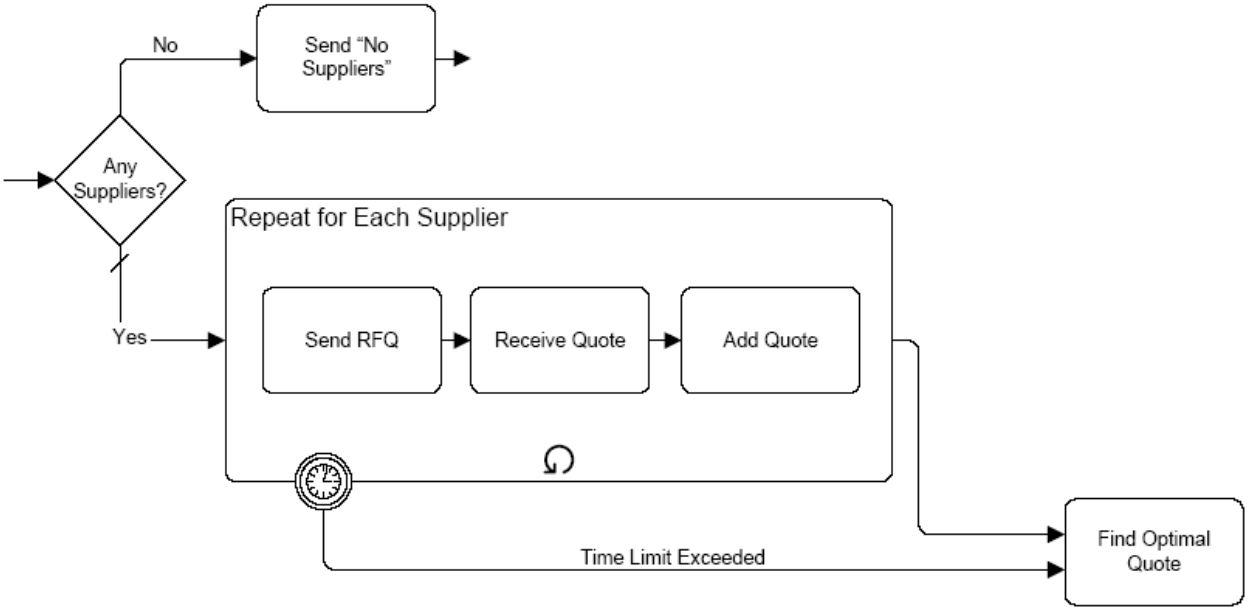
Поток исключений возникает за пределами Стандартного потока операций данного Процесса и основывается на событии (Промежуточном событии), которое происходит в ходе выполнения данного Процесса. Наряду с тем, что Промежуточные события могут быть включены в Стандартный поток операций для указания приостановки хода Процесса или его разрыва в ожидании получения сообщения, когда они могут присоединиться к границам действия (Задачи или Подпроцесса, см. фигуру 10.53), Промежуточные события также могут создавать Поток исключений.



Фигура 10.53 – Задача с Потоком исключений (прерывающим фрагмент События)

Благодаря использованию Потока исключений, разработчик модели создает фрагмент События. Фрагмент события реагирует на определенные Триггеры, направленные на прерывание хода действия, а также на переадресацию Потока операций посредством Промежуточного события. Однако он будет реагировать лишь в том случае, если является активным (запущенным) во время действия Триггера. В случае, если действие было завершено, то Триггер может остаться без ответного действия. Источник Триггера может быть внешним по отношению к выполняемому Процессу, таким, как сообщение или ошибка приложения, а также может возникать вследствие перемещения Промежуточного события из какого-либо другого активного участка Процесса.

В случае, если существует группа Задач, которая должна быть включена разработчиком модели в данный фрагмент События, то может быть добавлен Подпроцесс, направленный на выполнение этих Задач и каких-либо событий путем их присоединения к границам данного Подпроцесса (см. фигуру 10.54).



Фигура 10.54 – Подпроцесс с Потоком исключений (прерывающим фрагмент События)

Два Триггера, Сообщение и Ошибка (fault), предназначенные для Промежуточного события, используются фрагментом События на уровне языка исполнения (BPEL4WS). Событие «Сообщение» возникает тогда, когда Процесс получает сообщение, содержащее точное тождество, определенное в Промежуточном событии. Событие «Ошибка» возникает тогда, когда в Процессе обнаруживается ошибка. В случае, если в Промежуточном событии указан код ошибки, то для реакции фрагмента События код обнаруженной Ошибки должен соответствовать данному фрагменту События. Если же в Промежуточном событии код ошибки не указан, то реакцию фрагмента События может вызвать любая Ошибка.

Другие Триггеры, указанные в спецификации BPMN, такие, как Таймер, должны быть конвертированы в соответствии с конфигурацией BPEL4WS, которая создает соответствующее Сообщение или соответствующую Ошибку (дополнительную информацию о соответствии Потока исключений конфигурации BPEL4WS см. в разделе 11.13 «Поток Исключений», стр. 182 оригинала спецификации).

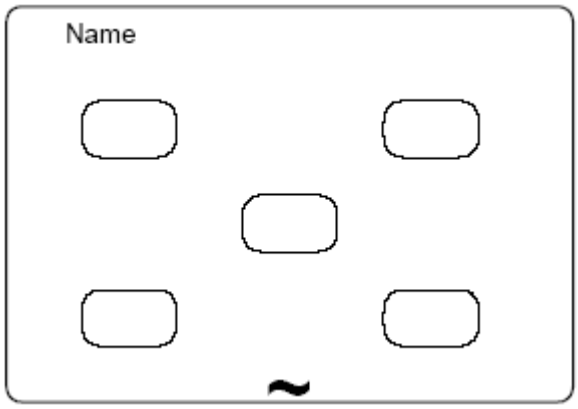
В случае, если при готовности фрагмента События не возникает инициирующего события, то Процесс продолжает свой ход в соответствии со Стандартным потоком операций, как и было указано Потоком операций.

§§ 10.2.3. Произвольный процесс (Ad Hoc)

Произвольный Процесс представляет собой группу действий, имеющих последовательные отношения, которые не могут быть предопределены. Действия, составляющие эту группу, могут быть определены для данного Процесса, однако, последовательность и количество выполнений этих действий полностью зависят от их исполнителя и не могут быть определены заранее.

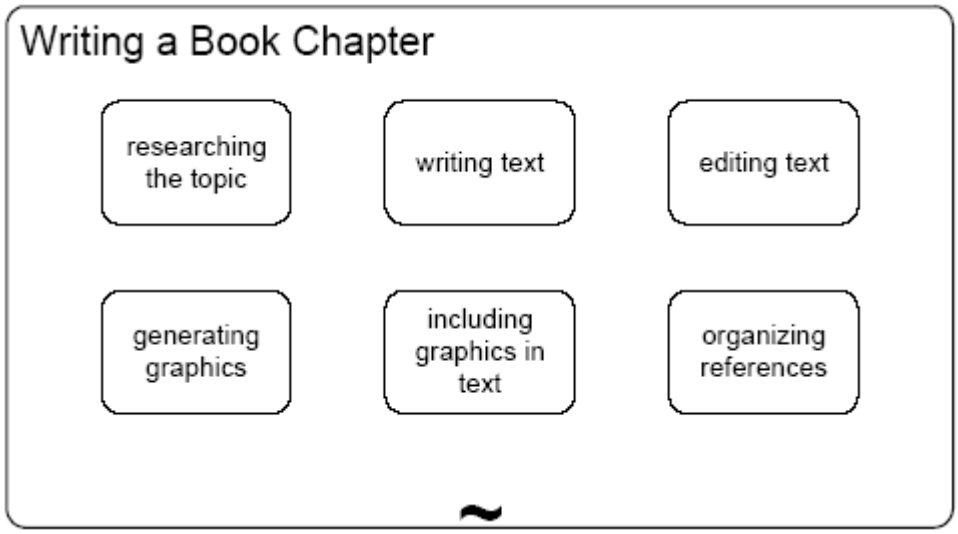
Подпроцесс, будучи произвольным, имеет маркер - значок «тильды», расположенный в центре нижней части графического элемента Подпроцесса (см. фигуры 10.55 и 10.56). Действия, находящиеся в Подпроцессе, отделены друг от друга. В ходе выполнения Процесса могут быть активными одно или более действий, которые могут выполняться практически в любой последовательности и с любой частотой.

Фигура 10.55 – Свернутый Произвольный Подпроцесс



Фигура 10.56 – Развернутый Произвольный Подпроцесс

Исполнитель действий определяет то, когда начнут выполняться действия, когда они завершатся, каким будет следующее действие и т.д. Примерами Произвольных Процессов могут служить создание машинного кода (на нижнем уровне), поддержка клиентов, а также написание главы книги. Если взглянуть на детали процесса написания главы книги, то можно увидеть, что действиями такого Процесса являются: исследования вопроса, написание текста, его редактирование, создание графических данных, внедрение графических данных в текст, составление библиографического списка и т.д. (см. фигуру 10.57). В таком Процессе может существовать зависимость Задач друг от друга, как, например, написание текста происходит до его редактирования, однако, экземпляры Задач написания и редактирования текста не обязательно должны быть взаимосвязаны. Редактирование может происходить достаточно редко и будет основываться на многих экземплярах Задачи написания текста.



Фигура 10.57 – Произвольный Процесс написания главы книги

В механизме моделирования бизнес-процессов существует проблема отслеживания статуса произвольного Процесса, который, как правило, выполняется при помощи группового программного обеспечения (например, с помощью электронной почты). Однако спецификация BPMN позволяет моделировать Процессы, возможность выполнения которых не является обязательным условием, и которые должны обеспечивать механизмами те средства моделирования бизнес-процессов, которые могут следовать за Произвольными Процессами. При этом Процесс будет завершен, что будет определено вычислением значения Условия Окончания, которое, в свою очередь, вычисляет значения атрибутов Процесса, обновленных действием Процесса.

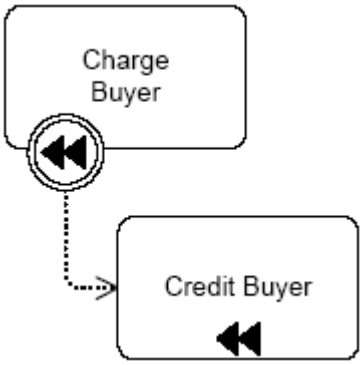
§ 10.3. Компенсирующая ассоциация

Некоторые действия приводят к сложным последствиям или создают определенные выходы. В случае, если, согласно определенным критериям (как, например, отмененный заказ), результат определен как нежелательный, то необходимо отменить эти действия. Существует три способа отмены действий:

- Восстановление копии внутренних значений данных, т.е. перезапись каких-либо изменений.
- Отсутствие действий (в случае, если ничего не было изменено, т.к. изменения были сделаны отдельно до подтверждения).
- Вызов действий (также называемых компенсацией), отменяющих результат.

Действие, которое может нуждаться в компенсации и представлять собой, например, оплату покупателем какой-либо услуги с использованием дебетовой карточки. Для такого рода действий обычно необходимо отдельно взятое действие, направленное на возвращение результата исходящего действия. Часто происходит так, что необходима запись обоих действий, что, таким образом, является ещё одной причиной незавершенности действия. Промежуточное событие «Компенсация» присоединяется к границам действия для того, чтобы указывать на то, что данному действию может потребоваться компенсация.

Компенсация, будучи одним из трех механизмов возврата действия, требует специальной нотации и является особым обстоятельством, возникающим за пределами Стандартного потока операций Процесса. Именно поэтому Промежуточное событие «Компенсация» не соединено с Исходящим потоком операций, однако, имеет исходящую Ассоциацию (см. фигуру 10.58).



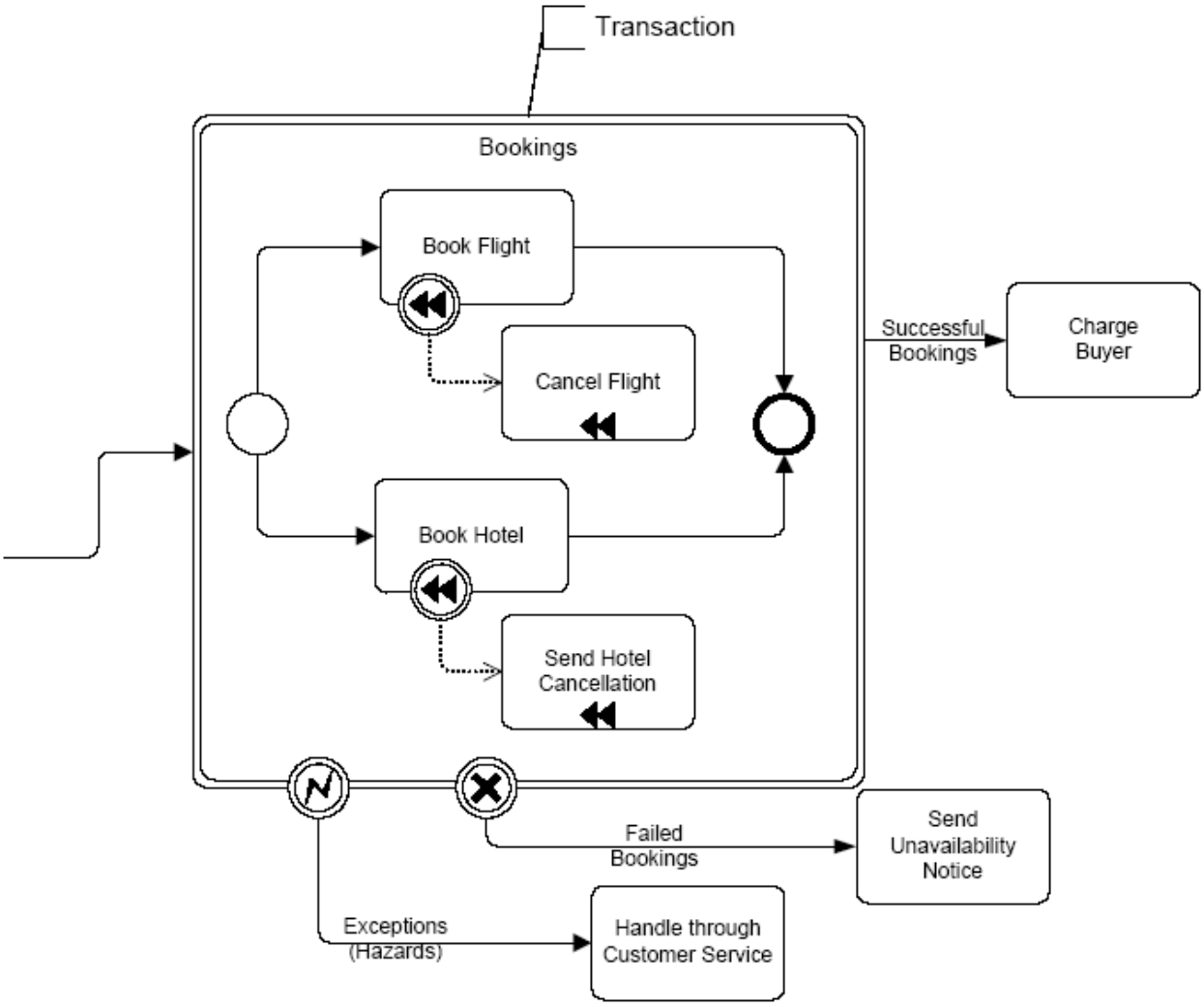
Фигура 10.58 – Задача с действием Компенсирующая Ассоциация

Целью такой Ассоциации является действие, компенсирующее работу, проделанную целевым действием, и относящееся к действию Компенсации. Действие Компенсация является уникальным, т.к. оно не соответствует требованиям Стандартного потока операций, а, как упоминалось выше, находится за его пределами. Такого рода действие не может иметь Входящих или Исходящих потоков операций. Маркер Компенсации (аналогичный маркеры Промежуточного события «Компенсация») отображается в центре нижней части графического элемента Действия для того, чтобы указать статус данного Действия (Задача «Credit Buyer» на фигуре 10.58). Обратите внимание, что для осуществления компенсации необходимо лишь одно целевое действие. Последовательность действий отображаться не должна. В случае, если компенсация требует наличия более одного действия, то эти действия должны быть помещены в один Подпроцесс, являющийся целевым для данной Ассоциации. Подпроцесс может быть как свернутым, так и развернутым. В случае, если Подпроцесс является развернутым, то добавление маркера Компенсации требуется лишь в сам Подпроцесс; действия, включенные в состав Подпроцесса, не требуют использования этого маркера.

Компенсированы могут быть только выполненные действия. Компенсация действия может быть вызвана двумя следующими способами:

- Действие, находящееся в транзакционном Подпроцессе, является отмененным (см. фигуру 10.59). В данном случае весь Подпроцесс будет «перемотан» или возвращен в начало: Поток операций Процесса начнет движение назад, а любое действие, которому необходима компенсация, будет компенсировано. Именно поэтому маркер Компенсации, предназначенный для Событий, выглядит как значок перемотки назад на проигрывателе. После завершения компенсации данный Процесс продолжит движение назад.

- Промежуточное или Конечное события типа «Компенсация», расположенные далее, «выбрасывают» идентификатор компенсации, который используется Промежуточным событием, присоединенным к границам действия.



Фигура 10.59 – Компенсация, использованная в Транзакции