

## Лекция № 9. Визуальное моделирование бизнес-процессов.

*В этой лекции рассматривается понятие бизнес-процесса, обсуждается язык описания бизнес-процессов BPMN, приводится пример бизнес-процесса покупки мебели в магазине, а также проводится его анализ. Затрагивается также вопрос циклической разработки (round-trip engineering) визуальных моделей. Затрагиваются вопросы исполняемой семантики бизнес-процессов и их связь с Web-сервисами.*

**Новая концепция бизнеса – ориентация на бизнес-процессы.** В 70-80-х годах началось массовое снижение конкурентоспособности американских бизнес-компаний. В частности, японские компании стали успешно конкурировать с американскими прямо на внутреннем рынке США. В связи с этим в начале 90-х годов появилось новая стратегия организации бизнеса, основанная на поддержке процессов. Вошли в обиход такие термины как бизнес-процесс (business process), реинжиниринг бизнеса (business reengineering), реинжиниринг бизнес-процессов (business process reengineering), моделирование бизнес-процессов (business process modeling).

До этого момента в бизнесе господствовала идея функционального разделения труда. Эту идею в конце 18-го века впервые сформулировал Адам Смит, на ее основе были созданы мануфактуры, которые в 19-м веке вытеснили ремесленные цеха и кустарное производство товаров. В начале 20-го века Генри Форд усовершенствовал эту идею и создал сборочный конвейер на своих автомобильных заводах, что позволило сильно увеличить производительность труда. Сейчас такие конвейеры существуют во многих отраслях промышленности. После этого Альфред Стоун, руководитель компании «Дженерал моторс», применил идею разделения труда к управлению крупным производством.

В начале 90-х годов Майкл Хаммер и Джеймс Чампли предложили иную форму организации бизнеса, ориентированную на процессы (бизнес-процессы). *Бизнес-процесс* – это цепочка действий от поступления заказа до полного завершения его обработки. Такой подход позволяет управлять и контролировать бизнес как деятельность, нацеленную на конечный результат. Иначе заказы и сервисы оказываются «размазанными» по функциональным отделам компании, у каждого из которых нет заинтересованности в конечном результате. В итоге падает качество сервисов, заказы обрабатываются не оптимально, с большими издержками.

**Моделирование бизнес-процессов.** Переориентация компаний на бизнес-процессы (бизнес-реинжиниринг) означает переделку внутренней работы и системы управления компаний. Это тяжелая и болезненная процедура, которая сопровождается ломкой привычных рабочих укладов работников компании, пересмотром их обязанностей, уменьшением/увеличением их зарплат и часто – увольнениями. Здесь много разной работы, и среди прочего – проектирование новых схемы бизнеса, то есть различных бизнес-процессов. В крупной компании их может быть много, они могут быть устроены иерархично, а также взаимодействовать друг с другом.

Далее мы сконцентрируемся на моделировании бизнес-процессов, что включает в себя не только тщательном продумывании новых правил бизнеса, но также и их формализацию. Почему хорошая формализация важна?

1. Она позволяет сделать наши мысли предметом широкого обсуждения.
2. Формализация позволяет донести новые правила работы до тех сотрудников, которые должны будут их выполнять.
3. Формализованные бизнес-процессы легче изменять и модернизировать.
4. Формализация бизнес-процессов является хорошей основой для последующей автоматизации бизнеса в компании, то есть создания/настройки различных информационных систем, а также для настройки стандартных пакетов автоматизации.

На сегодняшний день существуют стандартные системы комплексной автоматизации бизнеса под названием ERP-системы (Enterprise Resource Planning). Лидерами в этой области являются системы компании SAP, Oracle, Microsoft и пр. ERP-система пытается «воспроизвести» бизнес-процессы компании в программном обеспечении и сопровождать каждое действие того или иного сотрудника. ERP-система представляет собой набор стандартных модулей, например, главная книга, расчеты с кредиторами, расчеты с дебиторами, основные средства, преобразователь проводок, склад, управление закупками. Для каждой компании производится настройка некоторого числа модулей на некоторое количество пользователей (и тот и другой параметр сильно влияют на стоимость системы).

Важной частью настройки ERP-системы является формализация бизнес-процессов компании – именно эти бизнес-процессы погружаются в стандартные компоненты, то есть служат основой их настройки.

Преимущества таких систем очевидны – бизнес-компания, в виде ERP-системы получают интегрированные решения для своего бизнеса: разные их подразделения и филиалы будут связаны вместе единой системой учета, контроля, будут иметь доступ к единому банку данных и т.д.. Недостатком ERP-систем является высокая стоимость (по сравнению с ценой готовых систем, решающих какие-либо частные задачи бизнеса), а также высокая цена на их внедрение.

Нетрудно догадаться, что в качестве средств формализации предлагаются визуальные модели. Предпочтения этого способа перед обычными текстами и текстовыми формальными языками традиционны – людям тяжело читать большие тексты, но они легко обсуждают диаграммы. В то же время диаграммы являются достаточно формальными описаниями, на них можно пошагово определить виды действий, участников и результаты, а также сделать основой автоматизации.

**Пример бизнес-процесса.** В качестве примера рассмотрим крупный магазин по торговле мебелью. Опишем процесс покупки клиентом товара, представленный на рис. 9.1.

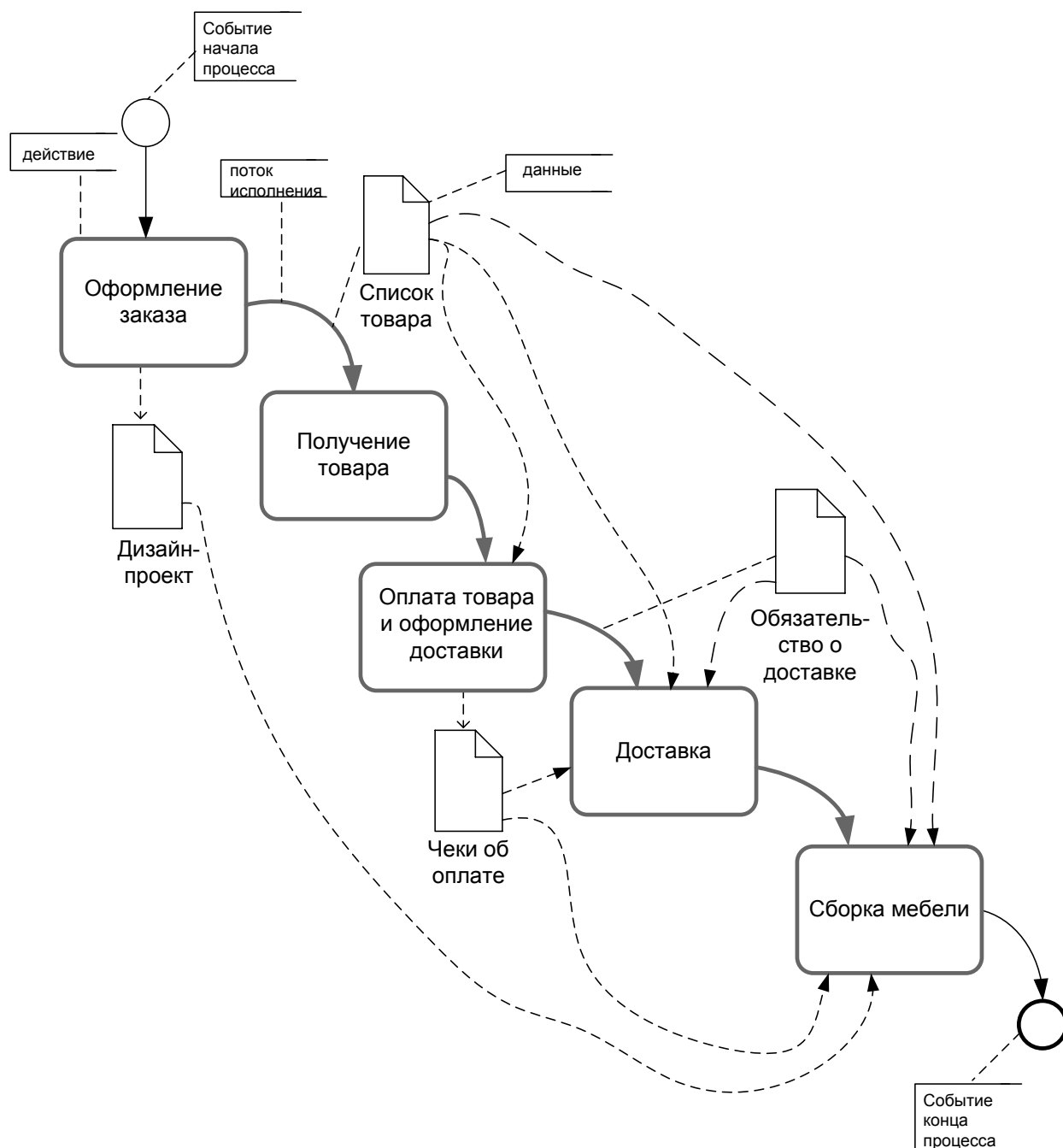


Рис. 9.1.

Весь бизнес-процесс разбит на действия, которые изображаются прямоугольниками со скругленными углами. Переходы между действиями показаны стрелками, документы, которые порождаются в результате исполнения каждым действием, показаны прямоугольниками с загнутым правым углом. Эти прямоугольники соединены штриховыми линиями с тем действием, в результате которого они созданы и с теми действиями, в которых они используются.

«Оформление товара». Сначала клиент оформляет заказ. Предполагается, что перед этим он определился в главном – например, ему нужен кухонный гарнитур. Тогда в отделе по торговле кухонной мебелью он, вместе с одним из менеджеров этого отдела,

составляет дизайн-проект для своей покупки (в соответствии с размерами помещения, в котором у него в квартире находится кухня и своими пожеланиями) и уточняет параметры своего заказа, точно определяется с материалами.

«Получение товара» Потом он идет на склад и сам выбирает заказ все составные части своего заказа, имея точный перечень того, что ему нужно. При этом ему помогают работники склада.

«Оплата товара и оформление доставки» Затем клиент, вместе со своим выбранным товаром идет на кассу и оплачивает то, что он выбрал. Далее, с оплаченным товаром он переходит в отдел доставки, где оформляет и оплачивает доставку своей мебели, а также сборку (если ему это нужно) после этого уезжает домой.

«Доставка». Оплаченный товар клиенту доставляют в течение трех дней.

«Сборка». После этого, если клиент оформил сборку, то к нему приезжает мастер-сборщик и собирает доставленную мебель.

Проанализируем оптимальность работы приведенного выше бизнес-процесса. Этот анализ удобно проводить, имея представленную диаграмму с рис. 9.1. Более того, можно составить дополнительные диаграммы для детализации действий этого бизнес-процесса. Предложения по оптимизации можно также оформить в виде диаграмм. Итак, каждое действие поддерживает свой отдел магазина: оформлении заказа – отдел формирования заказа; выдача заказа – отдел выбора и выдачи товара, оплата товара – кассиры, оформление доставки – отдел доставки; доставка – отдел доставки; сборка мебели – подразделение сборки мебели.

Каждый из отделов работает со всеми заказами магазина. Вроде бы, все нормально, но качество сервиса получается не очень хорошим. Например, рассмотрим следующие истории, которые могут возникнуть в таком мебельном магазине:

- дизайнеры забыли про встроенное освещение, а клиент и не знал про него, ведь не каждый же день он покупает, например, мебель на кухню, а когда все смонтировали и установили, то стало очевидно, что освещение нужно;
- при оформлении доставки забыли просьбу клиента о том, что он просил доставить мебель до 13 часов – доставили к 15 часам, руководствуясь стандартным интервалом, принятым обычно, но никого уже не было дома; «расхлебывать» это опять таки пришлось клиенту;
- грузчики были небрежны и повредили мебель, это обнаружилось при сборке, когда грузчиков и след простыл, и бумага о том, что все в порядке клиентом была подписана; не мог же он при них развернуть каждый из 58-ми доставленных предметов, сложенных в виде внушительного штабеля; «расхлебывать» это пришлось опять таки клиенту; все, конечно, заменили, но ему пришлось самому везти в магазин поврежденную мебель и обратно везти новую;
- наконец, при сборке мебели сборщик попробовал взять второй раз деньги за то, что было уже оплачено при оформлении доставки; и снова клиенту пришлось внимательно читать многочисленные приложения к выданным ему в магазине документам, общаться по телефону с менеджерами и т.д.

Итак, очевидно, что магазин централизованно не отслеживает прохождение одного заказа, в результате возникают следующие проблемы:

- накладки при взаимодействии разных отделов;
- клиенту приходится самостоятельно организовывать весь процесс в целом – стоять в разных очередях, разбирать противоречия между работой отдельных отделов и пр., все время общаясь с разными менеджерами;
- каждый из отделов не заботит клиент и его заказ в целом, поэтому, стремясь выполнить только свою работу (короткий отрезок), они часто забывают об отдельном клиенте как таковом, усредняя его запрос и закладывая разные недоработки и ошибки, которые всплывают позднее и за которые трудно с кого-то спросить: трудно также спросить с отдела в целом, ведь часто не сохраняется даже имени менеджера, который работал с клиентом, а если и сохраняется, то обратиться к нему резонно, только если есть его очевидная ошибка, тем более, что ошибки часто не носят характер прямых претензий.

Большая централизация и отслеживание бизнес-процесса со стороны магазина в нашем случае существенно повысило бы качество сервиса. Однако это же может существенно поднять цену за услуги. Соотношение цена/качество услуги – важный оптимизационный критерий при улучшении бизнес-процессов. Важен также «вес» недовольства клиента – он велик, например, в условиях большой конкуренции поставщиков услуг или при обслуживании небольшого количества «дорогих» клиентов, приносящих компании ощутимую (или даже основную) прибыль. Однако, в крупных компаниях часто велики издержки от неоптимальных бизнес-процессов, от отсутствия в компании внимания к «судьбе» отдельных заказов. Часто даже оказывается, что как таковых устойчивых бизнес-процессов в компании нет вообще – все делается каждый раз по-разному. Формализация, наладка и оптимизация бизнес-процессов в компании оказывается существенным фактором успешности компании.

**Обзор языка BPMN.** Рассмотрим известный язык визуального моделирования бизнес-процессов – BPMN (Business Process Management Notation). Исходно он был стандартизован международным комитетом BPMI (Business Process Modeling Initiative, <http://www.bpmi.org>), первая версия стандарта вышла в 2004 году. Позднее этот стандарт перешел под эгиду комитета OMG и в 2006 году была выпущена первая OMG-версия этого стандарта.

В BPMN имеются следующие конструкции:

- сущности (flow subjects), которые могут быть следующих видов:
  - действия (activities),
  - порты (gateways),
  - события (events);
- связи (connecting objects), которые определяют поток исполнения бизнес-процесса – переход от одного действия к другому, обмен сообщениями, переходы между действиями в особых ситуациях (например, при возникновении исключений); кроме того, связи используются для «прикрепления» комментариев, данных и пр. статической информации к различным объектам BPMN-диаграмм; связи могут быть следующих видов:
  - поток исполнения (sequence flow),
  - поток сообщений (message flow),
  - ассоциации (association);
- участники (swimlanes) процесса, которые могут быть:
  - внешними по отношению к нему (pools) и
  - внутренним (lanes), например, функциональные отделы, по которым проходит бизнес-процесс;
- артефакты (artifact) процесса: данные (data object), группы (groups), комментарии (annotations).

Рассмотрим кратко главные из этих конструкций.

**BPMN: действия (activities).** Итак, бизнес-процесс состоит из цепочки действий. Действия бывают следующих видов:

- задача (task) – рис. 9.2 (а);
- свернутый подпроцесс (collapse subprocess) – рис. 9.2 (г);
- развернутый подпроцесс (subprocess expanded) – рис. 9.2 (е).

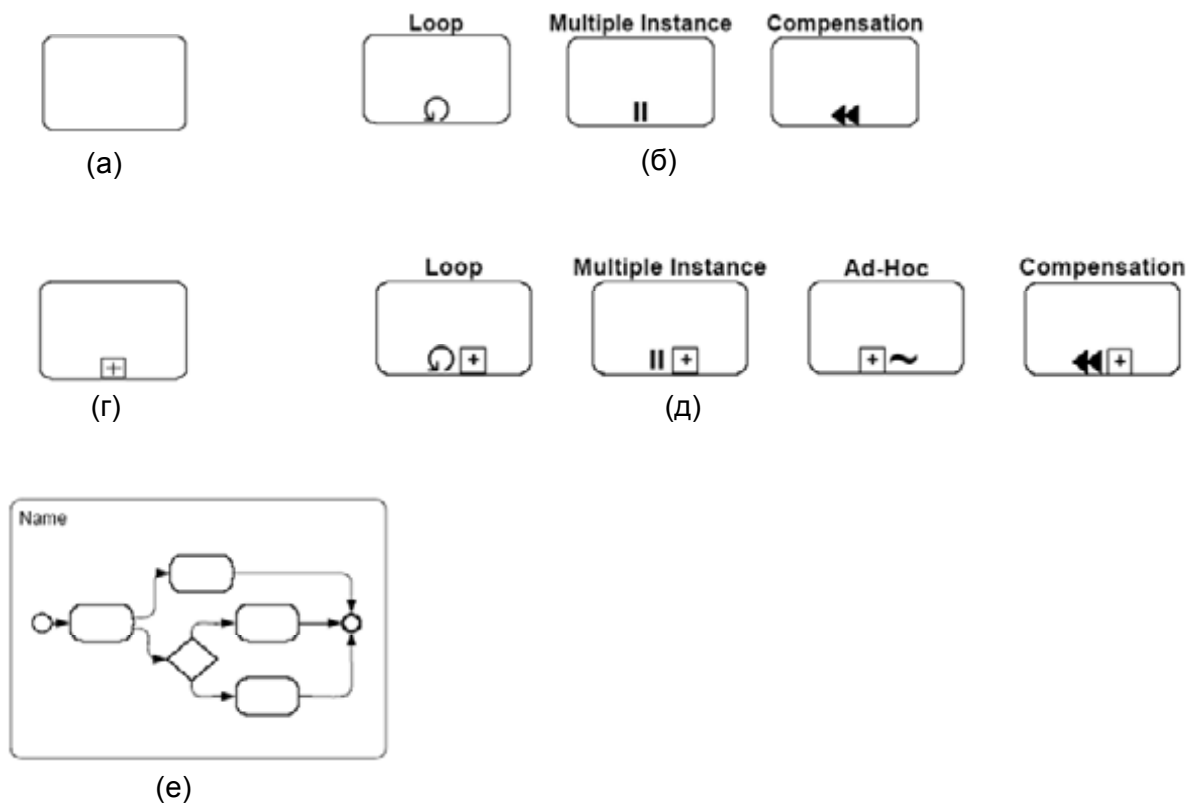


Рис. 9.2.

*Задача (task)* – это атомарное действие бизнес-процесса, не раскладываемое на более элементарные части. На диаграмме задача изображается, как показано на рис. 9.2 (а). На рис. 9.2 (б) приводятся три вида задач, которые могут быть заданы в BPMN – цикл, множественная задача и откат.

*Циклическая задача (loop)* – это задача, которая выполняется в цикле. В параметрах этой задачи можно указать, какой цикл имеется в виду – с пред или пост условием, определить это самое условие и указать некоторые дополнительные свойства цикла. обычный цикл с пред/пост условием, которое задается в тестовых параметрах задачи.

*Множественная задача (multiple instance)* – это циклическая задача, которая на самом деле выполняет в цикле целый набор однотипных задач. Условие цикла, количество однотипных задач, а также порядок их выполнения (последовательный или параллельный) задаются в текстовых параметрах множественной задачи.

*Откат (compensation)* – задача, которая вызывается в случае отклонения другой задачи – например, клиент отказался от забронированного отеля – тогда система должна освободить соответствующую бронь – пример приводится на рис. 9.3.

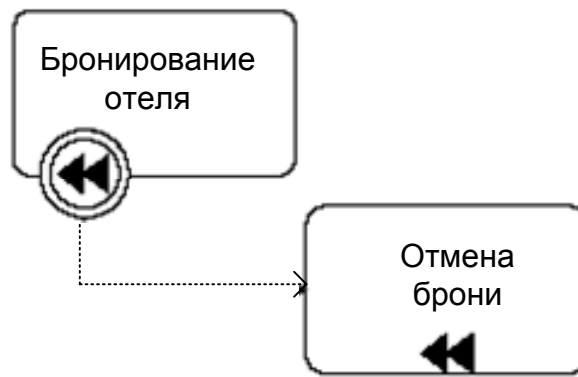


Рис. 9.3.

Кроме того, у задачи есть атрибут, который может иметь одно из следующих значений: Service, Receive, Send, User, Script, Manual, Reference, None. Эти значения не имеют графического представления и могут быть отражены, например, в имени задачи.

Еще одним типом действия является *подпроцесс* (subprocess) – способ декомпозиции сложных процессов на более мелкие. Подпроцессы бывают *свернутые* (collapsed subprocesses) – см. рис. 9.2 (г) и *развернутые* (expanded subprocesses) – см. рис. 9.2 (е). Также как и задачи подпроцессы могут быть циклическими, множественными и определять откат, но, кроме того, могут иметь еще маркер *произвольный* (ad hoc) – см. рис. 9.2 (д). Он означает, что задачи и другие подпроцессы, входящие в состав данного, исполняются в произвольном порядке.

Свернутый подпроцесс является ссылкой на другую диаграмму, где он определяется в виде задач и, возможно, других подпроцессов.

Развернутый подпроцесс позволяет иметь на диаграмме как бы второй этаж (а, возможно и третий и т.д. – все зависит от того, на сколько модель «глубока» и сколь большой мы хотим иметь нашу диаграмму) – то есть на родительской диаграмме один или несколько процессов детализированы, как показано на рис. 9.4.

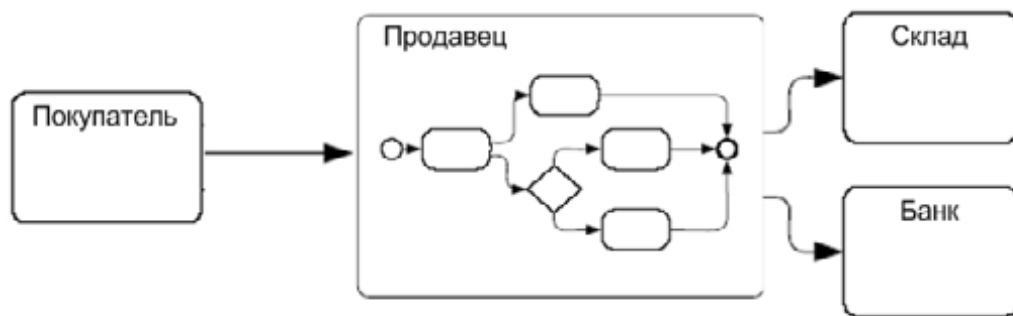


Рис. 9.4.



**BPMN: связи (connecting objects).** На рис. 9.5 показаны связи разного вида, существующие в BPMN:

- поток исполнения (sequence flow) – рис. 9.5. (а); это самый распространенный вид связи, с его помощью обозначается порядок выполнения действий бизнес-процесса;
- поток сообщений (message flow) – рис. 9.5. (б); с помощью этой связи определяются сообщения, которыми обмениваются действия; эти связи часто используются в контексте конструкций pool и swimlane, как будет рассказано ниже;
- ассоциации (association) рис. 9.5. (в), (г); ассоциации показаны также на рис. 9.1 – с их помощью данный соединяются с задачами и связями, а комментарии – с произвольными элементами диаграммы.

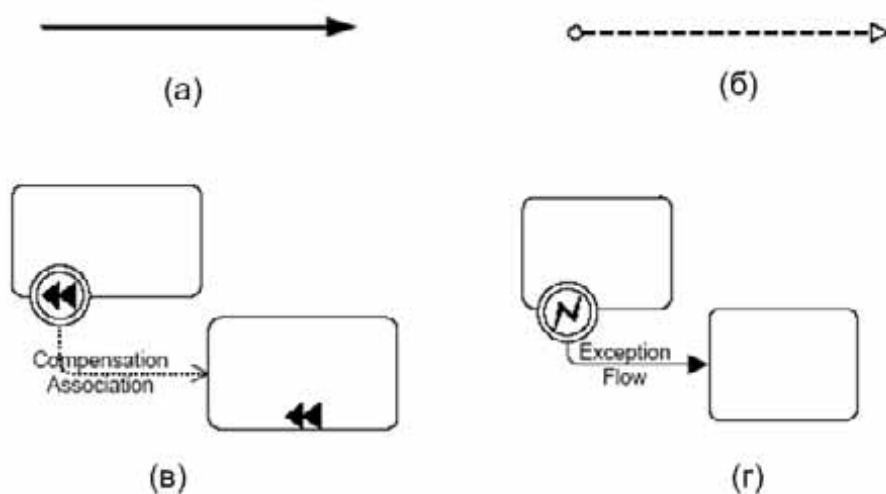


Рис. 9.5.

**BPMN: порты (gateways).** Теперь обсудим следующий вид сущностей – порты. Этот вид конструкций позволяет управлять потоком выполнения бизнес-процесса – ветвить его (в логическом смысле и в смысле распараллеливания) и соединять. Почти каждый вид порта может быть использован в двух вариантах – как *разветвитель* и как *соединитель*. Общий список портов BPMN показан на рис. 9.6.

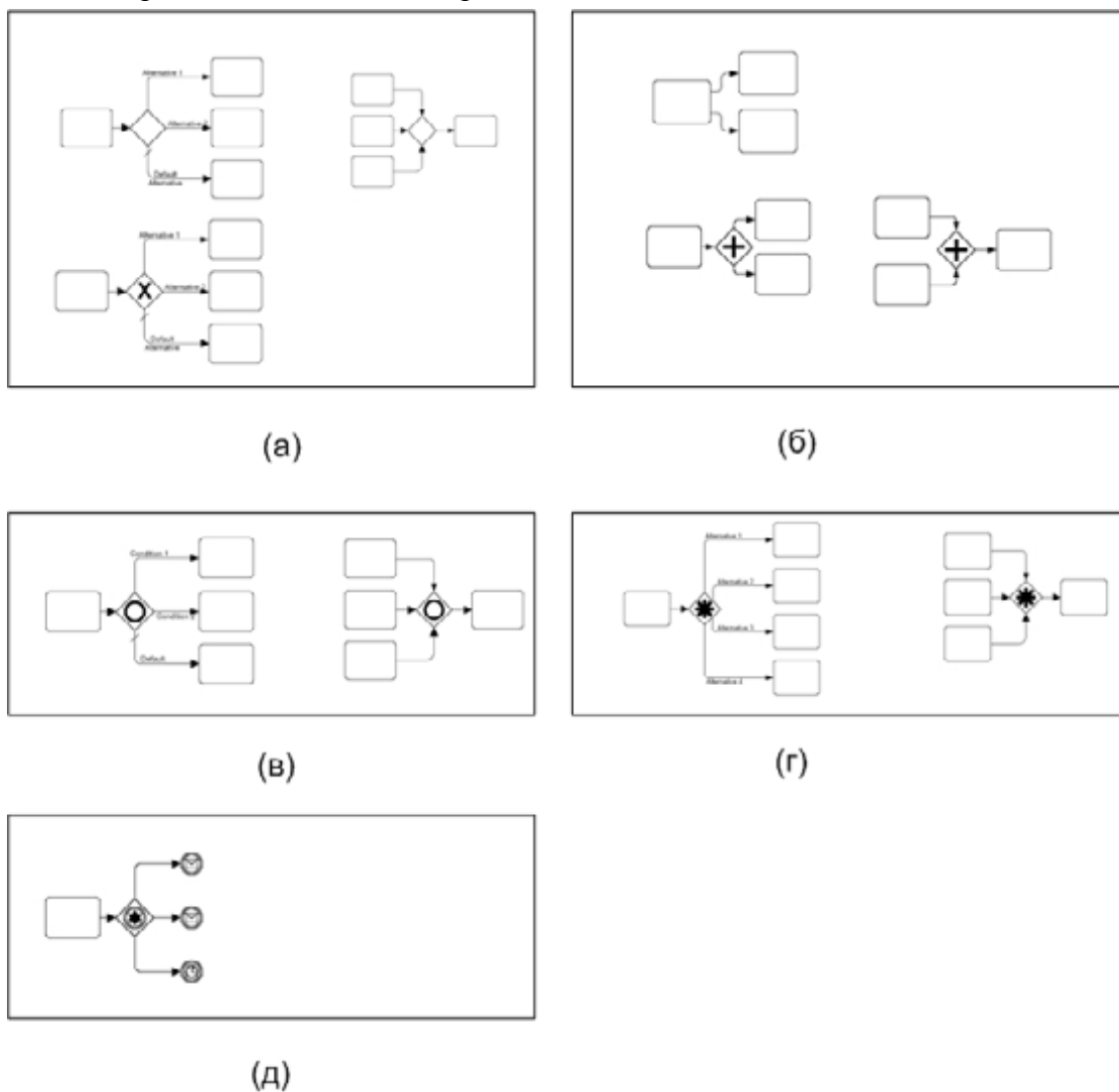


Рис. 9.6.

На рис. 9.6 (а) показан традиционный оператор логического ветвления по условию. BPMN предлагает два варианта его изображения – обычный ромбик и ромбик с крестиком внутри. Первый вариант удобен, если никаких других типов ромбиков на диаграмме нет, второй – если на диаграмме есть иные, экзотические ромбики – см. рис. 9.6 (б) и далее – дабы можно было их все отличать. Использование этой конструкции как соединителя означает, что все логически разошедшиеся ветки с места соединения сливаются. Например, пусть есть оператор switch с разными ветками, но вот он заканчивается, и какая бы ветка не выполнялась в этом операторе, далее поток управления одинаков для всех случаев.

На рис. 9.6 (б) показан оператор распараллеливания и соединения потоков управления. Как следует из этого рисунка, он может быть изображен – с ромбиком и без.

На рис. 9.6 (в) показан оператор, который разветвляет поток управления по всем веткам, условия которых оказались выполнены к моменту проверки. При использовании этого оператора в качестве соединителя он ждет все те потоки из множества направленных к нему, которые были до этого запущены. Ведь часть из тех потоков, которые показаны на диаграмме как входящие в него, могли быть не запущены (например, при использовании этого же оператора как разветвителя). В этом его отличие от предыдущего вида порта.

На рис. 9.6 (г) показан сложный разветвитель. Он введен для того, чтобы можно было задавать более сложную семантику ветвления потоков управления, чем было определено выше. Возможно, что новое условие будет некоторой комбинацией представленных выше операторов. Таким образом, этот оператор должен обязательно сопровождаться некоторым выражением, точно определяющим его семантику. То же самое можно сказать и про использование этого оператора как соединителя – требуется задать специальное выражение, которое определит условие, действующее на множестве все потоков, заданных на спецификации как входящие в этот оператор. Например, можно определить такой порт как соединитель, соединяющий на диаграмме три параллельных потока с условием, что он «пропускает» выполнение бизнес-процесса дальше, если дождался любых двух из трех потоков.

Наконец, на рис. 9.6 (д) показан разветвитель, который переключает поток управления в зависимости от получения того или иного события. Сами события обозначены в начале соответствующей ветки. В качестве соединителя этот оператор не используется.

**BPMN: события (events).** *Событие* – это некоторое происшествие вот время исполнения бизнес-процесса. Событиями могут быть инициация/завершение процесса, прием/посылка сообщения, завершение какой-либо задачи или подпроцесса и т.д. Не все события одинаково интересны, то есть достойны специального обозначения на диаграммах, но многие из них способны влиять на порядок выполнения процесса или на время исполнения его действий. Вот их-то в BPMN и предлагают специально выделить.

На диаграммах BPMN событие изображается, как показано на рис. 9.7 (а). Внизу, сразу под символом события, указывается его имя или источник. События бывают трех видов:

- начало (start) – событие, с которого начинается процесс или подпроцесс;
- промежуточное событие (intermediate), которое случается в «середине» жизненного цикла процесса;
- конечное событие (end), наступление которого означает завершение процесса или подпроцесса.

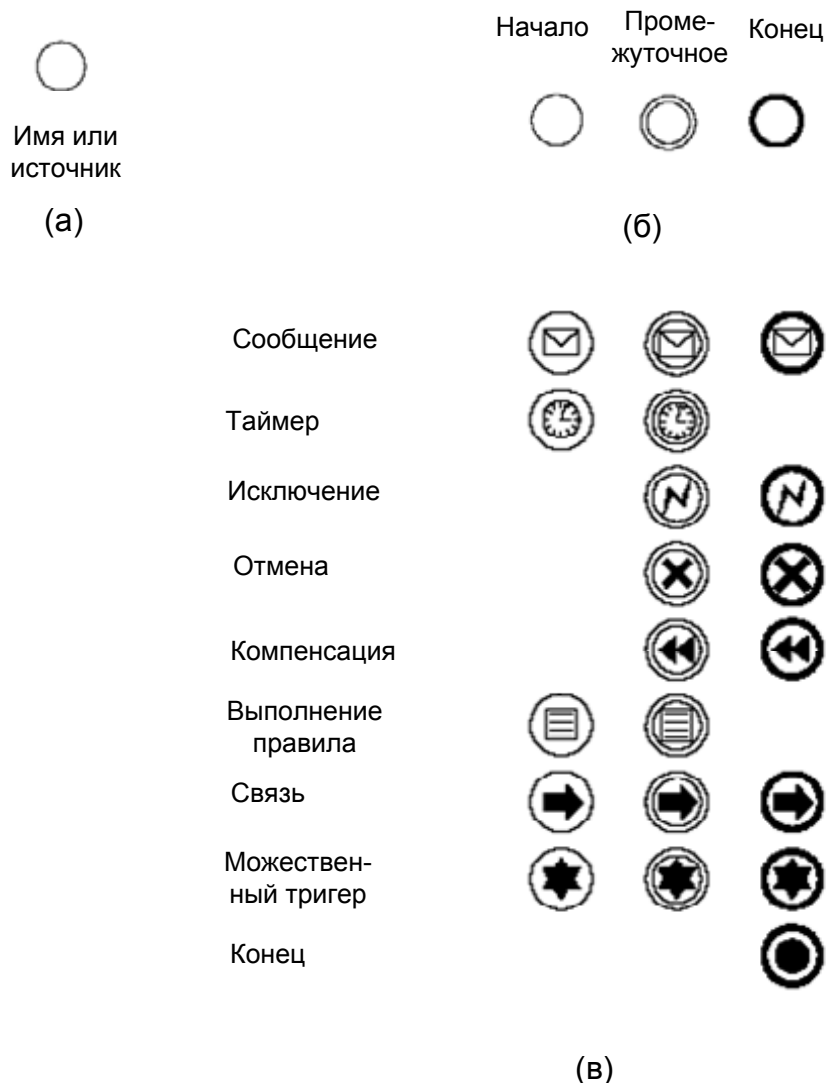


Рис. 9.7.

Эти виды событий по-разному изображаются на BPMN-спецификациях, как показано на рис. 9.7 (б). В контексте этих трех видов события могут различаться по типам – см. рис. 9.7 (в):

- *сообщение* (message) – прием/посылка сообщения;
- таймер (timer) – истечение определенного промежутка времени;
- *исключение* (error) – происшествие исключительного события, например, ошибки при обработке данных;
- *отмена* (cancel) – отмена действия или транзакции (то есть возврат объемлющего процесса или подпроцесса к состоянию, которое было до начала исполнения этого действия /транзакции);
- *компенсация* (compensation)– выполнение специальных отменяющих действий например, при отказе заказчика от услуги;
- *выполнение правила* (rule) – событие, которое обозначает, что в бизнес-процессе выполнилось какое-либо бизнес-правило, например, ставка акций компании поднялась выше определенной суммы и в результате этого нужно сделать что-то особое, определенное;
- *связь* (link) – способ переключаться между двумя процессами (как правило, подпроцессами одного общего) или как оператор goto в рамках одного процессора; в первом случае первый подпроцесс должен иметь конечное событие такого типа с пометкой в какой подпроцесс прыгать дальше. А тот, второй подпроцесс, должен либо стартовать с события, также помеченного как link, либо иметь такое же промежуточное событие. И в том и в другом случае целевые события link должны иметь идентификатор связывающий их с тем, исходным событием link;
- *множественный триггер* (multiple) – «ловит» (в качестве начального или промежуточного события) одно из списка, связанных с ним; в качестве заключительного события порождает весь список связанных с ним событий.
- *конец* (terminate) – имеет только тип конец, обозначает, что все действия процесса завершается, все его экземпляры (если их запущено более одного) также завершаются.

События могут «цепляться» к границе действия, а могут быть узлами, которые соединяются связями потока управления. Далее они могут обозначать ожидание события, а могут быть его источником (например, событие отправки сообщения). Существуют многочисленные правила, которые определяют детали того, где и при каких условиях может размещаться то или иное событие.

**Об исполняемой семантике бизнес-процессов.** Столь детально определенный язык описания бизнес-процессов был бы излишен и даже невозможен, если бы у моделей бизнес-процессов не существовало исполняемой семантики. Разберемся с тем, что это такое.

Очевидно, что модели, созданные с помощью BPMN алгоритмичны, то есть можно сконструировать некоторый вычислитель, который их будет исполнять. На практике таким вычислителем является, например ERP-система, в рамках которой определяется бизнес-процесс компании и тем самым система настраивается на его исполнение.

Кроме того, отдельным действиям могут соответствовать определенные, программные компоненты (в том числе и распределенные в сети). Тогда бизнес-процесс оказывается общим алгоритмом, связывающих их в единое целое и предоставляющий некоторым клиентам компьютеризированный сервис. Например, сервис по бронированию гостиницы через Интернет может включать в себя поиск отеля, удовлетворяющего определенным характеристикам, связь с этим отелем на предмет знакомства с его параметрами и его бронирование. Все эти три задачи выполняются на разных Интернет-узлах.

С бизнес-процессами крепко связаны Web-сервисы. Так, язык BPMN имеет исполняемые проекции в язык BEPL, а последний описывает бизнес-процессы как набор взаимодействующих Web-сервисов.

Web-сервисом, согласно [4] называется программная система, идентифицируемая строкой URI, чьи открытые интерфейсы и привязки определены и описаны посредством XML. Описание этой программной системы может быть найдено другими программными системами, которые могут взаимодействовать с ней согласно этому описанию посредством сообщений, основанных на XML, и передаваемых с помощью Интернет-протоколов. URI (Uniform Resource Indicator) состоит из унифицированного указателя информационного ресурса (URL – Uniform Resource Locator) и унифицированного имени ресурса (URN – Uniform Resource Name). URN – это имя, которое не ссылается на физический ресурс. Вокруг Web-сервисов существует большое количество стандартов [5], и в целом мировое сообщество здесь движется к созданию автоматизированных и интегрированных через Интернет бизнес-процессов, реализующих многочисленные B2B (Business to Business) связи. Однако, в настоящий момент существует большое количество параллельных стандартов, крупные производители, пользуясь этой парадигмой продвигают свои системы и платформы и т.д. То есть реализация этой идеи – пока дело будущего.

### **Основная литература**

1. М.Хаммер, Д.Чампли. Реинжиниринг корпораций. // Пер. с англ. – СПб. Изд-во СПб ун-та, 1999. – 328 с.
2. M.Havey. Essential Business Process Modeling. – O'REILLY, 2005. – 332 p.
3. Business Process Modeling Notation. Final Notation Specification dtc/06-02-01. OMG 2006. <http://www.omg.org/>

### **Дополнительная литература**

4. Web-services architecture requirements. – W3C Working Group, Note 11, February 2004. – <http://www.w3.org/TR/wsa-reqs/>
5. S.Dietzen. Standards for Service-oriented architecture. Weblogic Pr. May/June 2004.
6. BPEL4WS Specification: Business Process Execution Language for Web Services. – Version 1.1, 5 May 2003. – 137 p.