

Лекция 3. По дисциплине «Реинжиниринг в логистике»: МЕТОДОЛОГИЯ МОДЕЛИРОВАНИЯ БИЗНЕС-ПРОЦЕССОВ

Вопросы:

1. Требования к моделям бизнес-процессов .
2. Моделирование бизнеса на языке UML.
3. Объектно-ориентированный язык UML.
4. Прецедентная модель бизнес-процесса.
5. Объектная модель бизнес-процесса.

1. Требования к моделям бизнес-процессов

Моделирование – основной инструмент анализа и проектирования БП. Модель есть отображение (представление) объекта, системы или понятия в некоторой форме, отличной от формы их реального существования .

Существует огромное количество различного рода моделей. Классифицировать модели можно различными способами (рисунок 1).

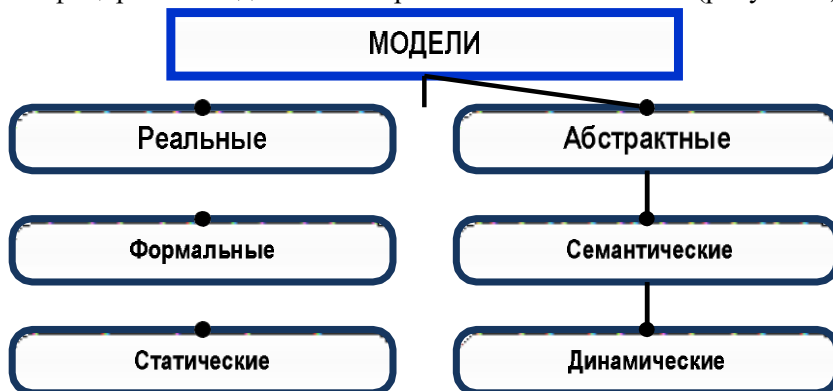


Рисунок 1. Классификация моделей

По способу воплощения модели можно разделить на реальные (материальные, физические) и абстрактные (идеальные). *Реальная* модель – модель, построенная из реальных объектов. Примерами являются макеты, тренажеры. Для моделирования бизнес-процессов

подобные модели практически не используются. *Абстрактная* модель есть отображение реального объекта в виде идеальных конструкций, выполненных средствами мышления. Выделим два основных класса абстрактных моделей: формальные (математические) и семантические (содержательные).

В *формальных* моделях отражаются математические закономерности, существующие между количественными параметрами. К этому классу моделей относятся, в частности, системы уравнений, статистические модели, модели исследования операций др. Математические модели универсальны в том смысле, что одна и та же модель может описывать весьма различные физические процессы или явления. Основное достоинство математических моделей заключается в том, что они позволяют с помощью формального аппарата вычислений находить решение при заданных условиях, но не все системы удастся адекватно описать некоторой единой математической моделью. При анализе и проектировании бизнес-процессов математические модели используются для расчета отдельных характеристик процессов, например, стоимостных, временных, пространственных.

В *семантических* моделях, в отличие от формальных, сохраняется семантика (смысл, содержание) объекта. Примерами семантических моделей могут служить дерево целей организации, модель организационной структуры, модель информационных коммуникаций компании и т. п. Как правило, семантические модели описывают некие объекты (сущности, системы, подсистемы, элементы), свойства объектов, их состояния и поведение, а также отношения между объектами. В качестве языковых средств семантического моделирования используются графы, диаграммы, таблицы, блок-схемы, естественный язык (язык общения между людьми). Противопоставление семантических моделей формальным вовсе не означает, что в них полностью отсутствует формализация. Как правило, можно формально описать форму, структуру семантической модели, но оперирование моделью, анализ осуществляется не на формальном уровне, а на самой модели – с учетом семантики, заложенной в ней.

Семантические модели незаменимы на ранних этапах проектирования сложных систем, когда формируется концепция системы. Интегрированная семантическая модель системы позволяет представить общую картину, составить обобщенное описание, в котором

подчеркнуты основные сущности, а детали скрыты. Главное в такой модели – краткость и понятность. Такая модель может служить основой для построения более детальных моделей, описывающих отдельные аспекты, подсистемы.

Таким образом, семантическая модель может служить каркасом для построения других моделей, в том числе и математических. Она служит для структуризации информации о моделируемом объекте.

Основой большинства семантических моделей являются базовые модели системного анализа – модель «черного ящика», модель состава и модель структуры системы. Модель «черного ящика» описывает только входы и выходы системы, но не ее внутреннее устройство. Модель состава описывает, из каких частей (подсистем, элементов) состоит система. Как правило, модель состава строится путем декомпозиции – последовательного разбиения системы на все более мелкие части. Модель структуры описывает все отношения (связи) между частями системы.

В зависимости от того, учитывается или нет в модели фактор времени, семантические модели разделяют на динамические и статические.

Статические модели не учитывают временной параметр и отражают постоянные, устойчивые состояния объектов (систем, процессов), их состав, структуру, устойчивые внутренние и внешние связи.

Динамические модели отображают поток событий, т. е. изменение во времени состояний объектов, последовательность выполнения взаимодействий объектов.

Любая компания, как известно, – сложная система. Для нее невозможно построить одну-единственную модель, охватывающую абсолютно все детали. В общем случае необходима не одна, а несколько согласованных моделей, отражающих различные аспекты.

В интегрированной модели бизнеса должны найти отражение:

1. Функция компании во внешнем мире: что она делает, для кого, с какой целью. Для этого используются модели типа «черного ящика», диаграммы взаимодействия с окружением (например, диаграмма вариантов использования системы окружением – Use Case Diagram). Такая модель описывает: окружающую среду компании, включающую клиентов, поставщиков, партнеров, субподрядчиков и

т. д.; основные бизнес-процессы, а также взаимодействие процессов с окружением.

Кроме того, для отражения системы целей компании и требований, предъявляемых к ней окружением, могут быть использованы модели типа дерева целей.

2. Описание бизнес-процессов, отдельных шагов процессов (функций, работ, операций). Для описания последовательности выполнения процессов разработаны различные виды моделей.

Диаграммы деятельности (Activity Diagrams) описывают последовательность действий (в том числе с условными переходами) в виде схем, напоминающих блок-схемы алгоритмов.

Диаграммы функциональной декомпозиции (например, модель IDEF0) позволяют описывать процесс на разных уровнях детальности: на верхнем уровне представлен процесс целиком, затем он декомпозируется и описывается взаимовлияние полученных подпроцессов друг на друга. Затем каждый из подпроцессов может быть подвергнут дальнейшему разбиению и для его частей может быть построена диаграмма взаимосвязей.

В дополнение к моделям функциональной декомпозиции используются модели *функционально-стоимостного анализа (ФСА-модели или ABC – Activity Based Costing)*, описывающие механизм формирования стоимости продукции на основе стоимости функций и ресурсов, задействованных в бизнес-процессах.

Для отражения логики взаимовлияния работ и событий используются *диаграммы потоков работ (Work Flow Diagrams)*. Они позволяют отражать в модели, например, следующие ситуации: «все предшествующие работы должны быть завершены одновременно», «одиночки нескольких следующих процессов запускаются одновременно».

Несмотря на то, что перечисленные виды моделей БП отражают последовательность выполнения отдельных шагов, время явным образом не присутствует. Чтобы отразить временные характеристики процессов, используют *модели календарного планирования* (управления проектом), в частности, график Ганта, сетевой график и т. д. Наиболее полную картину состояния процесса в любой момент времени позволяют получить компьютерные *имитационные модели*. Они копируют бизнес-процессы путем отображения «живой» картины процесса (при помощи анимации) в режиме сжатого времени.

3. Описание объектов, участвующих в выполнении БП или обрабатываемых, создаваемых бизнесом и отношений между объектами. Объектами являются исполнители, управленцы, оборудова-

ние, инструменты, продукция, сырье, материалы и т. д.

Для отражения свойств, характеристик объектов используются модели данных. Так, например, *модель «сущность – связь»* (**Entity – Relationship Diagram**) описывает объекты (сущности), их свойства (атрибуты) и отношения между объектами (связи).

Модель классов (**Class Diagram**) отражает информационную структуру классов объектов и отношения между ними.

Для моделирования поведения объектов используются *диаграммы состояний* (**Statechart Diagrams**). Они описывают последовательности состояний объекта, события и переходы между состояниями.

Взаимодействие объектов во время выполнения бизнес-процессов отражают с помощью статических и динамических диаграмм взаимодействия (**Interaction Diagrams**). К наиболее важным статическим структурам относится также организационная структура компании.

Определим теперь требования к методологиям моделирования бизнеса. Прежде всего методология должна позволять строить понятные и обзримые модели. Модель должна исключать ненужные детали, акцентируя внимание на наиболее существенных аспектах. Лучше использовать интегрированную методологию, объединяющую методы построения нескольких взаимосвязанных моделей, отражающих различные аспекты моделируемого бизнеса. Необходимо, чтобы язык описания модели был выразителен. С другой стороны, методология должна быть достаточно формализована, т. е. должна включать в себя некоторые довольно четкие процедуры или техники генерации компонент модели. Желательно, чтобы методология поддерживалась инструментальными компьютерными системами. Использование инструментальных средств поддержки существенно облегчает процесс разработки больших проектов, особенно при коллективной разработке, в которой участвуют различные группы разработчиков.

Нами рассмотрены две методологии моделирования бизнес-процессов: объектно-ориентированная методология (язык) **UML (Unified Modeling Language** – унифицированный язык моделирования) и методология функциональной декомпозиции **IDEF (Integrated computer-aided manufacturing DEFINITION)**. Обе методологии используют выразительный графический язык и позволяют строить

наглядные, легко воспринимаемые модели. При этом используется принцип многомодельности, т. е. методологии позволяют формировать совокупность взаимосвязанных моделей, отражающих различные представления системы. Так, **IDEF** – пять основных представлений (рисунок 2), **UML** включает восемь видов моделей-диаграмм (рисунок 3).

Немаловажным фактором является поддержка этих методологий компьютерными инструментальными средствами. На рынке **CASE**-средств представлен целый ряд инструментариев.

В частности, методология **IDEF** поддерживается такими популярными средствами, как **BPWin**, **Design/IDEF**.

Самым известным **CASE**-средством, ориентированным на язык **UML**, является **Rational Rose**. Во многом именно благодаря инструментальной поддержке рассматриваемые методологии получили очень широкое распространение.

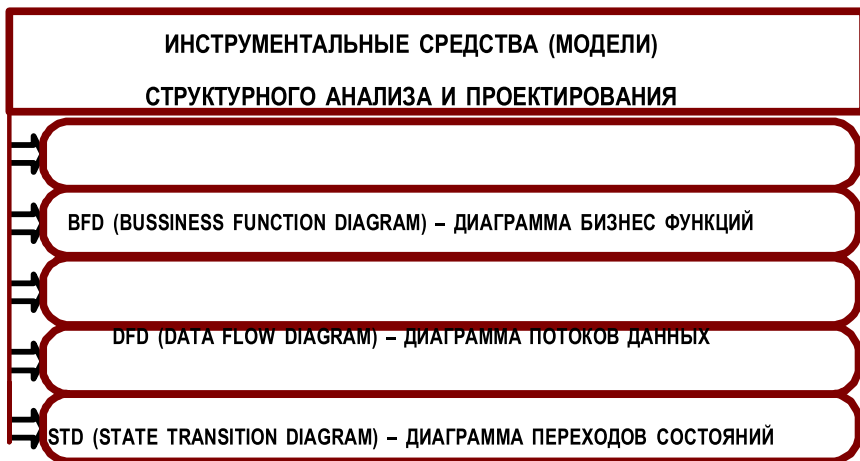


Рисунок 2. Инструментальные средства структурного проектирования

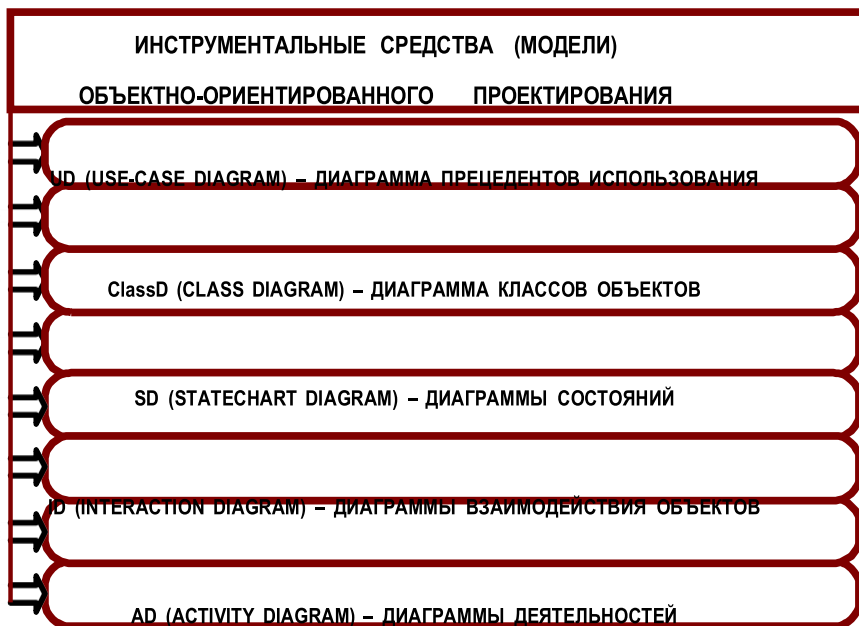


Рисунок 3. Инструментальные средства объектно-ориентированного проектирования (UML)

2. Моделирование бизнеса на языке UML

2.1. Объектно-ориентированный язык UML

Язык **UML** был разработан для создания моделей систем с целью их последующей реализации в виде объектно-ориентированных программ. Поскольку в основе **UML** лежит методология объектно-ориентированного программирования (ООП), необходимо пояснить, в чем ее суть.

Традиционно в программировании разделяли понятия данных и процедур. Примерами данных являются: целое число (**Integer**), действительное число (**Real**), строка (**String**) и т. д. Новый тип данных – *объект (object)* – объединяет в одну структуру совокупность данных (атрибутов) и процедур (методов). Ниже приведено описание типа объекта **TGraphObj** (графический объект) на языке Pascal:

Type

TGraphObj = **object** (имя класса – объектного типа)

X, Y: Integer; (атрибуты – координаты реперной точки)
Color: Word; (атрибут – цвет линии)
Constructor Init (aX,aY: Integer: aColor: Word); (процедура создания объекта)
Procedure Draw; virtual; (процедура рисовании) **Procedure MoveTo (dX,dY: Integer);** (процедура сдвига)**end;**

Может быть создано множество объектов данного типа. Таким образом, тип объекта определяет *класс* однородных объектов. Каждый конкретный объект (экземпляр объекта) содержит свои собственные значения атрибутов.

Самым замечательным свойством классов является возможность на основе одних классов создавать другие (наследовать) путем добавления новых атрибутов и методов, а также переопределения методов.

Ниже приведено описание класса-потомка **TRectan** (прямоугольник), наследуемого от класса-предка **TGraphObj**:

Type

TRectan = object (TGraphObj) (TRectan наследуется от TGraphObj)

dX, dY: Integer; (новые атрибуты – размер прямоугольника)

Constructor Init (aX,aY,adX,adY: Integer: aColor: Word);
(процедура создания объекта)

Procedure Draw; virtual; (переопределенная процедура рисования)
end;

В определении класса-потомка приводятся только новые атрибуты/методы и переопределенные методы, атрибуты и методы предка наследуются «автоматически». Классы никогда не переписываются, они только расширяются и переопределяются. Это произвело революцию в программировании, т. к. позволило создавать программу из готовых «кирпичиков» – библиотечных объектов. Для различных языков программирования были созданы библиотеки стандартных классов, содержащие классы стандартных визуальных компонент – окон, меню, списков выбора, командных кнопок, кнопок выбора и т. д. Библиотечные классы компонент, построенные на принципах прямого манипулирования, «умеют» в ответ на манипуляции пользователя мышью или ввод с клавиатуры совершать

стандартные действия – вызывать соответствующие методы. Появились средства «визуального» программирования или средства быстрой разработки приложений (**RAD – Rapid Application Development**), с помощью которых программист создает большую часть программы путем манипулирования мышью (передвигая на экране визуальные компоненты) и ввода свойств компонент – соответствующий программный код генерируется автоматически.

Внимание программистов переключилось с непосредственного написания программного кода на предшествующие этапы – анализ предметной области и проектирование программы. Появились методы объектно-ориентированного анализа и проектирования (**OOA/OOD – Object- Oriented Analysis/Design**).

Разные авторы создавали различные языки объектно-ориентированного моделирования, отличающиеся составом, видом диаграмм, используемыми обозначениями.

Наиболее известными к середине 1990-х годов стали: метод Booch'93 Гради Буча, метод OMT (**Object Modeling Technique**) Джеймса Румбаха и метод **OOSE (Object-Oriented Software Engineering)** Айвара Джекобсона.

Авторы этих методов решили объединить свои представления и создать унифицированный метод, что и привело к появлению языка **UML**.

Благодаря поддержке консорциума **OMG** и разработке **CASE**-средства **Rational Rose**, ориентированного на **UML**, этот язык стал фактически стандартом в области объектно-ориентированного моделирования.

В технологии **BPR**, пожалуй, впервые объектно-ориентированные методы (в том числе **UML**) стали применяться не только и не столько для создания информационных систем, сколько для анализа и перепроектирования БП. Вместо моделей процессов, реализуемых информационной системой, строятся модели бизнес-процессов, даже если они и не будут подвергнуты автоматизации, вместо объектов ИС (программных объектов) в моделях отражаются объекты БП (исполнители, продукция, услуги и т. д.), вместо окружения ИС (пользователей ИС) моделируется окружение бизнеса (поставщики, партнеры, клиенты).

Моделирование бизнеса с помощью **UML** предполагает последовательное построение двух видов моделей – прецедентной моде- ли

(П-модели) и объектной модели (О-модели).

2.2. Прецедентная модель бизнес-процесса

Прецедентная модель описывает бизнес-процессы компании. Если проводить аналогию с моделями системного анализа, то П-модель можно отнести к функциональным моделям, описывающим деятельность системы. Однако в отличие от системного анализа, где под функцией понимается любое изменение состояния системы во времени (любая деятельность), в BPR понятие «функция» трактуется значительно уже и относится лишь к видам деятельности, выполняемым отдельными функциональными подразделениями. Процесс включает в себя различные виды деятельности (функции), начиная с обработки входов и заканчивая созданием продукции, необходимой клиенту.

Например, процесс «обслуживание авиапассажира» включает в себя такие функции, выполняемые работниками различных подразделений, как регистрация пассажиров, прием и выдача багажа, техническое обслуживание самолета, проведение полета и т. д.

Построение прецедентной модели начинается с формирования так называемой внешней диаграммы или *диаграммы вариантов использования (Use Case Diagram)*, описывающей бизнес так, как он виден извне, т. е. как он воспринимается клиентами и другим окружением. Данная диаграмма отражает представление о том, *что* делает бизнес, а не *как* делает.

Основными элементами диаграммы являются: прецеденты (варианты использования) и акторы (субъекты). *Прецедентом* или вариантом использования в UML называется законченная совокупность действий моделируемой системы, начинающаяся при получении стимула извне и заканчивающаяся предоставлением некоторого сервиса актору – пользователю системы. При моделировании информационной системы прецедент соответствует отдельному сервису, предоставляемому ИС пользователю.

При моделировании бизнеса прецедент ставится в соответствие БП. Это, прежде всего, «внешние» бизнес-процессы, ориентированные на клиента, т. е. заканчивающиеся получением продукта (товара или услуги) – измеримой потребительской ценности для некоторого индивидуального клиента бизнес-системы. В виде прецедентов могут быть представлены и «внутренние» бизнес-процессы, клиентами которых выступают другие части бизнес-системы, не участвующие в

выполнении процесса.

Прецеденты представляют собой поток взаимосвязанных событий. При этом прецеденты могут иметь множество вариантов хода событий. Каждый конкретный прецедент (вариант) называется *экземпляр*. Экземпляр реализует конкретный поток событий в конкретных условиях для конкретного клиента. Похожие варианты хода событий группируются в *классы* прецедентов. Можно рассматривать класс как обобщенный прецедент.

Например, класс прецедентов «Продажа» описывает общий ход событий, выполняемых при продаже любого продукта любому клиенту. Конкретный экземпляр прецедента «Продажа» может отличаться в деталях, например, в зависимости от того, новый это клиент или нет, компетентный или несведущий и т. п.

Акторами, или субъектами, в модели бизнеса являются элементы окружения – клиенты, партнеры, поставщики. Акторы обозначают все то в окружении, что взаимодействует с бизнесом. Это может быть человек (например, клиент, покупатель), другая компания (например, организация-поставщик, субподрядчик), техническая система (например, компьютерная система другой компании).

Актор представляет собой абстракцию кого-либо или чего-либо, использующего бизнес. Он обозначает *роль*, которую что-то или кто-то может играть по отношению к бизнесу. Не следует путать реальных людей с акторами: реальная личность может играть несколько ролей в бизнес-системе. Например, конкретный человек может быть и клиентом, и поставщиком.

Так же, как и для прецедентов, для акторов различают понятия класса и экземпляра. Класс описывает общие характеристики некоторого типа акторов, а экземпляр – характеристики конкретного актора. На диаграмме вариантов использования прецеденты обозначаются эллипсом, а акторы – фигуркой «человечка» (рисунок 4).

Как правило, на диаграмме отображаются классы прецедентов и классы акторов. На диаграмме могут быть также отображены примечания в виде прямоугольников с «загнутым» уголком (рисунок 4).

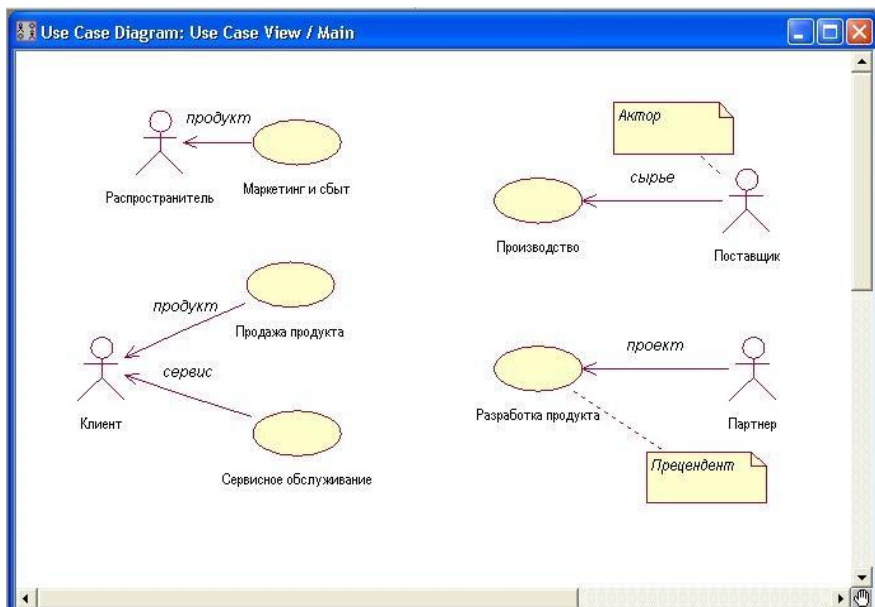


Рисунок 4. Диаграмма вариантов использования

Между прецедентами и акторами могут быть установлены отношения коммуникации (*communicate*), относящиеся к классу отношений *ассоциации*. На диаграмме они обозначаются сплошной линией со стрелками (рисунок 2). Отношения коммуникации отражают взаимосвязи прецедентов с окружением, заключающиеся в обмене веществом (сырьем, инструментами, готовой продукцией), энергией и информацией.

Таким образом отношения коммуникации моделируют материальные, энергетические и информационные потоки.

Следует заметить, что, как правило, между прецедентами не устанавливаются отношения коммуникации, т. к. в процессе своего выполнения они не «обращаются» друг к другу. Допустимо установление отношений зависимости между прецедентами. Между акторами отношения коммуникации также не указываются, т. к. с точки зрения бизнес-системы они не представляют интереса. Тот факт, что в виде «человечка» в UML обозначаются только внешние элементы, а не внутренние, объясняется тем, что UML создавался для моделирования информационных систем, в которых человек присутствует только как внешний элемент – пользователь АСУ.

Следующий шаг в построении П-модели – описание прецедентов последовательностью мелких шагов. Такое описание называется потоком событий. С точки зрения системного подхода осуществляется декомпозиция процесса-прецедента на подпроцессы-события.

Рассмотрим для примера описание прецедента «Продажа продукта»:

1. Продавец получает заявку клиента.
2. Если в заявке указан готовый продукт, то Продавец проверяет наличие требуемого продукта на складе. Если продукта нет в наличии, прецедент заканчивается. Если продукт есть на складе, то прецедент продолжается с шага б.
3. Если в заявке указывается заказной продукт, то Продавец формирует заказ и передает его Изготовителю продукта.
4. Изготовитель изготавливает продукт в соответствии с требованиями клиента и сообщает о готовности Продавцу.
5. Изготовитель отправляет продукт на Склад.
6. Продавец сообщает Клиенту о готовности продукта и принимает от Клиента оплату.
7. Продавец сообщает Отправителю количество продукта и адрес клиента и заказывает транспорт.
8. Отправитель получает продукт со склада и доставляет его клиенту.

Каждый шаг (событие) прецедента представляет собой некоторое действие, переводящее прецедент в новое состояние. В свою очередь новое состояние прецедента является стимулом для выполнения следующего шага (события). Таким образом, прецедент рассматривается как *машина состояний-событий*. Поток событий прецедента может быть представлен в виде *диаграммы деятельности (Activity diagram)*, имеющей некоторое сходство с блок-схемой или структурной схемой алгоритма.

Графически диаграмма деятельности представляется в форме графа, вершинами которого являются действия (шаги процесса), а дугами – переходы от одного действия к другому.

На диаграмме можно также отразить ветвление, т. е. возможность перехода к различным действиям в зависимости от некоторых условий.

На рисунке 5 приведена диаграмма деятельности для

прецедента «Продажа продукта». Начальное состояние, соответствующее началу процесса, обозначается в виде закрашенного кружка; конечное состояние, соответствующее завершению процесса, в виде закрашенного кружка, помещенного в окружность. Действие изображается фигурой, напоминающей прямоугольник с закругленными сторонами, внутри которого записывается выражение действия. Переход изображается сплошной линией со стрелкой. Если после выполнения некоторого действия процесс должен разделиться на альтернативные ветви в зависимости от некоторого условия, то ставится знак ветвления в виде ромба, внутри которого нет никакого текста.



Рисунок 5. Диаграмма деятельности прецедента «Продажа продукта»

В него может входить только одна стрелка. Выходящих стрелок может быть две или более. данный переход.

Для каждой из них указывается соответствующее условие, при котором выполняется

Прецедент может содержать различные альтернативные потоки событий, в том числе достаточно редкие и исключительные потоки, выполняемые при определенных условиях. Поэтому описание прецедента может быть очень сложным и запутанным, содержать множество условных переходов и проверок. Чтобы упростить описание прецедента, необходимо его *структурировать*. Рассмотрим два способа структурирования.

Первый способ структурирования сложных прецедентов заключается в использовании *отношения включения (include)*. Если из общего описания прецедента с альтернативными потоками событий можно выделить фрагмент, представляющий собой относительно законченную последовательность событий, то данный фрагмент рассматривается как отдельный прецедент и между ним и исходным прецедентом устанавливается отношение включения.

Поток событий включенного прецедента «встраивается» в поток событий базового прецедента, т. е. когда экземпляр базового прецедента в процессе своего выполнения достигает точки включения, выполняется последовательность шагов включенного прецедента, после чего продолжается выполнение исходного прецедента.

На рисунке 6 приведен пример структурирования прецедента «Продажа продукта», из которого был извлечен фрагмент, содержащий последовательность действий, выполняемую только в том случае, если клиент в заявке указывает заказной продукт.

Этот фрагмент выделен в отдельный прецедент «Исполнение заказа». На диаграмме вариантов использования отношение включения между прецедентами «Продажа продукта» и «Исполнение заказа» обозначено пунктирной линией со стрелкой и ключевым словом «include». Поток событий прецедента «Исполнение заказа» вызывается из прецедента «Продажа продукта». В диаграмме деятельности вызов вложенной последовательности действий обозначается с помощью действия, имеющего специальную пиктограмму.

Частным случаем структурирования с помощью выделения фрагментов является использование *отношения расширения (ex-*

tend). Данное отношение устанавливается между базовым прецедентом и прецедентом, содержащим некоторое дополнительное поведение, выполняемое при определенных условиях.

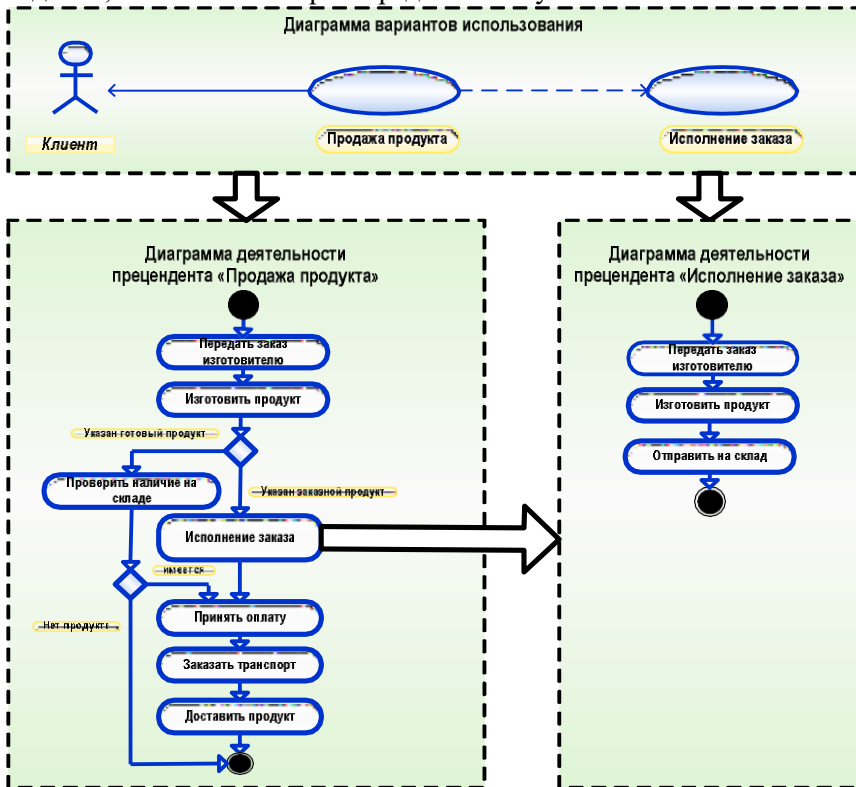


Рисунок 6. Структурирование прецедента «Продажа продукта» посредством отношения включения

Отношения включения и расширения используются для того, чтобы показать условные части прецедента, смоделировать фрагменты потока событий, выполняемые в особых случаях, а также смоделировать добавление в поток событий нескольких различных фрагментов в любой комбинации.

Второй способ структурирования сложных прецедентов основан на использовании *отношения обобщения (generalization)*. Если несколько прецедентов имеют похожее поведение, то следует выделить общее поведение в отдельный прецедент (родительский) и установить между ним и исходными отношение обобщения. В этом

случае общее поведение описывается только один раз. Описания конкретных прецедентов (потомков) содержат только дополнительные шаги (или модифицированные шаги), которых нет в обобщенном описании. Можно сказать, что прецеденты-потомки наследуются от некоего обобщенного прецедента-родителя, являются его подклассами. Например, чтобы структурировать введенное выше описание прецедента «Продажа продукта», вводится абстрактный класс «Общий вид продаж» и два наследуемых класса – «Продажа готового продукта» и «Продажа заказного продукта». На рисунке 7 приведена модель взаимосвязей между этими тремя прецедентами. Как видно из рисунка, клиент взаимодействует не с обобщенным прецедентом, а с одним из конкретных прецедентов, который в ходе своего выполнения может вызывать поток событий предка. Поскольку в этом случае сначала начинает выполняться поток событий одного из прецедентов-потомков, то необходимо, чтобы заранее было известно, какой из прецедентов будет выполняться. Например, заказы на готовый продукт и на заказной принимаются разными продавцами в разных отделах, т. е.

«Получить заказ на готовый продукт» и «Получить заказ на заказной продукт» – это разные действия, включенные в соответствующие прецеденты (рисунке 7).

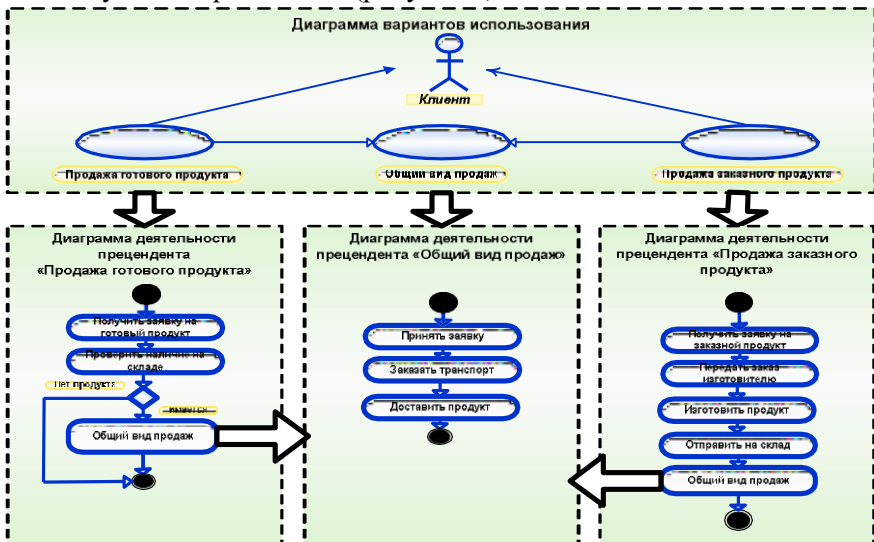


Рисунок 7. Структурирование прецедентов «Продажа готового продукта» и «Продажа заказного продукта» посредством отношения обобщения

2.2.1. Объектная модель бизнес-процесса

П-модель иллюстрирует функции бизнеса, его окружение и бизнес-процессы. Однако для более полного понимания бизнеса такого описания не достаточно. Необходима модель, показывающая, *как, за счет чего* реализуются прецеденты.

Объектная модель раскрывает внутреннее устройство бизнеса, а именно: какие виды ресурсов используются для реализации прецедентов, и каким образом они взаимодействуют.

Описание О-модели базируется на понятии «объект». *Объекты* представляют *участников* процессов (исполнителей, менеджеров) и различного рода *сущности* (продукцию, предметы, задачи и т. д.). Участники процессов называются активными объектами, сущности – пассивными.

Различают *классы* объектов, описывающие общие характеристики некоторого типа объектов, и *экземпляры*, описывающие характеристики конкретного объекта.

О-модель, представленную в терминах классов объектов, называют идеальной моделью. Такая модель не учитывает некоторых деталей реализации модели на практике. О-модель, описанную в терминах экземпляров объектов, называют реальной.

Выделяют следующие *категории (роли)* объектов:

1. Интерфейсные (Boundary)

– активные объекты, осуществляющие взаимодействие с окружением, т. е. с акторами. Примерами интерфейсных объектов являются Продавец, Регистратор, Секретарь. В роли интерфейсного объекта может выступать не только человек, но и, например, ИС.

2. Управляющие (Control)

– активные объекты, участвующие в выполнении процессов, но не имеющие контакта с окружением. Типичные примеры управляющих объектов – Разработчик продукции, Менеджер проекта. Следует отметить, что в роли управляющих объектов выступают не только менеджеры (управленцы), но и исполнители (рабочие, служащие, клерки), а также подразделения компании, ИС.

3. Объекты-сущности (Entity)

– пассивные объекты, которые обрабатываются бизнесом. Как правило, объекты-сущности не являются человеческими или техническими ресурсами. Типичные примеры сущностей в компании – Продукция, Заказ, Извещение.

Один и тот же объект может участвовать не только в одном прецеденте, он может принимать участие во многих событиях в бизнесе. Кроме того, как и в случае с актерами, один реальный человек или подразделение может выполнять роли нескольких объектов. Например, продавец продукции, кроме роли интерфейсного объекта, выполняющего контакт с клиентом, может играть роль менеджера.

Взаимодействие объектов во время выполнения БП отображается с помощью статических и динамических диаграмм взаимодействия.

На рисунке 8 представлена диаграмма кооперации для прецедента «Продажа заказного продукта», описанного в предыдущем параграфе. Прежде всего на диаграмме в виде прямоугольников изображаются объекты. Внутри прямоугольника записывается имя объекта, после которого через слэш может быть указана роль (буквой «и» обозначается роль интерфейсного объекта, буквой «у» – роль управляющего объекта, буквой «с» – роль объекта-сущности), следом через двоеточие может быть указано имя класса. Вся запись подчеркивается, что является признаком объекта, отличающим его от класса.

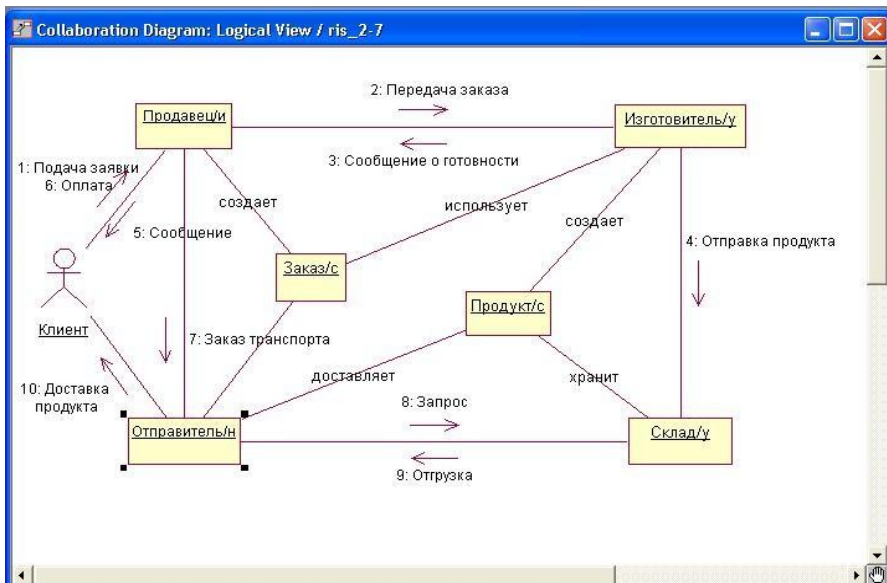


Рисунок 8. Диаграмма кооперации прецедента «Продажа заказного продукта»

Между объектами устанавливаются отношения. Отношение отображается на диаграмме в виде линии, соединяющей объекты (или объект с актором), рядом с которой может быть указана метка – текст, специфицирующий отношение.

Отношения, представленные на диаграмме кооперации, относятся к двум типам: сообщения и связи.

Отношение *сообщения* (message) аналогично отношению коммуникации диаграммы вариантов использования и отражает передачу информации (или некоторый материальный поток) между объектами. При этом прием сообщения инициирует выполнение определенных действий тем объектом, которому сообщение передано.

На диаграмме сообщение отображается линией, над которой помещен небольшой отрезок линии со стрелкой и меткой. Метка может содержать номер. Номера сообщений позволяют отразить последовательность передачи сообщений. Следует отметить, что отношения сообщений устанавливаются только между активными объектами или между активным объектом и актором.

Отношение *связи* (link), относящееся к классу отношений ассоциации, отражает некоторую произвольную связь (ассоциацию) между двумя объектами. При моделировании бизнес-процессов в виде отношения связи чаще всего представляют отношение *использования*, показывающее, что один объект некоторым образом использует другой. Например, объект Продавец создает объект Заказ, объект Изготовитель использует Заказ для получения описания продукта, Отправитель продукта использует Заказ для получения информации о том, куда доставлять продукт.

Диаграмма кооперации отражает статический взгляд на реализацию прецедента. Чтобы отразить динамику реализации прецедента участвующими в нем объектами, используется динамическая модель взаимодействия объектов. Примером такой модели является *диаграмма последовательности* (Sequence Diagram). При ее построении описание взаимодействия объектов как бы «накладывается» на поток событий прецедента, т. е. создается еще одно описание потока событий в терминах участвующих объектов.

Методика построения подобных диаграмм использовалась долгое время в сфере телекоммуникаций в основном для описания взаимодействия между блоками аппаратуры. И. Якобсон предложил использовать эти диаграммы для объектно-ориентированных моделей

бизнес-процессов.

На рисунке 9 представлен пример диаграммы последовательности для прецедента «Продажа заказного продукта».

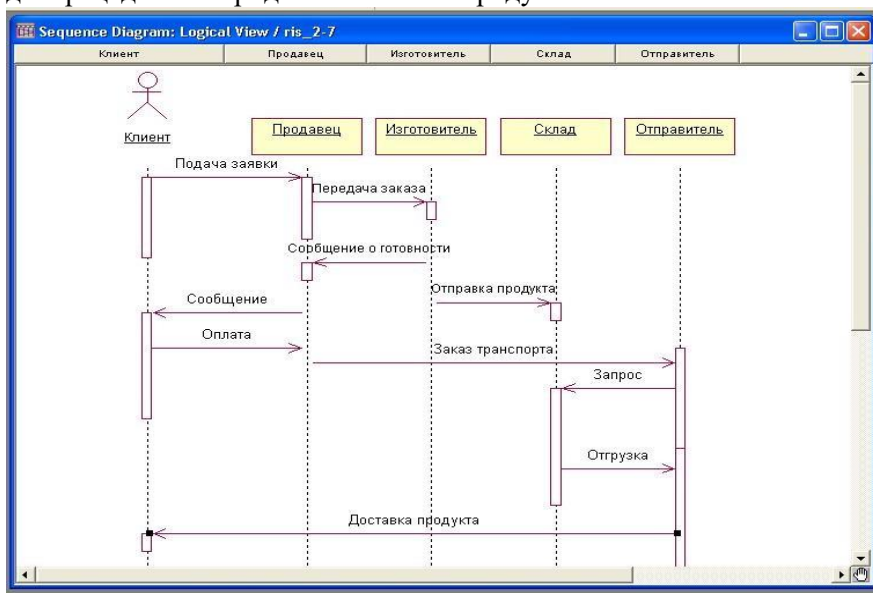


Рисунок 9. Диаграмма последовательности прецедента «Продажа заказного продукта»

На диаграмме последовательности отражаются только отношения сообщений. Каждый объект (актор), участвующий в реализации прецедента, изображается в верхней части диаграммы в виде прямоугольника («человечка»), от которого вниз проведена линия («линия жизни»). Сообщение изображается отрезком горизонтальной линии со стрелкой, проведенным от линии жизни объекта (актора), посылающего сообщение, до линии жизни объекта (актора), получающего сообщение.

При этом сообщения упорядочены по времени: первое сообщение изображается вверху диаграммы, следующее – ниже, следующее – еще ниже и т. д.

Таким образом, ось времени в диаграмме идет сверху вниз, однако она не связана с метрикой и служит только для идентификации порядка взаимодействия, т. е. расстояние между взаимодействиями (сообщениями) на диаграмме не имеет ничего общего с интервалами

времени между событиями.

Диаграмма кооперации и диаграмма последовательности одного и того же прецедента должны быть согласованы, т. к. это разные представления одних и тех же взаимодействий. Однако на диаграмме последовательности не отображаются объекты-сущности и отношения связи.

Все вышеописанные аспекты О-модели – статическая и динамическая модели взаимодействия объектов – были направлены на описание участия объектов в прецедентах. Причем один и тот же объект может участвовать в различных вариантах прецедента и в различных прецедентах. Для того чтобы иметь представление обо всех ролях и обязанностях объекта, нужно составить описание объекта.

Описание объекта состоит из двух частей: описание свойств и описание поведения. Для составления *описания свойств* объекта прежде всего необходимо выделить все его характеристики, называемые атрибутами. Например, объект Заказ может иметь атрибуты, указывающие название заказываемого товара, его цвет, количество, имя клиента, заказавшего товар и т. д. Как правило, состав атрибутов одинаков для всего класса однотипных объектов. Различные объекты одного класса отличаются лишь набором конкретных значений атрибутов. Например, для разных экземпляров класса Заказ будут указаны разные наименования товара, цвет, количество и т. д.

Для описания состава атрибутов классов, а также для отображения взаимосвязей между классами используется *диаграмма классов* (Class diagram). Пример подобной диаграммы приведен на рисунке 8. Каждый класс на диаграмме представлен в виде прямоугольника, который может быть разделен на три секции. В первой секции записывается имя класса (имя класса в отличие от имени объекта не подчеркивается), во второй – список атрибутов, в третьей – операции (методы) класса. Атрибут должен иметь имя, после которого через двоеточие может быть указан тип (Integer, Real, String, Class и т. д.).

Если один из классов представляет собой некоторую сущность, включающую в себя в качестве составных частей другие сущности, то между классами, сопоставленными исходной сущности и ее частям, устанавливается отношение *включения* (include). Например, вся информация о заказе может быть разбита на две части – информация о клиенте и информация о заказываемом продукте (наименование, цвет, количество). В этом случае кроме класса *Заказ* вводятся классы *Клиент* и *Продукт* и между каждым из новых классов и исходным

устанавливается отношение включения (рисунке 10).

Если несколько классов имеют одинаковые атрибуты, то можно ввести обобщенный класс, содержащий эти атрибуты, и установить между исходными классами и новым отношением *обобщения* (generalization). В этом случае общие атрибуты можно описать только в обобщенном классе (классе-предке), а для классов-потомков описываются только те атрибуты, которых нет у предка.

Другими словами, потомок наследует все характеристики (атрибуты) предка, однако может иметь дополнительные характеристики. Например, класс *Заказ*, как и любой другой документ, имеет такие атрибуты, как номер документа, дата создания, создатель документа и т. д. Поэтому можно ввести класс *Документ*, описать в нем атрибуты, общие для всех документов, и установить между классами *Заказ* и *Документ* отношения обобщения. Класс *Заказ* при этом наследует атрибуты класса *Документ* и имеет свои собственные атрибуты (рисунок 10).

Описание поведения объекта заключается в выявлении всех его *обязательств*, т. е. всех взаимодействий объекта с другими объектами и акторами в ходе выполнения всех прецедентов.

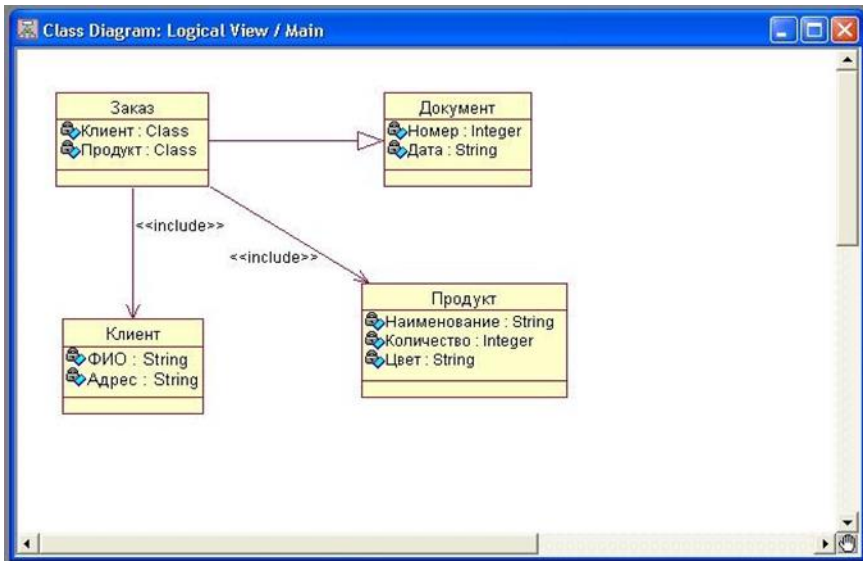


Рисунок 10. Пример диаграммы классов

Из всех диаграмм, где фигурирует описываемый объект, вычленяются все обязательства объекта (взаимодействия) и объединяются (рисунок 11). В результате получается описание всех обязательств объекта во всех прецедентах.

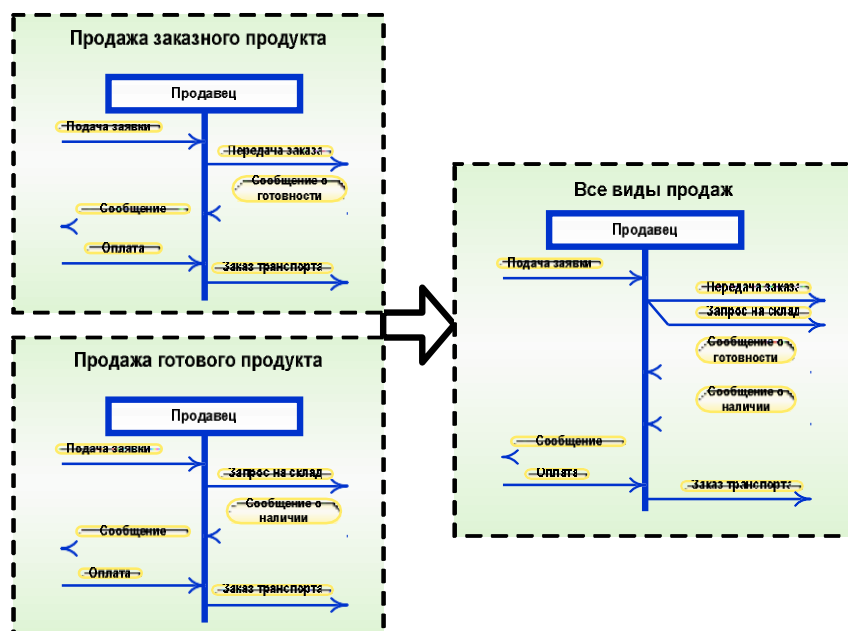


Рисунок 11. Обязательства объекта, определяемые из диаграмм взаимодействия

Все требования к объекту, состоящие из описания состояний объекта и описания поведения, собираются в документ, называемый спецификацией объекта.

Контрольные вопросы:

1. Чем отличаются реальные и абстрактные модели, формальные и семантические, статические и динамические?
2. Что должно найти отражение в интегрированной модели бизнеса?
3. Каковы основные требования к методологии моделирования бизнеса?
4. Что понимается под объектом в объектно-ориентированном программировании?
5. В чем разница между экземпляром объекта и классом?
6. Что описывает прецедентная модель бизнес-процесса?
7. Что такое прецедент? Что такое актер? Что обозначают эти понятия при моделировании бизнеса?
8. Приведите примеры экземпляров и классов прецедентов.
9. Приведите примеры экземпляров и классов акторов.
10. Что отображается на диаграмме вариантов использования?
11. Между какими элементами диаграммы вариантов использования могут быть установлены отношения коммуникации?
12. Что такое поток событий прецедента?
13. Каковы основные элементы диаграммы деятельности?
14. Охарактеризуйте способ структурирования прецедентов, основанный на использовании отношения включения.
15. Охарактеризуйте способ структурирования прецедентов, основанный на использовании отношения обобщения.
16. Приведите примеры интерфейсных, управляющих объектов и объектов-сущностей.
17. В чем разница между статической и динамической моделями прецедента?