

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

**Тема 3. Описание поведения системы, поведенческие  
диаграммы**

## 1. Диаграммы состояний (statechart diagram)

Главное предназначение этой диаграммы – описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла. Чаще всего диаграммы состояний используются для описания поведения отдельных экземпляров классов (объектов), но они также могут быть применены для спецификации функциональности других компонентов моделей, таких как варианты использования, актеры, подсистемы, операции и методы.

Диаграмма состояний по существу является графом специального вида, который представляет некоторый автомат. Вершинами этого графа являются состояния и некоторые другие типы элементов автомата (псевдосостояния), которые изображаются соответствующими графическими символами. Дуги графа служат для обозначения переходов из состояния в состояние. Для понимания семантики конкретной диаграммы состояний необходимо представлять не только особенности поведения моделируемой сущности, но и знать общие сведения по теории автоматов.

### 1.1. Автоматы

*Автомат* (state machine) в языке UML представляет собой некоторый формализм для моделирования поведения элементов модели и системы в целом. Автомат описывает поведение отдельного объекта в форме последовательности состояний. Графически эта информация может быть представлена в виде изображенной ниже диаграммы состояний компьютера (рис. 1).



Рис. 1 Простейший пример диаграммы состояний для технического устройства типа компьютер

Основными понятиями, входящими в формализм автомата, являются состояние и переход.

В языке UML под *состоянием* понимается абстрактный метакласс, используемый для моделирования отдельной ситуации, в течение которой имеет место выполнение некоторого условия.

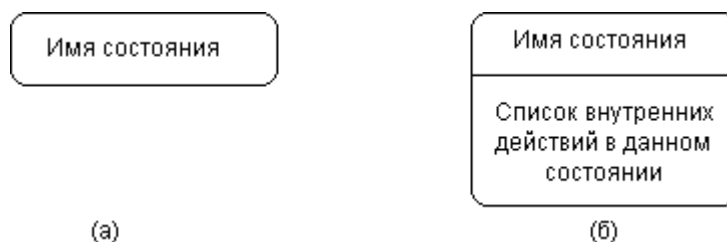


Рис. 2 Графическое изображение состояний на диаграмме состояний

Состояние на диаграмме изображается прямоугольником со скругленными вершинами (рис. 2). Этот прямоугольник, в свою очередь, может быть разделен на две секции горизонтальной линией. Если указана лишь одна секция, то в ней записывается только имя состояния (рис. 2, а). В противном случае в первой из них записывается имя состояния, а во второй – список некоторых внутренних действий или переходов в данном состоянии (рис. 2, б).

*Начальное состояние* представляет собой частный случай состояния, которое не содержит никаких внутренних действий. В этом состоянии находится объект по умолчанию в начальный момент времени. Графически начальное состояние в языке UML обозначается в виде закрашенного кружка (рис. 3, а), из которого может только выходить стрелка, соответствующая переходу.



Рис. 3 Графическое изображение начального и конечного состояний на диаграмме состояний

*Конечное (финальное) состояние* представляет собой частный случай состояния, которое также не содержит никаких внутренних действий. В этом состоянии будет находиться объект по умолчанию после завершения работы автомата в конечный момент времени. Графически конечное состояние в языке UML обозначается в виде закрашенного кружка, помещенного в окружность (рис. 55, б), в которую может только входить стрелка, соответствующая переходу.

*Простой переход* (simple transition) представляет собой отношение между двумя последовательными состояниями, которое указывает на факт смены одного состояния другим. На диаграмме состояний переход изображается сплошной линией со стрелкой, которая направлена в целевое состояние (рис. 4).

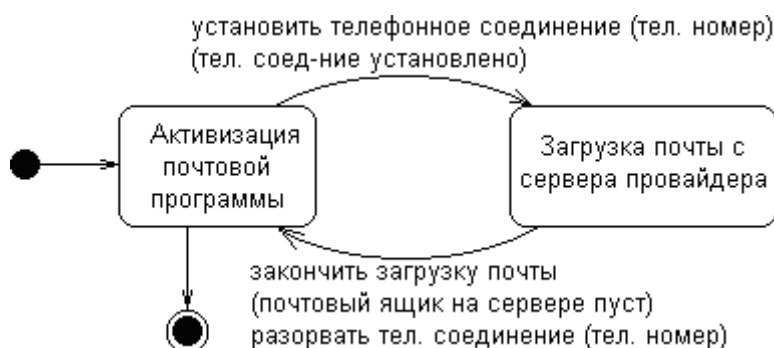


Рис. 4 Диаграмма состояний для моделирования почтовой программы-клиента

## 1.2 Пример диаграммы состояний

Рассмотрим пример диаграммы состояний для моделирования поведения банкомата (рис. 5).

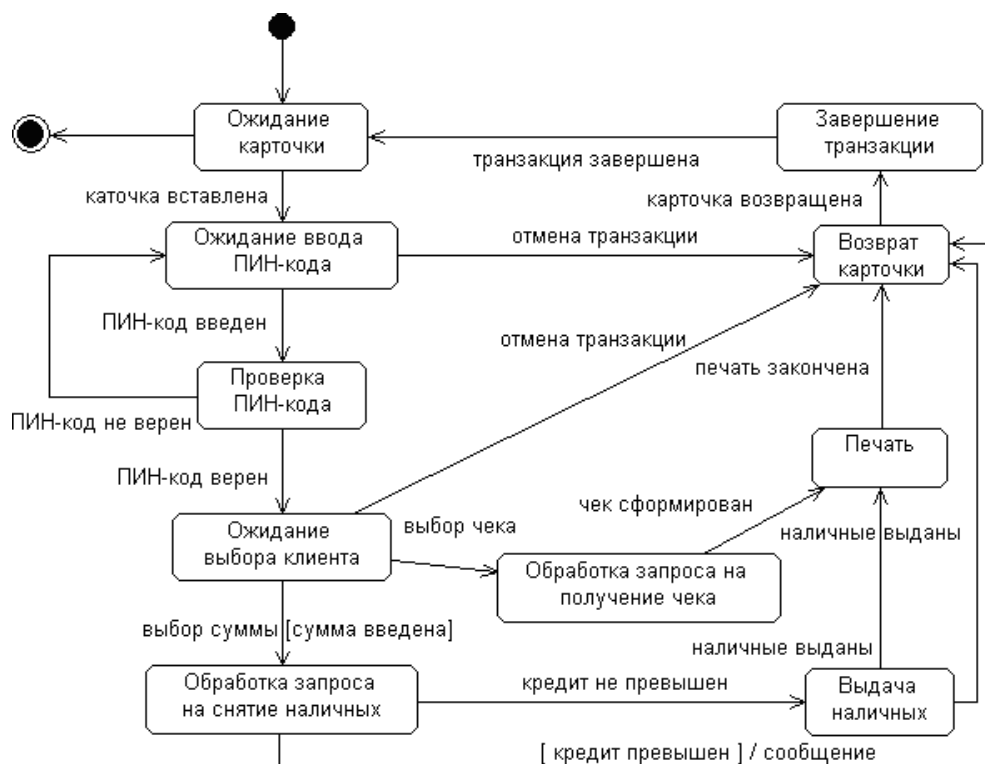


Рис. 5 Диаграмма состояний для моделирования поведения банкомата

Следует заметить, что в разрабатываемой модели диаграмма состояний является единственной и описывает поведение системы управления банкоматом в целом.

## 2. Диаграммы последовательности (sequence diagram)

Для моделирования взаимодействия объектов в языке UML используются соответствующие диаграммы взаимодействия. При этом учитываются два аспекта: во-первых, взаимодействия объектов можно рассматривать во времени, и тогда для представления временных особенностей передачи и приема сообщений между объектами используется *диаграмма последовательности*. Во-вторых, можно рассматривать структурные особенности взаимодействия объектов. Для представления структурных особенностей передачи и приема сообщений между объектами используется диаграмма кооперации.

### 2.1. Объекты диаграммы последовательности

Диаграммы последовательности отражают поток событий, происходящих в рамках варианта использования. На этих диаграммах изображаются только те объекты, которые непосредственно участвуют во взаимодействии т.к. ключевым моментом является именно динамика взаимодействия объектов во времени и не используются возможные статические ассоциации с другими объектами. При этом диаграмма последовательности имеет два измерения (рис. 6). Одно – слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии. Второе измерение – вертикальная временная ось, направленная сверху вниз. При этом

взаимодействия объектов реализуются посредством сообщений, которые посылаются одними объектами другим. Сообщения изображаются в виде горизонтальных стрелок с именем сообщения и также образуют порядок по времени своего возникновения. Другими словами, сообщения, расположенные на диаграмме последовательности выше, инициируются раньше тех, которые расположены ниже.

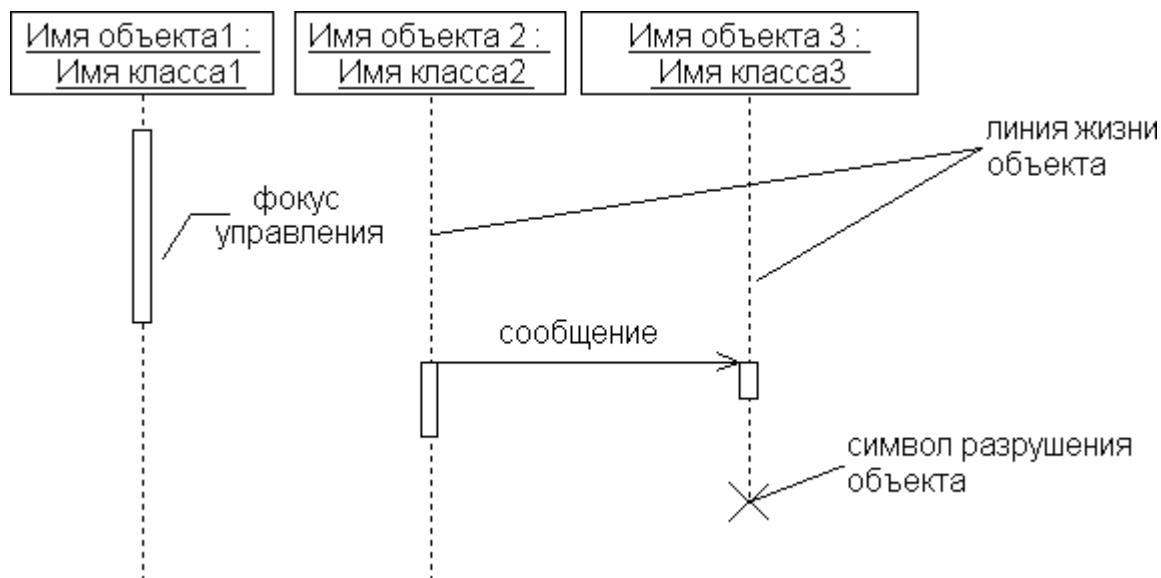


Рис. 6 Графические примитивы диаграммы последовательности

*Линия жизни* объекта (object lifeline) изображается пунктирной вертикальной линией, ассоциированной с единственным объектом на диаграмме последовательности. Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях. Если объект существует в системе постоянно, то его линия жизни должна начинаться в верхней части диаграммы и заканчиваться в нижней части (объекты 1 и 2 на рис. 31). Отдельные объекты в системе могут создаваться по мере необходимости, существенно экономя ресурсы системы и повышая ее производительность (объект 6 на рис. 7).

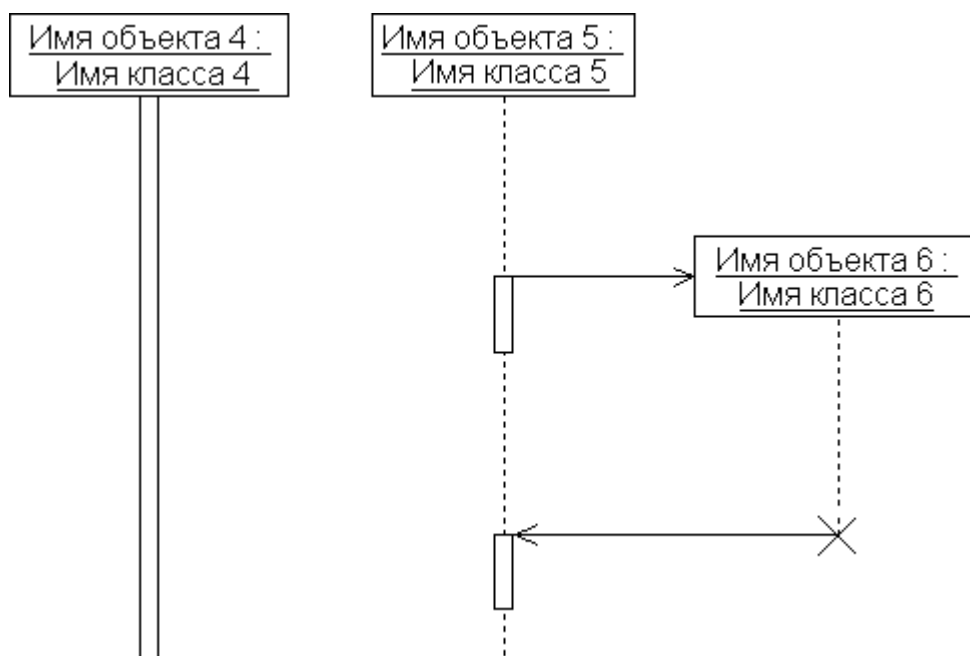


Рис. 7 Варианты линий жизни и фокусов управления объектов

Комментарии или примечания уже рассматривались ранее при изучении других видов диаграмм. Они могут включаться и в диаграммы последовательности, ассоциируясь с отдельными объектами или сообщениями.

Как уже отмечалось выше, взаимодействия объектов реализуются с помощью сообщений. У каждого сообщения должно быть имя, соответствующее его цели. Существует несколько видов сообщений: простое, синхронное, с отказом становиться в очередь и др. (рис. 8).



Рис. 8 Примеры сообщений

*Простое сообщение* используется по умолчанию. Означает, что все сообщения выполняются в одном потоке управления (рис. 8, 1).

*Синхронное* (synchronous) применяется, когда клиент посылает сообщение и ждет ответа пользователя (рис. 8, 2).

*Сообщение с отказом становиться в очередь (balking):* клиент посылает сообщение серверу и, если сервер не может немедленно принять сообщение, оно отменяется (рис. 8, 3).

*Сообщение с лимитированным временем ожидания (timeout):* клиент посылает сообщение серверу, а затем ждет указанное время; если в течение этого времени сервер не принимает сообщение, оно отменяется (рис. 8, 4).

*Асинхронное сообщение (asynchronous):* клиент посылает сообщение серверу и продолжает свою работу, не ожидая подтверждения о получении (рис. 8, 5).

### **3. Диаграммы кооперации (collaboration diagram)**

Подобно диаграммам последовательности, *диаграммы кооперации* отображают поток событий в конкретном сценарии варианта использования. Главная особенность диаграммы кооперации заключается в возможности графически представить не только последовательность взаимодействия, но и все структурные отношения между объектами, участвующими в этом взаимодействии.

Прежде всего, на диаграмме кооперации в виде прямоугольников изображаются участвующие во взаимодействии объекты, содержащие имя объекта, его класс и, возможно, значения атрибутов. Далее, как и на диаграмме классов, указываются ассоциации между объектами в виде различных соединительных линий. При этом можно явно указать имена ассоциации и ролей, которые играют объекты в данной ассоциации. Дополнительно могут быть изображены динамические связи – потоки сообщений. Они представляются также в виде соединительных линий между объектами, над которыми располагается стрелка с указанием направления, имени сообщения и порядкового номера в общей последовательности инициализации сообщений.

#### **3.1. Пример диаграммы кооперации**

На рис. 9 приведена кооперативная диаграмма, описывающая, как клиент снимает со счёта 20\$.



Рис. 9 Диаграмма кооперации для снятия клиентом 20\$

Из Кооперативной диаграммы легче понять поток событий и отношения между объектами, однако труднее уяснить последовательность событий, поэтому для сценария создают диаграммы обоих типов.

#### 4. Диаграммы деятельности (activity diagram)

При моделировании поведения проектируемой или анализируемой системы возникает необходимость не только представить процесс изменения ее состояний, но и детализировать особенности алгоритмической и логической реализации выполняемых системой операций.

Для моделирования процесса выполнения операций в языке UML используются так называемые *диаграммы деятельности*. Применяемая в них графическая нотация во многом похожа на нотацию диаграммы состояний, поскольку на диаграммах деятельности также присутствуют обозначения состояний и переходов. Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия, а дугами – переходы от одного состояния действия к другому.

В контексте языка UML деятельность (activity) представляет собой некоторую совокупность отдельных вычислений, выполняемых автоматом. При этом отдельные элементарные вычисления могут приводить к некоторому результату или действию (action). На диаграмме деятельности отображается логика или последовательность перехода от одной деятельности к другой, при этом внимание фиксируется на результате деятельности. Сам же результат может привести к изменению состояния системы или возвращению некоторого значения.



#### 4.1. Основные элементы диаграммы деятельности

Состояние действия (action state) является специальным случаем состояния с некоторым входным действием и по крайней мере одним выходящим из состояния переходом. Графически состояние действия изображается фигурой, напоминающей прямоугольник, боковые стороны которого заменены выпуклыми дугами (рис. 10). Внутри этой фигуры записывается выражение действия (action-expression), которое должно быть уникальным в пределах одной диаграммы деятельности.



Рис. 10 Графическое изображение состояния действия

Действие может быть записано на естественном языке, некотором псевдокоде или языке программирования. Никаких дополнительных или неявных ограничений при записи действий не накладывается.

При построении диаграммы деятельности используются только те переходы, которые переводят деятельность в последующее состояние сразу, как только закончится действие в предыдущем состоянии (нетриггерные). На диаграмме такой переход изображается сплошной линией со стрелкой.

Ветвление на диаграмме деятельности обозначается небольшим ромбом, внутри которого нет никакого текста (рис. 10).

В языке UML для распараллеливания вычислений используется специальный символ для разделения (рис. 11, а) и слияния (рис. 11, б) параллельных вычислений или потоков управления.

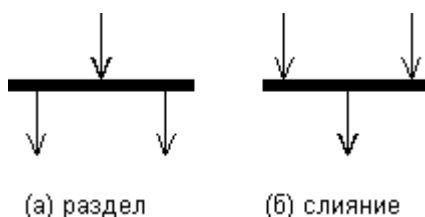


Рис. 11 Разделения и слияния параллельных потоков управления

Диаграммы деятельности могут быть использованы не только для спецификации алгоритмов вычислений или потоков управления в программных системах. Не менее важная область их применения связана с моделированием бизнес-процессов. Для моделирования этих особенностей в языке UML используется специальная конструкция, получившее название *дорожки* (swimlanes).

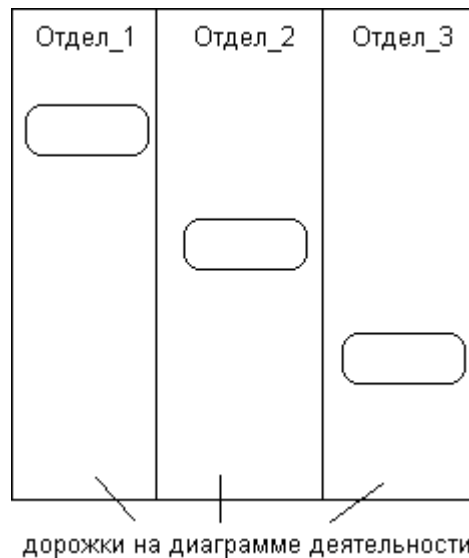


Рис. 12 Вариант диаграммы деятельности с дорожками

В общем случае действия на диаграмме деятельности выполняются над теми или иными объектами. Эти объекты либо инициируют выполнение действий, либо определяют некоторый результат этих действий. При этом действия специфицируют вызовы, которые передаются от одного объекта графа деятельности к другому. Поскольку в таком ракурсе объекты играют определенную роль в понимании процесса деятельности, иногда возникает необходимость явно указать их на диаграмме деятельности.

Соответствующая зависимость определяет состояние конкретного объекта после выполнения предшествующего действия.

#### 4.2. Пример диаграммы деятельности

В качестве примера рассмотрим фрагмент диаграммы деятельности торговой компании, обслуживающей клиентов по телефону. Подразделениями компании являются отдел приема и оформления заказов, отдел продаж и склад.

Этим подразделениям будут соответствовать три дорожки на диаграмме деятельности, каждая из которых специфицирует зону ответственности подразделения (рис. 13).

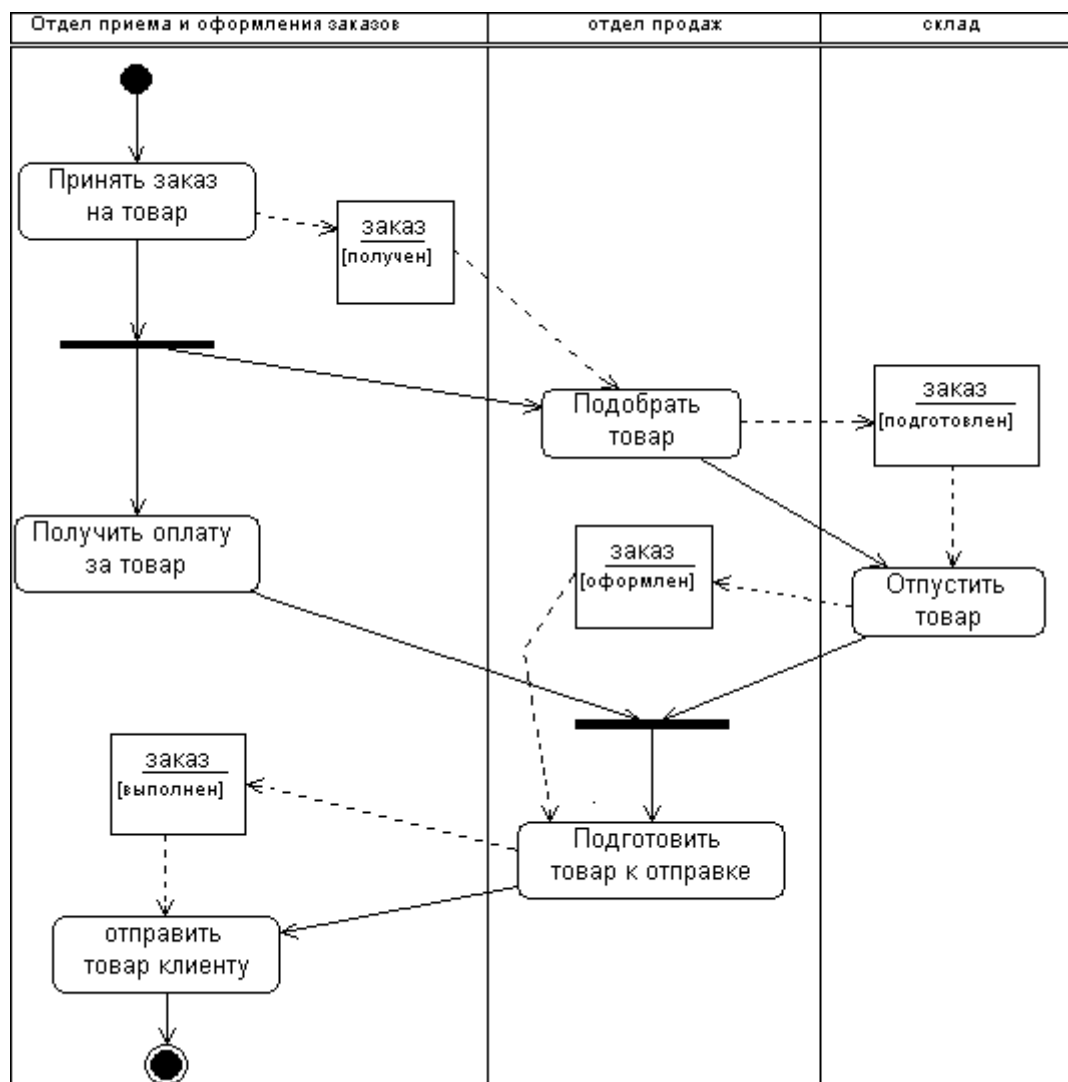


Рис. 13 Диаграмма деятельности торговой компании с объектом-заказом

В данном случае диаграмма деятельности включает в себе не только информацию о последовательности выполнения рабочих действий, но и о том, какое из подразделений торговой компании должно выполнять то или иное действие. Кроме того центральным объектом процесса продажи является заказ или вернее состояние его выполнения. Вначале до звонка от клиента заказ как объект отсутствует и возникает лишь после такого звонка. Однако этот заказ еще не заполнен до конца, поскольку требуется еще подобрать конкретный товар в отделе продаж. После его подготовки он передается на склад, где вместе с отпуском товара заказ окончательно дооформляется. Наконец, после получения подтверждения об оплате товара эта информация заносится в заказ, и он считается выполненным и закрытым.