

Приложение 5

Обзор основных диаграмм UML

Диаграмма вариантов использования

Диаграмма вариантов использования (диаграмма прецедентов или use-case диаграмма) описывает функциональное назначение системы. На диаграмме вариантов использования проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью вариантов использования.

Актер представляет собой любую внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей. Стандартным графическим обозначением актера является фигурка человечка, под которой записывается имя актера. Актеры взаимодействуют с системой посредством обмена сообщениями с вариантами использования.

Сообщение представляет собой запрос актером определенного сервиса системы и получение этого сервиса. Это взаимодействие может быть выражено посредством ассоциаций между отдельными актерами и вариантами использования. Кроме этого, с актерами могут быть связаны интерфейсы, которые определяют, каким образом другие элементы модели взаимодействуют с этими актерами.

Интерфейсы определяют совокупность операций, которые обеспечивают необходимый набор сервисов для актеров. Интерфейс изображается в виде маленького круга, рядом с которым записывается его имя. В качестве имени может быть существительное или строка текста. Если имя записывается на английском языке, то оно должно начинаться с заглавной буквы I. Графический символ отдельного интерфейса соединяется на диаграмме сплошной линией или пунктирной линией со стрелкой с тем вариантом использования, который его поддерживает. Сплошная линия указывает, что связанный с интерфейсом вариант использования должен реализовывать все необходимые для него сервисы. Пунктирная линия со стрелкой означает, что вариант использования предназначен для спецификации только того сервиса, который необходим для реализации данного интерфейса.

Вариант использования определяет законченный аспект или фрагмент поведения некоторой сущности без раскрытия её внутренней структуры. В качестве такой сущности может выступать система или любой элемент модели, который обладает собственным поведением. Каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемая сущность по запросу актера, то есть определяет способ применения этой сущности. Вариант использования обозначается эллипсом, внутри которого содержится его краткое название или имя в форме глагола с пояснительными словами.

Для включения в модель произвольной текстовой информации используются *примечания*. На диаграмме примечания обозначаются прямоугольником с загнутым верхним правым углом. Внутри прямоугольника содержится текст примечания. Если в примечании указывается ключевое слово «constraint», то оно является ограничением, налагаемым на соответствующий элемент модели.

Между элементами диаграммы вариантов использования могут существовать различные отношения, которые описывают взаимодействие экземпляров актеров и вариантов использования:

- *ассоциации*; отношение ассоциации специфицирует семантические особенности взаимодействия актеров и вариантов использования в графической модели системы, то есть, это отношение устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования. Графически отношение ассоциации обозначается сплошной линией между актером и вариантом использования. Эта линия может иметь условные обозначения, такие как имя и кратность.
- *расширения*; отношение расширения отмечает тот факт, что один из вариантов использования может присоединять к своему поведению некоторое дополнительное поведение, определенное для другого варианта использования. Графически отношение расширения обозначается пунктирной линией со стрелкой (вариант отношения зависимости), направленной от того варианта использования, который является расширением для исходного варианта использования. Данная линия со стрелкой помечается ключевым словом «extend».
- *обобщения*; отношение обобщения служит для указания того факта, что некоторый вариант использования А может быть обобщен до варианта использования В. Графически Отношение

обобщения обозначается сплошной линией со стрелкой в форме незакрашенного треугольника, которая указывает на родительский вариант использования.

- **включения**; отношение включения между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования. Графически Отношение включения обозначается пунктирной линией со стрелкой, которая помечается ключевым словом «include» (включает).

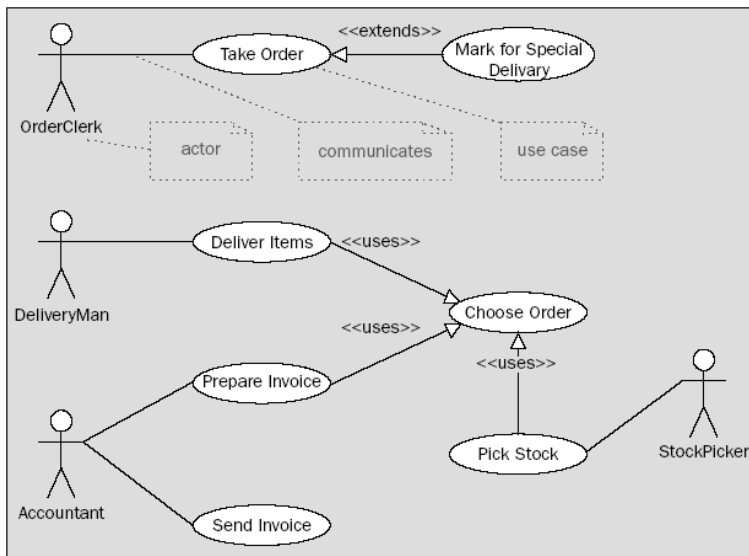


Диаграмма последовательности

Диаграмма последовательности показывает взаимодействия объектов, упорядоченные по времени их проявления. Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники), вертикальные линии (lifeline), отображающие течение времени при деятельности объекта, и стрелки, показывающие выполнение действий объектами. Обычно объекты на диаграмме располагаются слева направо.

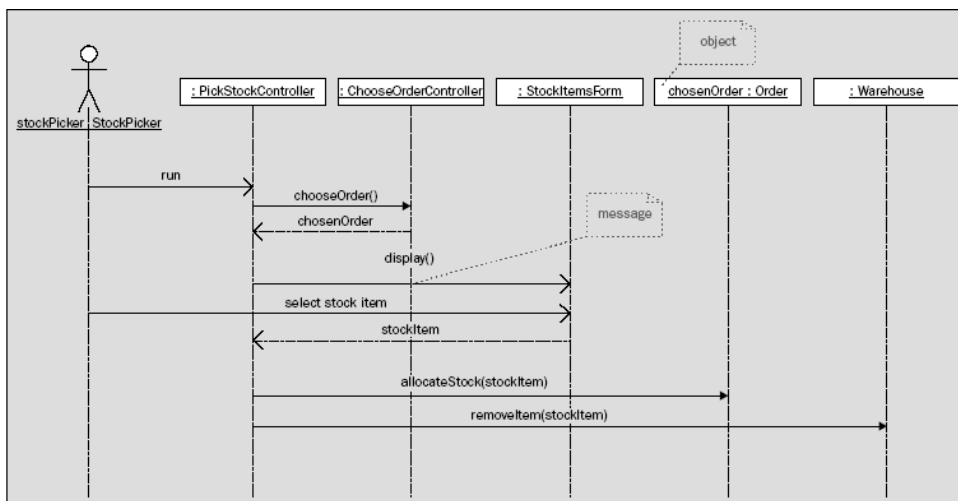


Диаграмма состояний

Диаграмма состояний является графом специального вида, который представляет некоторый автомат. Вершинами графа являются возможные состояния автомата, изображаемые соответствующими графическими символами, а дуги обозначают его переходы из состояния в состояние. Диаграммы состояний могут быть вложены друг в друга для более детального представления отдельных элементов модели. Состояние на диаграмме изображается прямоугольником со скругленными вершинами. Прямоугольник может быть разделен на две секции горизонтальной линией. Если указана лишь одна секция, то в ней записывается только имя состояния. При наличии двух секций, в первой из них записывается имя состояния, а во второй список некоторых внутренних действий или переходов в данном состоянии. Начальное состояние обозначается в виде

закрашенного кружка, из которого может только выходить стрелка, соответствующая переходу. Конечное состояние обозначается в виде закрашенного кружка, помещенного в окружность, которую может только входить стрелка, соответствующая переходу.

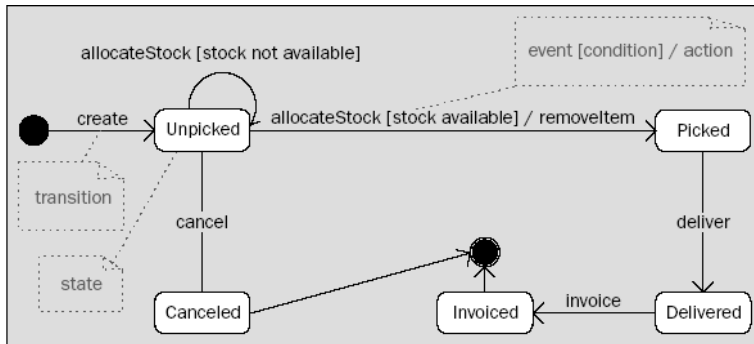


Диаграмма кооперации

Понятие кооперации служит для обозначения множества взаимодействующих с определенной целью объектов в общем контексте моделируемой системы. Цель самой кооперации состоит в том, чтобы специфицировать особенности реализации отдельных наиболее значимых операций в системе. Кооперация определяет структуру поведения системы в терминах взаимодействия участников этой кооперации. Кооперация может быть представлена на двух уровнях:

- на уровне спецификации - показывает роли классификаторов и роли ассоциаций в рассматриваемом взаимодействии;
- на уровне примеров - указывает экземпляры и связи, образующие отдельные роли в кооперации.

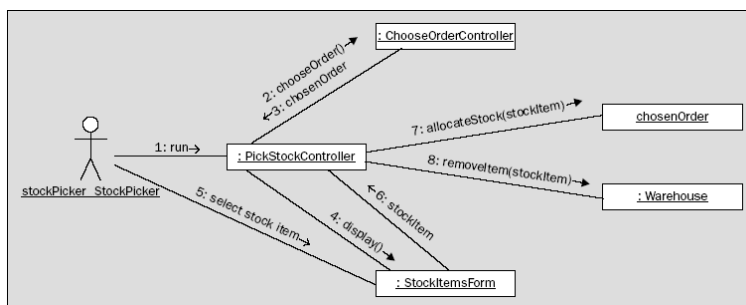


Диаграмма деятельности

Диаграмма деятельности показывает разбиение некоторой деятельности на её составные части. Аналогом диаграмм деятельности являются схемы алгоритмов. Диаграмму деятельности принято располагать таким образом, чтобы действия следовали сверху вниз. В этом случае начальное состояние будет изображаться в верхней части диаграммы, а конечное - в нижней. Графически состояние действия изображается прямоугольником с закругленными углами. Внутри этого изображения записывается выражение действия, которое должно быть уникальным в пределах одной диаграммы деятельности. Разделение и слияние параллельных вычислений или потоков управления изображается отрезком горизонтальной линии, толщина которой несколько шире основных сплошных линий диаграммы деятельности.

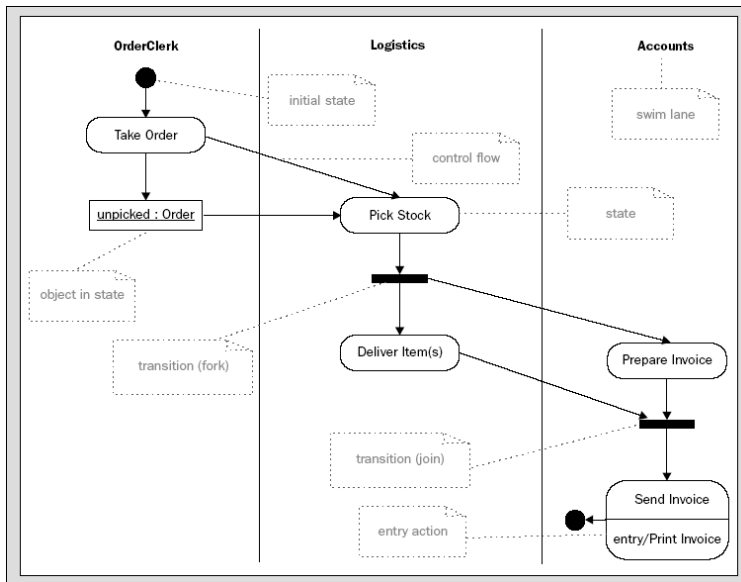


Диаграмма статической структуры или диаграмма классов

Центральное место в объектно-ориентированном программировании занимает разработка логической модели системы в виде диаграммы классов. Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывать их внутреннюю структуру и типы отношений. Графически класс изображается в виде прямоугольника, который дополнительно может быть разделен горизонтальными линиями на разделы или секции. В этих разделах могут указываться имя класса, атрибуты (переменные) и операции (методы).

«+» обозначает атрибут с областью видимости типа общедоступный (public).

«#» обозначает атрибут с областью видимости типа защищенный (protected).

«-» обозначает атрибут с областью видимости типа закрытый (private).

Кроме внутреннего устройства или структуры классов на соответствующей диаграмме указываются отношения между классами. При этом совокупность типов таких отношений фиксирована в языке UML и предопределена семантикой этих типов отношений. Базовыми отношениями в языке UML являются:

- *отношение зависимости* (пунктирная линия между соответствующими элементами со стрелкой, направленной от класса-клиента зависимости к независимому классу или классу-источнику);
- *отношение ассоциации* (сплошная линия с дополнительными специальными символами, которые характеризуют отдельные свойства конкретной ассоциации);
- *отношение обобщения* (сплошная линия с треугольной стрелкой на одном из концов; стрелка указывает на общий класс (класс-предок или суперкласс), а ее отсутствие - на специальный класс (класс-потомок или подкласс))

Диаграмма классов может содержать изображения абстрактных классов, интерфейсов и шаблонов или параметризованных классов.

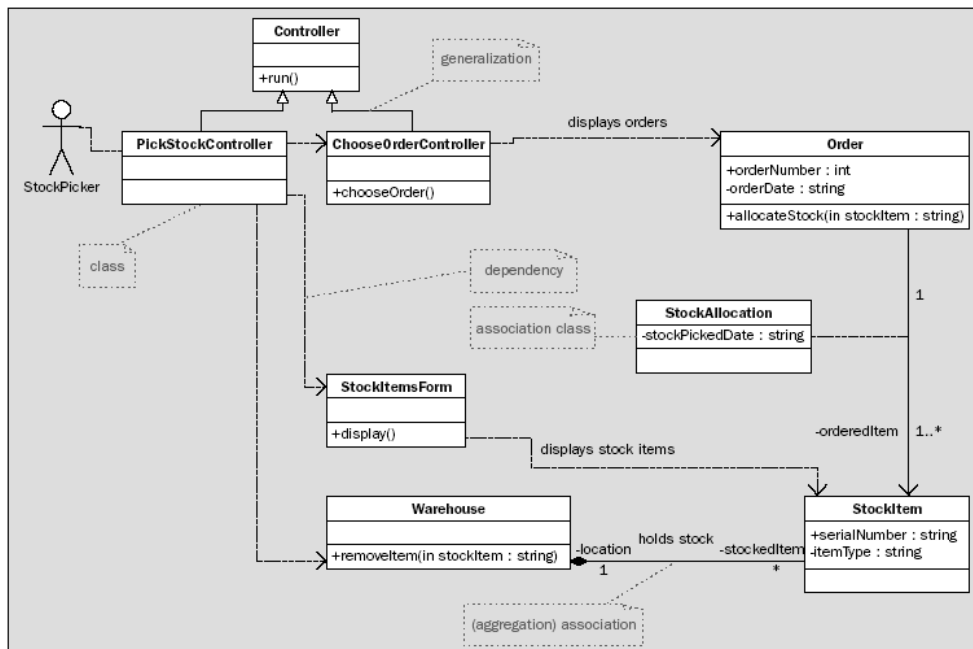


Диаграмма компонентов

Приложение будет доставлено или развернуто, как правило, не в виде отдельных классов или даже пакета классов, а скорее как один или несколько развертываемых компонентов - исполняемых файлов и библиотек - в которых классы или пакеты были собраны. Для физического представления моделей систем используются диаграммы реализации, которые включают в себя диаграмму компонентов и диаграмму развертывания.

Диаграмма компонентов описывает особенности физического представления системы. Она позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный и исполняемый код. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

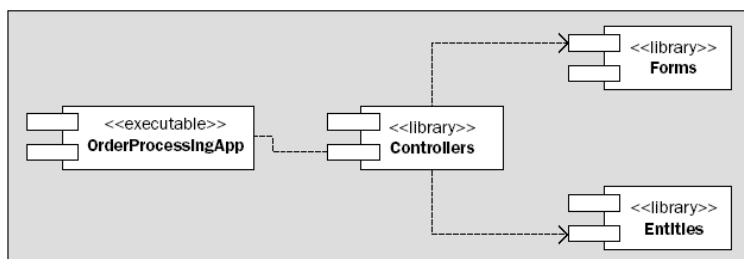


Диаграмма развертывания

Диаграмму развертывания рассмотрим на примере Web-приложения. Чтобы описать веб-сайт диаграмма развертывания должна показывать, какие аппаратные компоненты ("узлы") существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты ("артефакты") работают на каждом узле (например, web-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, REST, RMI).

Узлы представляются как прямоугольные параллелепипеды с артефактами, расположенными в них, изображенными в виде прямоугольников. Узлы могут иметь подузлы, которые представляются как вложенные прямоугольные параллелепипеды. Один узел диаграммы развертывания может концептуально представлять множество физических узлов, таких как кластер серверов баз данных. Существует два типа узлов:

Узел устройства - физические вычислительные ресурсы со своей памятью и сервисами для выполнения программного обеспечения, такие как обычные ПК, мобильные телефоны.

Узел среды выполнения - программный вычислительный ресурс, который работает внутри внешнего узла и который предоставляет собой сервис, выполняющий другие исполняемые программные элементы.

