

Лабораторная работа № 1.

Диаграммы прецедентов (вариантов использования)

Цель работы: построить диаграмму прецедентов (use case diagram)

Теоретические сведения

Диаграмма вариантов использования (use case diagram)

Визуальное моделирование в UML можно представить как некоторый процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной системы к логической, а затем и к физической модели соответствующей программной системы. Для достижения этих целей вначале строится модель в форме так называемой *диаграммы вариантов использования (use case diagram)*, которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Разработка диаграммы вариантов использования преследует цели:

- Определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы.
- Сформулировать общие требования к функциональному поведению проектируемой системы.
- Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей.
- Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

Примечание

Рассматривая диаграмму вариантов использования в качестве модели системы, можно ассоциировать ее с моделью черного ящика". Действительно, подробная детализация данной диаграммы на начальном этапе проектирования скорее имеет отрицательный характер, поскольку предопределяет I способы реализации поведения системы. А согласно рекомендациям RUP, именно эти аспекты должны быть скрыты от разработчика на диаграмме вариантов использования

В самом общем случае, диаграмма вариантов использования представляет собой граф специального вида, который является графической нотацией для представления

конкретных вариантов использования, актеров, возможно некоторых интерфейсов, и отношений между этими элементами. При этом отдельные компоненты диаграммы могут быть заключены в прямоугольник, который обозначает проектируемую систему в целом. Следует отметить, что отношениями данного графа могут быть только некоторые фиксированные типы взаимосвязей между актерами и вариантами использования, которые в совокупности описывают сервисы или функциональные требования к моделируемой системе.

Рекомендации по разработке диаграмм вариантов использования

Главное назначение диаграммы вариантов использования заключается в формализации функциональных требований к системе с помощью понятий соответствующего пакета и возможности согласования полученной модели с заказчиком на ранней стадии проектирования. Любой из вариантов использования может быть подвергнут дальнейшей декомпозиции на множество подвариантов использования отдельных элементов, которые образуют исходную сущность. Рекомендуемое общее количество актеров в модели — не более 20, а вариантов использования — не более 50. В противном случае модель теряет свою наглядность и, возможно, заменяет собой одну из некоторых других диаграмм.

Варианты использования могут быть специфицированы в виде текста, а в последующем — с помощью операций и методов вместе с атрибутами, в виде графа деятельности, посредством автомата или любого другого механизма описания поведения, включающего предусловия и постусловия. Взаимодействие между вариантами использования и актерами может уточняться на диаграмме кооперации, когда описываются взаимосвязи между сущностью, содержащей эти варианты использования, и окружением или внешней средой этой сущности

В случае, когда для представления иерархической структуры проектируемой системы используются подсистемы, система может быть определена в виде вариантов использования на всех уровнях. Отдельные подсистемы или классы могут выступать в роли таких вариантов использования. При этом вариант, соответствующий некоторому из этих элементов, в последующем может уточняться множеством более мелких вариантов использования, каждый из которых будет определять сервис элемента модели, содержащийся в сервисе исходной системы. Вариант использования в целом может рассматриваться как суперсервис для уточняющих его подвариантов, которые, в свою очередь, могут рассматриваться как подсервисы исходного варианта использования.

Функциональность, определенная для более общего варианта использования, полностью наследуется всеми вариантами нижних уровней. Однако следует заметить, что структура элемента-контейнера не может быть представлена вариантами использования, поскольку они могут представлять только функциональность отдельных элементов модели. Подчиненные варианты использования кооперируются для совместного выполнения суперсервиса варианта использования верхнего уровня. Эта кооперация также может быть представлена на диаграмме кооперации в виде совместных действий отдельных элементов модели.

Отдельные варианты использования нижнего уровня могут участвовать в нескольких кооперациях, т. е. играть определенную роль при выполнении сервисов нескольких вариантов верхнего уровня. Для отдельных таких коопераций могут быть определены соответствующие роли актеров, взаимодействующих с конкретными вариантами использования нижнего уровня. Эти роли будут играть актеры нижнего уровня модели системы. Хотя некоторые из таких актеров могут быть актерами верхнего уровня, это не противоречит принятым в языке UML семантическим правилам построения диаграмм вариантов использования. Более того, интерфейсы вариантов использования верхнего уровня

могут полностью совпадать по своей структуре с соответствующими интерфейсами вариантов нижнего уровня.

Окружение вариантов использования нижнего уровня является самостоятельным элементом модели, который в свою очередь содержит другие элементы модели, определенные для этих вариантов использования. Таким образом, с точки зрения общего представления верхнего уровня взаимодействие между вариантами использования нижнего уровня определяет результат выполнения сервиса варианта верхнего уровня. Отсюда следует, что в языке UML вариант использования является элементом-контейнером.

Варианты использования классов соответствуют операциям этого класса, поскольку сервис класса является по существу выполнением операций данного класса. Некоторые варианты использования могут соответствовать применению только одной операции, в то время как другие — конечного множества операций, определенных в виде последовательности операций. В то же время одна операция может быть необходима для выполнения нескольких сервисов класса и поэтому будет появляться в нескольких вариантах использования этого класса.

Реализация варианта использования зависит от типа элемента модели, в котором он определен. Например, поскольку варианты использования класса определяются посредством операций этого класса, они реализуются соответствующими методами. С другой стороны, варианты использования подсистемы реализуются элементами, из которых состоит данная подсистема. Поскольку подсистема не имеет своего собственного поведения, все предлагаемые подсистемой сервисы должны представлять собой композицию сервисов, предлагаемых отдельными элементами этой подсистемы, т. е., в конечном итоге, классами. Эти элементы могут взаимодействовать друг с другом для совместного обеспечения требуемого поведения отдельного варианта использования. Такое совместное обеспечение требуемого поведения описывается специальным элементом языка UML — *кооперация* или *сотрудничество*, который будет рассмотрен в главе 9, посвященной построению диаграмм кооперации. Здесь лишь отметим, что кооперации используются как для уточнения спецификаций в виде вариантов использования нижних уровней диаграммы, так и для описания особенностей их последующей реализации.

Если в качестве моделируемой сущности выступает система или подсистема самого верхнего уровня, то отдельные пользователи вариантов использования этой системы моделируются актерами. Такие актеры, являясь внутренними по отношению к моделируемым подсистемам нижних уровней, часто в явном виде не указываются, хотя и присутствуют неявно в модели подсистемы. Вместо этого варианты использования непосредственно обращаются к тем модельным элементам, которые содержат в себе подобные неявные актеры, т. е. экземпляры которых играют роли таких актеров при взаимодействии с вариантами использования. Эти модельные элементы могут содержаться в других пакетах или подсистемах. В последнем случае роли определяются в том пакете, к которому относится соответствующая подсистема. С системно-аналитической точки зрения построение диаграммы вариантов использования специфицирует не только функциональные требования к проектируемой системе, но и выполняет исходную структуризацию предметной области. Последняя задача сочетает в себе не только следование техническим рекомендациям, но и является в некотором роде искусством, умением выделять главное в модели системы. Хотя рациональный унифицированный процесс не исключает итеративный возврат в последующем к диаграмме вариантов использования для ее модификации, не вызывает сомнений тот факт, что любая подобная модификация потребует, как по цепочке, изменений во всех других представлениях системы. Поэтому всегда необходимо стремиться к возможно более точному представлению модели именно в форме диаграммы вариантов использования.

Если же варианты использования применяются для спецификации части системы, то они будут эквивалентны соответствующим вариантам использования в модели

подсистемы для части соответствующего пакета. Важно понимать, что все сервисы системы должны быть явно определены на диаграмме вариантов использования, и никаких других сервисов, которые отсутствуют на данной диаграмме, проектируемая система не может выполнять по определению. Более того, если для моделирования реализации системы используются сразу несколько моделей (например, модель анализа и модель проектирования), то множество вариантов использования всех пакетов системы должно быть эквивалентно множеству вариантов использования модели в целом.

Диаграмма вариантов использования состоит из следующих основных компонент:

1. **Вариант использования**

Конструкция или стандартный элемент языка UML *вариант использования* применяется для спецификации общих особенностей поведения системы или любой другой сущности предметной области без рассмотрения внутренней структуры этой сущности. Каждый вариант использования определяет последовательность действий, которые должны быть выполнены проектируемой системой при взаимодействии ее с соответствующим актером. Диаграмма вариантов может дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов. Такой пояснительный текст получил название *примечания* или *сценария*.

Отдельный вариант использования обозначается на диаграмме эллипсом, снизу которого содержится его краткое название или имя в форме глагола с пояснительными словами (рис.1).



Проверить текущее состояние счета клиента в банке

Рис. 1. Графическое обозначение варианта использования

Цель варианта использования заключается в том, чтобы определить законченный аспект или фрагмент поведения некоторой сущности без раскрытия внутренней структуры этой сущности. В качестве такой сущности может выступать исходная система или любой другой элемент модели, который обладает собственным поведением, подобно подсистеме или классу в модели системы.

Каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемую сущность или систему по запросу пользователя (актера), т. е. определяет способ применения этой сущности. Сервис, который инициализируется по запросу пользователя, представляет собой законченную последовательность действий. Это означает, что после того как система закончит обработку запроса пользователя, она должна возвратиться в исходное состояние, в котором готова к выполнению следующих запросов.

Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы. Для удобства множество вариантов использования может рассматриваться как отдельный пакет.

С системно-аналитической точки зрения варианты использования могут применяться как для спецификации внешних требований к проектируемой системе, так и для спецификации функционального поведения уже существующей системы. Кроме этого, варианты использования неявно устанавливают требования, определяющие, как пользователи должны взаимодействовать с системой, чтобы иметь возможность корректно работать с предоставляемыми данной системой сервисами. Применение вариантов использования на всех уровнях диаграммы позволяет не только достичь требуемого уровня унификации обозначений для представления функциональности подсистем и системы в целом, но и является мощным средством последовательного уточнения требований к проектируемой системе на основе поуровневого спуска от пакетов системы к операциям классов. С другой стороны, модификация отдельных операций класса может оказать обратное влияние на уточнение сервиса соответствующего варианта использования, т. е. реализовать эффект обратной связи с целью уточнения спецификаций или требований на уровне пакетов системы. В метамодели UML вариант использования является подклассом классификатора, который описывает последовательности действий, выполняемых отдельным экземпляром варианта использования. Эти действия включают изменения состояния и взаимодействия со средой варианта использования. Эти последовательности могут описываться различными способами, включая такие, как графы деятельности и автоматы.

Примерами вариантов использования могут являться следующие действия: проверка состояния текущего счета клиента, оформление заказа на покупку товара, получение дополнительной информации о кредитоспособности клиента, отображение графической формы на экране монитора и другие действия.

2. Актеры (актанты)

Актер представляет собой любую внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей или решения частных задач. Были рассмотрены в Л.Р.№2.

Актанты не являются частью системы — они представляют собой кого-то или что-то, что должно взаимодействовать с системой. Актанты могут:

- только снабжать информацией систему;
- только получать информацию из системы;
- снабжать информацией и получать информацию из системы.

Обычно актанты определяются из описания задачи или путем переговоров с заказчиками и экспертами. Для выявления актантов может быть использована следующая группа вопросов:

1. Кто заинтересован в определенном системном требовании?
2. Какую роль система будет выполнять в организации?
3. Кто получит преимущества от использования системы?
4. Кто будет снабжать систему информацией, использовать информацию и получать информацию от системы?
5. Кто будет осуществлять поддержку и обслуживание системы?
6. Использует ли система внешние ресурсы?
7. Выступает ли какой-либо участник системы в нескольких ролях?
8. Выступают ли различные участники в одной роли?
9. Будет ли новая система взаимодействовать со старой?

В языке UML актант изображается в виде фигуры человечка - см. рис. 1.



Рис. 1. Изображение актанта в языке UML

В некоторых случаях актант может обозначаться в виде прямоугольника класса с ключевым словом "актант" и обычными составляющими элементами класса. Имена актантов должны записываться заглавными буквами и следовать рекомендациям использования имен для типов и классов модели. При этом символ отдельного актанта связывает соответствующее описание актанта с конкретным именем. Имена абстрактных актантов, как и других абстрактных элементов языка UML, рекомендуется обозначать курсивом.

Примечание

Имя актанта должно быть достаточно информативным с точки зрения семантики. Вполне подходят для этой цели наименования должностей в компании (например, продавец, кассир, менеджер, президент). Не рекомендуется давать актантам имена собственные (например, "О Бендер") или моделей конкретных устройств (например, "маршрутизатор Cisco 3640"), даже если это с очевидностью следует из контекста проекта. Дело в том, что одно и то же лицо может выступать в нескольких ролях и, соответственно, обращаться к различным сервисам системы. Например, посетитель банка может являться как потенциальным клиентом, и тогда он востребует один из его сервисов, а может быть и налоговым инспектором или следователем прокуратуры. Сервис для последнего может быть совершенно исключительным по своему характеру.

Примерами актантов могут быть: клиент банка, банковский служащий, продавец магазина, менеджер отдела продаж, пассажир авиарейса, водитель автомобиля, администратор гостиницы, сотовый телефон и другие сущности, имеющие отношение к концептуальной модели соответствующей предметной области.

Примечание

В метамодели актант является подклассом классификатора. Актанты могут взаимодействовать с множеством вариантов использования и иметь множество интерфейсов, каждый из которых может представлять особенности взаимодействия других элементов с отдельными актантами.

Актанты используются для моделирования внешних по отношению к проектируемой системе сущностей, которые взаимодействуют с системой и используют ее в качестве отдельных пользователей. В качестве актантов могут выступать другие системы, подсистемы проектируемой системы или отдельные классы. Важно понимать, что каждый актант определяет некоторое согласованное множество ролей, в которых могут выступать пользователи данной системы в процессе взаимодействия с ней. В каждый момент времени с системой взаимодействует вполне определенный пользователь, при этом он играет или выступает в одной из таких ролей. Наиболее наглядный пример актанта — конкретный пользователь системы со своими собственными параметрами аутентификации.

Любая сущность, которая согласуется с подобным неформальным определением актанта, представляет собой экземпляр или пример актанта. Для моделируемой системы актантами могут быть как субъекты-пользователи, так и другие системы. Поскольку

пользователи системы всегда являются внешними по отношению к этой системе, то они всегда представляются в виде актантов.

Так как в общем случае актант всегда находится вне системы, его внутренняя структура никак не определяется. Для актанта имеет значение только его внешнее представление, т. е. то, как он воспринимается со стороны системы. Актанты взаимодействуют с системой посредством передачи и приема сообщений от вариантов использования. Сообщение представляет собой запрос актантом сервиса от системы и получение этого сервиса. Это взаимодействие может быть выражено посредством ассоциаций между отдельными актантами и вариантами использования или классами. Кроме этого, с актантами могут быть связаны интерфейсы, которые определяют, каким образом другие элементы модели взаимодействуют с этими актантами.

Два и более актанта могут иметь общие свойства, т. е. взаимодействовать с одним и тем же множеством вариантов использования одинаковым образом. Такая общность свойств и поведения представляется в виде рассматриваемого ниже отношения обобщения с другим, возможно, абстрактным актантом, который моделирует соответствующую общность ролей. Совокупность отношений, которые могут присутствовать на диаграмме вариантов использования, будет рассмотрена ниже в данной главе.

3. **Отношения на диаграмме вариантов использования**

Между компонентами диаграммы вариантов использования могут существовать различные отношения, которые описывают взаимодействие экземпляров одних актеров и вариантов использования с экземплярами других актеров и вариантов. Один актер может взаимодействовать с несколькими вариантами использования. В этом случае этот актер обращается к нескольким сервисам данной системы. В свою очередь один вариант использования может взаимодействовать с несколькими актерами, предоставляя для всех них свой сервис. Следует заметить, что два варианта использования, определенные для одной и той же сущности, не могут взаимодействовать друг с другом, поскольку каждый из них самостоятельно описывает законченный вариант использования этой сущности. Более того, варианты использования всегда предусматривают некоторые сигналы или сообщения, когда взаимодействуют с актерами за пределами системы. В то же время могут быть определены другие способы для взаимодействия с элементами внутри системы.

В языке UML имеется несколько стандартных видов отношений между актерами и вариантами использования:

- Отношение ассоциации (association relationship)
- Отношение расширения (extend relationship)
- Отношение обобщения (generalization relationship)
- Отношение включения (include relationship)

При этом общие свойства вариантов использования могут быть представлены тремя различными способами, а именно с помощью отношений расширения, обобщения и включения.

Отношение ассоциации

Отношение ассоциации является одним из фундаментальных понятий в языке UML и в той или иной степени используется при построении всех графических моделей систем в форме канонических диаграмм. Применительно к диаграммам вариантов использования оно служит для обозначения специфической роли актера в отдельном варианте использования. Другими словами, ассоциация специфицирует семантические особенности взаимодействия

актеров и вариантов использования в графической модели системы. Таким образом, это отношение устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования. На диаграмме вариантов использования, так же как и на других диаграммах, отношение ассоциации обозначается сплошной линией между актером и вариантом использования. Эта линия может иметь дополнительные условные обозначения, такие, например, как имя и кратность (рис. 2).



Рис. 2. Пример графического представления отношения ассоциации между актером и вариантом использования

Кратность (multiplicity) ассоциации указывается рядом с обозначением компонента диаграммы, который является участником данной ассоциации. Кратность характеризует общее количество конкретных экземпляров данного компонента, которые могут выступать в качестве элементов данной ассоциации. Применительно к диаграммам вариантов использования кратность имеет специальное обозначение в форме одной или нескольких цифр и, возможно, специального символа "*" (звездочка).

Примечание

Возвращаясь к общей теории множеств, основы которой были рассмотрены в главе 2, следует заметить, что кратность представляет собой мощность множества экземпляров сущности, участвующей в данной ассоциации. Что касается самого понятия ассоциации, то это одна из наиболее общих форм отношений в языке UML.

Для диаграмм вариантов использования наиболее распространенными являются четыре основные формы записи кратности отношения ассоциации:

Целое неотрицательное число (включая цифру 0). Предназначено для указания кратности, которая является строго фиксированной для элемента соответствующей ассоциации. В этом случае количество экземпляров актеров или вариантов использования, которые могут выступать в качестве элементов отношения ассоциации, в точности равно указанному числу.

Примером этой формы записи кратности ассоциации является указание кратности "1" для актера "Клиент банка" (рис. 2). Эта запись означает, что каждый экземпляр варианта использования "Оформить кредит для клиента банка" может иметь в качестве своего элемента единственный экземпляр актера "Клиент банка". Другими словами, при оформлении кредита в банке необходимо иметь в виду, что каждый конкретный кредит оформляется на единственного клиента этого банка. Два целых неотрицательных числа, разделенные двумя точками и записанные в виде: "первое число ... второе число". Данная запись в языке UML соответствует нотации для множества или интервала целых чисел, которая применяется в некоторых языках программирования для обозначения границ массива элементов. Эту запись следует понимать как множество целых неотрицательных чисел, следующих в последовательно возрастающем порядке:

{первое_число, первое_число+1, первое_число+2, ..., второе_число}. Очевидно, что первое число должно быть строго меньше второго числа в арифметическом смысле, при этом первое число может быть равно 0.

Пример такой формы записи кратности ассоциации — "7..5". Эта запись означает, что количество отдельных экземпляров данного компонента, которые могут выступать в

качестве элементов данной ассоциации, равно некоторому заранее неизвестному числу из множества целых чисел $\{1, 2, 3, 4, 5\}$. Эта ситуация может иметь место, например, в случае рассмотрения в качестве актера — клиента банка, а в качестве варианта использования — процедуру открытия счета в банке. При этом количество отдельных счетов каждого клиента в данном банке, исходя из некоторых дополнительных соображений, может быть не больше 5. Эти дополнительные соображения как раз и являются внешними требованиями по отношению к проектируемой системе и определяются ее заказчиком на начальных этапах ООАП.

Два символа, разделенные двумя точками. При этом первый из них является целым неотрицательным числом или 0, а второй — специальным символом "*". Здесь символ "*" обозначает произвольное конечное целое неотрицательное число, значение которого неизвестно на момент задания соответствующего отношения ассоциации.

Пример такой формы записи кратности ассоциации — "2.. *". Запись означает, что количество отдельных экземпляров данного компонента, которые могут выступать в качестве элементов данной ассоциации, равно некоторому заранее неизвестному числу из подмножества натуральных чисел: $\{2, 3, 4\}$.

Единственный символ "*", который является сокращением записи интервала "0.. *". В этом случае количество отдельных экземпляров данного компонента отношения ассоциации может быть любым целым неотрицательным числом. При этом 0 означает, что для некоторых экземпляров соответствующего компонента данное отношение ассоциации может вовсе не иметь места

В качестве примера этой записи можно привести кратность отношения ассоциации для варианта использования "Оформить кредит для клиента банка" (рис. 2). Здесь кратность "*" означает, что каждый отдельный клиент банка может оформить для себя несколько кредитов, при этом их общее число заранее неизвестно и ничем не ограничивается. При этом некоторые клиенты могут совсем не иметь оформленных на свое имя кредитов (вариант значения 0).

Если кратность отношения ассоциации не указана, то по умолчанию принимается ее значение, равное 1.

Более детальное описание семантических особенностей отношения ассоциации будет дано при рассмотрении других диаграмм в последующих главах книги.

Отношение расширения

Отношение расширения определяет взаимосвязь экземпляров отдельного варианта использования с более общим вариантом, свойства которого определяются на основе способа совместного объединения данных экземпляров. В метамодели отношение расширения является направленным и указывает, что применительно к отдельным примерам некоторого варианта использования должны быть выполнены конкретные условия, определенные для расширения данного варианта использования. Так, если имеет место отношение расширения от варианта использования А к варианту использования В, то это означает, что свойства экземпляра варианта использования В могут быть дополнены благодаря наличию свойств у расширенного варианта использования А.

Отношение расширения между вариантами использования 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" ("расширяет"), как показано на рис. 3.

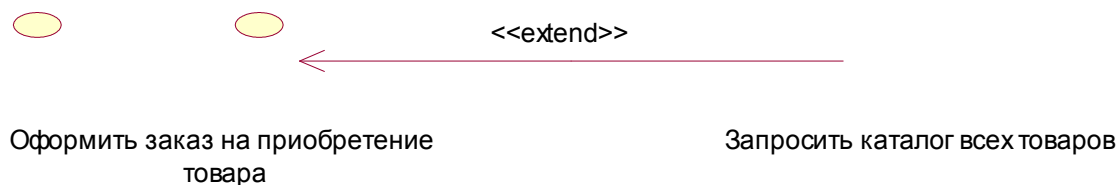


Рис. 3. Пример графического изображения отношения расширения между вариантами использования

Отношение расширения отмечает тот факт, что один из вариантов использования может присоединять к своему поведению некоторое дополнительное поведение, определенное для другого варианта использования. Данное отношение включает в себя некоторое условие и ссылки на точки расширения в базовом варианте использования. Чтобы расширение имело место, должно быть выполнено определенное условие данного отношения. Ссылки на точки расширения определяют те места в базовом варианте использования, в которые должно быть помещено соответствующее расширение при выполнении условия. Один из вариантов использования может быть расширением для нескольких базовых вариантов, а также иметь в качестве собственных расширений несколько других вариантов. Базовый вариант использования может дополнительно никак не зависеть от своих расширений.

Семантика отношения расширения определяется следующим образом. Если экземпляр варианта использования выполняет некоторую последовательность действий, которая определяет его поведение, и при этом имеется точка расширения на экземпляр другого варианта использования, которая является первой из всех точек расширения у исходного варианта, то проверяется условие данного отношения. Если условие выполняется, исходная последовательность действий расширяется посредством включения действий экземпляра другого варианта использования. Следует заметить, что условие отношения расширения проверяется лишь один раз — при первой ссылке на точку расширения, и если оно выполняется, то все расширяющие варианты использования вставляются в базовый вариант.

В представленном выше примере (рис. 3) при оформлении заказа на приобретение товара только в некоторых случаях может потребоваться предоставление клиенту каталога всех товаров. При этом условием расширения является запрос от клиента на получение каталога товаров. Очевидно, что после получения каталога клиенту необходимо некоторое время на его изучение, в течение которого оформление заказа приостанавливается. После ознакомления с каталогом клиент решает либо в пользу выбора отдельного товара, либо отказа от покупки вообще. Сервис или вариант использования "Оформить заказ на приобретение товара" может отреагировать на выбор клиента уже после того, как клиент получит для ознакомления каталог товаров.

Точка расширения может быть как отдельной точкой в последовательности действий, так и множеством отдельных точек. Важно представлять себе, что если отношение расширения имеет некоторую последовательность точек расширения, только первая из них может определять множество отдельных точек. Все остальные должны определять в точности одну такую точку. Какая из точек должна быть первой точкой расширения, т. е. определяться единственным расширением. Такие ссылки на расположение точек расширения могут быть представлены различными способами, например, с помощью текста примечания на естественном языке, пред- и постусловий, а также с использованием имен состояний в автомате.

Отношение обобщения

Отношение обобщения служит для указания того факта, что некоторый вариант использования А может быть обобщен до варианта использования В. В этом случае вариант А будет являться специализацией варианта В. При этом В называется предком или

родителем по отношению А, а вариант А — потомком по отношению к варианту использования В. Следует подчеркнуть, что потомок наследует все свойства и поведение своего родителя, а также может быть дополнен новыми свойствами и особенностями поведения. Графически данное отношение обозначается сплошной линией со стрелкой в форме не закрашенного треугольника, которая указывает на родительский вариант использования (рис. 4). Эта линия со стрелкой имеет специальное название — стрелка "обобщение".

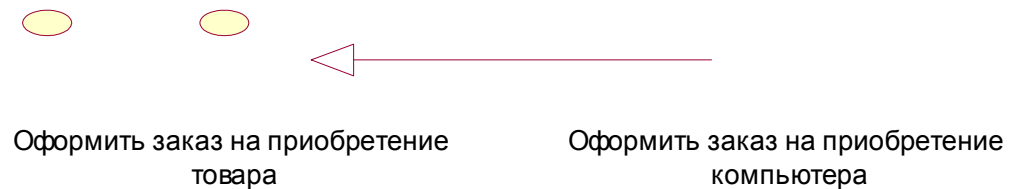


Рис. 4. Пример графического изображения отношения обобщения между вариантами использования

Отношение обобщения между вариантами использования применяется в том случае, когда необходимо отметить, что дочерние варианты использования обладают всеми атрибутами и особенностями поведения родительских вариантов. При этом дочерние варианты использования участвуют во всех отношениях родительских вариантов. В свою очередь, дочерние варианты могут наделяться новыми свойствами поведения, которые отсутствуют у родительских вариантов использования, а также уточнять или модифицировать наследуемые от них свойства поведения.

Применительно к данному отношению, один вариант использования может иметь несколько родительских вариантов. В этом случае реализуется множественное наследование свойств и поведения отношения предков. С другой стороны, один вариант использования может быть предком для нескольких дочерних вариантов, что соответствует таксономическому характеру отношения обобщения.

Между отдельными актерами также может существовать отношение обобщения. Данное отношение является направленным и указывает на факт специализации одних актеров относительно других. Например, отношение обобщения от актера А к актеру В отмечает тот факт, что каждый экземпляр актера А является одновременно экземпляром актера В и обладает всеми его свойствами. В этом случае актер В является родителем по отношению к актеру А, а актер А, соответственно, потомком актера В. При этом актер А обладает способностью играть такое же множество ролей, что и актер В. Графически данное отношение также обозначается стрелкой обобщения, т. е. сплошной линией со стрелкой в форме не закрашенного треугольника, которая указывает на родительского актера (рис. 5).



Рис. 5. Пример графического изображения отношения обобщения между актерами

Отношение 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. Данное отношение является направленным бинарным отношением в том смысле, что пара экземпляров вариантов использования всегда упорядочена в отношении включения.

Семантика этого отношения определяется следующим образом. Когда экземпляр первого варианта использования в процессе своего выполнения достигает точки включения в последовательность поведения экземпляра второго варианта использования, экземпляр первого варианта использования выполняет последовательность действий, определяющую поведение экземпляра второго варианта использования, после чего продолжает выполнение действий своего поведения. При этом предполагается, что даже если экземпляр первого варианта использования может иметь несколько включаемых в себя экземпляров других вариантов, выполняемые ими действия должны закончиться к некоторому моменту, после чего должно быть продолжено выполнение прерванных действий экземпляра первого варианта использования в соответствии с заданным для него поведением.

Один вариант использования может быть включен в несколько других вариантов, а также включать в себя другие варианты. Включаемый вариант использования может быть независимым от базового варианта в том смысле, что он предоставляет последнему некоторое инкапсулированное поведение, детали реализации которого скрыты от последнего и могут быть легко перераспределены между несколькими включаемыми вариантами использования. Более того, базовый вариант может зависеть только от результатов выполнения включаемого в него поведения, но не от структуры включаемых в него вариантов.

Отношение включения, направленное от варианта использования А к варианту использования В, указывает, что каждый экземпляр варианта А включает в себя функциональные свойства, заданные для варианта В. Эти свойства специализируют поведение соответствующего варианта А на данной диаграмме. Графически данное отношение обозначается пунктирной линией со стрелкой (вариант отношения зависимости), направленной от базового варианта использования к включаемому. При этом данная линия со стрелкой помечается ключевым словом "include" ("включает"), как показано на рис. 6.

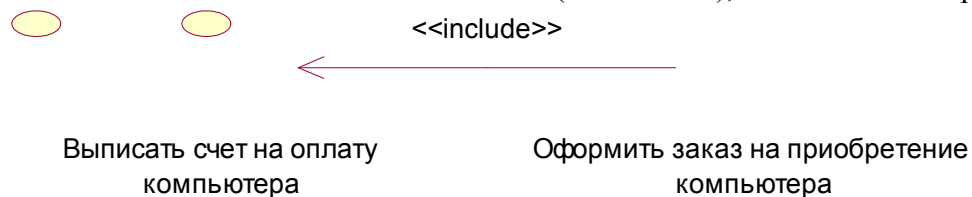


Рис. 6. Пример графического изображения отношения включения между вариантами использования

Примечание

Следует заметить, что рассмотренные три последних отношения могут существовать только между вариантами использования, которые определены для одной и той же сущности. Причина этого заключается в том, что поведение некоторой сущности обусловлено вариантами использования только этой сущности. Другими словами, все экземпляры варианта использования выполняются лишь внутри данной сущности. Если некоторый вариант использования должен иметь отношение обобщения, включения или расширения с вариантом использования другой сущности, получаемые в результате экземпляры вариантов должны быть включены в обе сущности, что противоречит семантическим правилам представления элементов языка UML. Однако эти отношения, определенные в пределах одной сущности, могут быть использованы в пределах другой сущности, если обе сущности связаны между собой отношением обобщения. В этом случае

поведение соответствующих вариантов использования подчиняется общим правилам наследования свойств и поведения сущности-предка всеми дочерними сущностями.

Пример построения диаграммы вариантов использования

В качестве примера рассмотрим процесс моделирования системы продажи товаров по каталогу, которая может быть использована при создании соответствующих информационных систем.

В качестве актеров данной системы могут выступать два субъекта, один из которых является продавцом, а другой — покупателем. Каждый из этих актеров взаимодействует с рассматриваемой системой продажи товаров по каталогу и является ее пользователем, т. е. они оба обращаются к соответствующему сервису "Оформить заказ на покупку товара". Как следует из существа выдвигаемых к системе требований, этот сервис выступает в качестве варианта использования разрабатываемой диаграммы, первоначальная структура которой может включать в себя только двух указанных актеров и единственный вариант использования (рис. 7).

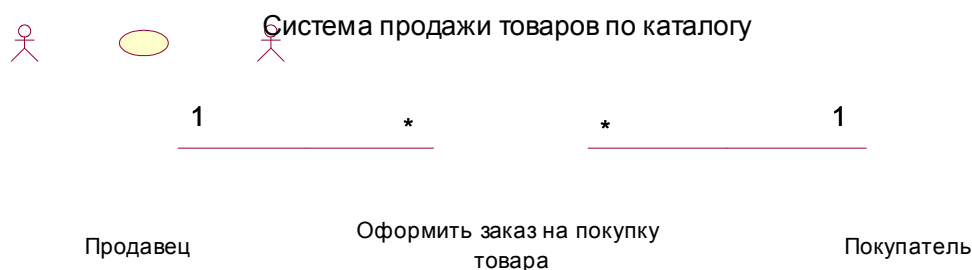


Рис. 7. Исходная диаграмма вариантов использования для примера разработки системы продажи товаров по каталогу

Значения указанных на данной диаграмме кратностей отражают общие правила или логику оформления заказов на покупку товаров. Согласно этим правилам, один продавец может участвовать в оформлении нескольких заказов, в то же время каждый заказ может быть оформлен только одним продавцом, который несет ответственность за корректность его оформления и, в связи с этим, будет иметь агентское вознаграждение за его оформление. С другой стороны, каждый покупатель может оформлять на себя несколько заказов, но, в то же время, каждый заказ должен быть оформлен на единственного покупателя, к которому переходят права собственности на товар после его оплаты.

На следующем этапе разработки данной диаграммы вариант использования "Оформить заказ на покупку товара" может быть уточнен на основе введения в рассмотрение четырех дополнительных вариантов использования. Это следует из более детального анализа процесса продажи товаров, что позволяет выделить в качестве отдельных сервисов такие действия, как обеспечить покупателя информацией о товаре, согласовать условия оплаты товара и заказать товар со склада. Вполне очевидно, что указанные действия раскрывают поведение исходного варианта использования в смысле его конкретизации, и поэтому между ними будет иметь место отношение включения.

С другой стороны, продажа товаров по каталогу предполагает наличие самостоятельного информационного объекта — каталога товаров, который в некотором смысле не зависит от реализации сервиса по обслуживанию покупателей. В нашем случае, каталог товаров может запрашиваться покупателем или продавцом при необходимости выбора

товара и уточнения деталей его продажи. Вполне резонно представить сервис "Запросить каталог товаров" в качестве самостоятельного варианта использования.

Полученная в результате последующей детализации уточненная диаграмма вариантов использования будет содержать 5 вариантов использования и 2 актеров (рис. 8), между которыми установлены отношения включения и расширения.



Рис. 8. Уточненный вариант диаграммы вариантов использования для примера системы продажи товаров по каталогу

Приведенная выше диаграмма вариантов использования, в свою очередь, может быть детализирована далее с целью более глубокого уточнения предъявляемых к системе требований и конкретизации деталей ее последующей реализации. В рамках общей парадигмы ООАП подобная детализация может выполняться в двух основных направлениях.

С одной стороны, детализация может быть выполнена на основе установления дополнительных отношений типа отношения "обобщение-специализация" для уже имеющихся компонентов диаграммы вариантов использования. Так, в рамках рассматриваемой системы продажи товаров может иметь самостоятельное значение и специфические особенности отдельная категория товаров — компьютеры. В этом случае диаграмма может быть дополнена вариантом использования "Оформить заказ на покупку компьютера" и актерами "Покупатель компьютера" и "Продавец компьютеров", которые связаны с соответствующими компонентами диаграммы отношением обобщения (рис. 8).

Уточненный таким способом вариант диаграммы вариантов использования содержит одну важную особенность, которую необходимо отметить. А именно, хотя на данной диаграмме (рис. 8) отсутствуют изображения линий отношения ассоциации между актером "Продавец компьютеров" и вариантом использования "Оформить заказ на покупку компьютера", а также между актером "Покупатель компьютера" и вариантом использования "Оформить заказ на покупку компьютера", наличие отношения обобщения между соответствующими компонентами позволяет им наследовать отношение ассоциации от своих предков. Поскольку принцип наследования является одним из фундаментальных принципов объектно-ориентированного программирования, в нашем примере можно с уверенностью утверждать, что эти линии отношения ассоциации с соответствующими кратностями присутствуют на данной диаграмме в скрытом виде.



Рис. 9. Один из вариантов последующего уточнения диаграммы вариантов использования для примера рассматриваемой системы продажи

Второе из основных направлений детализации диаграмм вариантов использования связано с последующей структуризацией ее отдельных компонентов в форме элементов других диаграмм. Например, конкретные особенности реализации вариантов использования в терминах взаимодействующих объектов, определенных в виде классов данной сущности, могут быть заданы на диаграмме кооперации. Спецификация требований к проектируемой системе в форме диаграммы вариантов использования представляет собой самостоятельную модель, которая в языке UML получила название модели вариантов использования и имеет свое специальное стандартное имя или стереотип "useCaseModel"