

диаграммы последовательности (sequence diagrams) и кооперативные диаграммы (collaboration diagrams) — для моделирования процесса обмена сообщениями между объектами;

диаграммы состояний (statechart diagrams) — для моделирования поведения объектов системы при переходе из одного состояния в другое;

диаграммы деятельности (activity diagrams) — для моделирования поведения системы в рамках различных вариантов использования, или потоков управления.

2.5.1.

ДИАГРАММЫ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

Идея описания функциональных требований в виде вариантов использования (use case) была сформулирована в 1986 г. Иваром Якобсоном. Эта идея была признана конструктивной и получила широкое одобрение. Впоследствии наиболее значительный вклад в решение проблемы определения сущности вариантов использования и способов их описания внес Алистер Коберн².

Вариант использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). Вариант использования описывает типичное взаимодействие между пользователем и системой и отражает представление о поведении системы с точки зрения пользователя. В простейшем случае вариант использования определяется в процессе обсуждения с пользователем тех функций, которые он хотел бы реализовать, или целей, которые он преследует по отношению к разрабатываемой системе.

Действующее лицо (actor) — это роль, которую пользователь играет по отношению к системе. Действующие лица представляют собой роли, а не конкретных людей или наименования работ.

² Коберн А. Современные методы описания функциональных требований к системам.:

Пер. с англ. - М.: ЛОРИ, 2002

Несмотря на то что на диаграммах вариантов использования они изображаются в виде стилизованных человеческих фигурок, действующее лицо может также быть внешней системой, которой необходима некоторая информация от данной системы.

Действующие лица делятся на три основных типа — пользователи системы, другие системы, взаимодействующие с данной, и время. Время становится действующим лицом, если от него зависит запуск каких-либо событий в системе.

Для наглядного представления вариантов использования применяются диаграммы вариантов использования. На рис. 2.47 показан пример такой диаграммы для банковской системы.

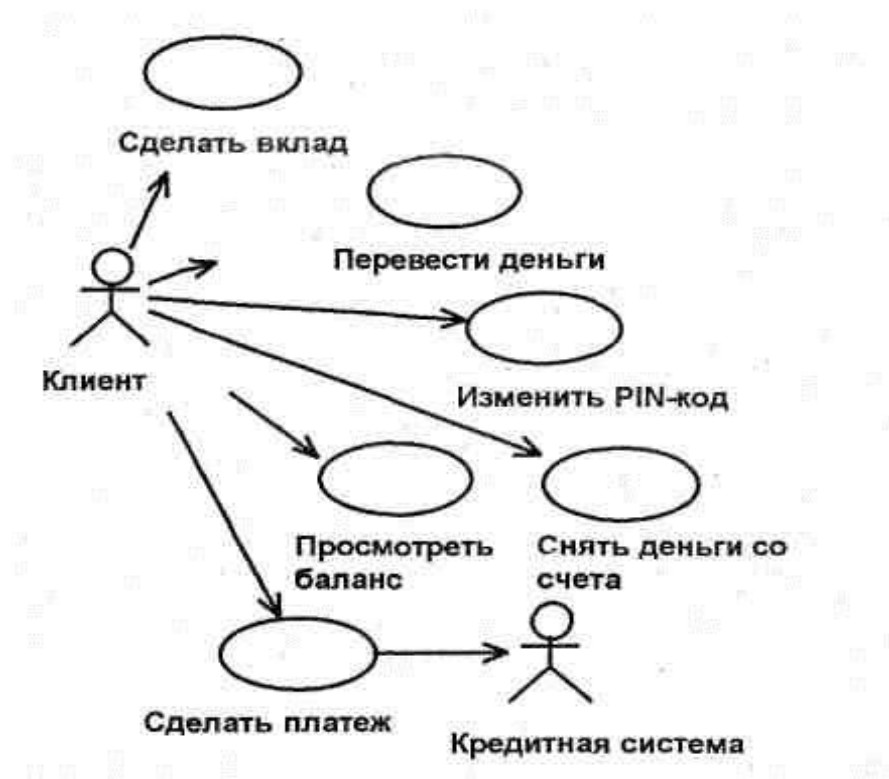


Рис. 2.47. Пример диаграммы вариантов использования

На данной диаграмме человеческие фигурки обозначают действующих лиц, овалы — варианты использования, а линии и стрелки — различные связи между действующими лицами и вариантами использования. На этой диаграмме показаны два действующих лица: клиент и кредитная система. Существует также шесть основных действий, выполняемых моделируемой

системой: перевести деньги, сделать вклад, снять деньги со счета, просмотреть баланс, изменить PIN-код и сделать платеж.

На диаграмме вариантов использования показано взаимодействие между вариантами использования и действующими лицами. Она отражает функциональные требования к системе с точки зрения пользователя. Таким образом, варианты использования — это функции, выполняемые системой, а действующие лица — это заинтересованные лица (stakeholders) по отношению к создаваемой системе. Такие диаграммы показывают, какие действующие лица инициируют варианты использования. Из них также видно, когда действующее лицо получает информацию от варианта использования. Направленная от варианта использования к действующему лицу стрелка показывает, что вариант использования предоставляет некоторую информацию, используемую действующим лицом. В данном случае вариант использования «Сделать платеж» предоставляет Кредитной системе информацию об оплате по кредитной карточке.

Действующие лица могут играть различные роли по отношению к варианту использования. Они могут пользоваться его результатами сами непосредственно в нем участвовать. Значимость различных ролей действующего лица зависит от того, каким образом используются его связи.

Цель построения диаграмм вариантов использования — документирование функциональных требований к системе в самом общем виде, поэтому они должны быть предельно простыми. При построении диаграмм вариантов использования нужно придерживаться следующих правил:

- Не моделируйте связи между действующими лицами. По определению действующие лица находятся вне сферы действия системы. Это означает, что связи между ними также не относятся к ее компетенции.
- Не соединяйте стрелкой два варианта использования непосредственно. Диаграммы данного типа описывают только сами варианты использования, а не порядок их выполнения. Для отображения порядка выполнения вариантов использования применяют диаграммы

деятельности.

- Каждый вариант использования должен быть инициирован действующим лицом. Это означает, что всегда должна быть стрелка, начинающаяся на действующем лице и заканчивающаяся на варианте использования.

Хорошим источником для идентификации вариантов использования служат внешние события. Следует начать с перечисления всех событий, происходящих во внешнем мире, на которые система должна каким-то образом реагировать. Какое-либо конкретное событие может повлечь за собой реакцию системы, не требующую вмешательства пользователей, или, наоборот, вызвать чисто пользовательскую реакцию. Идентификация событий, на которые необходимо реагировать, помогает идентифицировать варианты использования.

Диаграмма вариантов использования является самым общим представлением функциональных требований к системе. Однако моделирование вариантов использования не сводится только к рисованию диаграмм. Для последующего проектирования системы требуются более конкретные детали. Эти детали описываются в документе, называемом **«сценарий варианта использования»** или **«поток событий» (flow of events)**. Целью потока событий является подробное документирование процесса взаимодействия действующего лица с системой, реализуемого в рамках варианта использования. В потоке событий должно быть описано все, что служит удовлетворению запросов действующих лиц.

Хотя поток событий и описывается подробно, он также не должен зависеть от реализации. Цель - описать, что будет делать система, а не как она будет делать это. Обычно описание потока событий включает следующие разделы:

- краткое описание;
- предусловия (pre-conditions);
- основной поток событий;
- альтернативные потоки событий;

- постусловия (post-conditions);
- расширения (extensions).

Последовательно рассмотрим эти составные части.

Краткое описание. Каждый вариант использования должен иметь краткое описание того, что в нем происходит. Например, вариант использования «Перевести деньги» может содержать следующее описание.

Вариант использования «Перевести деньги» позволяет клиенту или служащему банка переводить деньги с одного счета до востребования или сберегательного счета на другой.

Предусловия. Предусловия варианта использования - это такие условия, которые должны быть выполнены, прежде чем вариант использования начнет выполняться сам. Например, таким условием может быть выполнение другого варианта использования или наличие у пользователя прав доступа, требуемых для начала работы. Не у всех вариантов использования бывают предусловия. Ранее упоминалось, что диаграммы вариантов использования не должны отражать порядок их выполнения. Такую информацию можно описать с помощью предусловий. Например, предусловием одного варианта использования может быть то, что в это время должен выполняться другой.

Основной и альтернативный потоки событий. Конкретные детали вариантов использования описываются в основном в альтернативных потоках событий. Поток событий поэтапно описывает, что должно происходить во время выполнения заложенной в варианты использования функциональности. Поток событий уделяет внимание тому, *что* будет делать система, а не *как* она будет делать это, причем описывает все это с точки зрения пользователя. Основной поток событий описывает нормальный ход событий (при отсутствии ошибок), и при наличии нескольких возможных вариантов хода событий может разветвляться на подчиненные потоки (subflow). Альтернативные потоки описывают отклонения от нормального хода событий (ошибочные ситуации) и их обработку. Например, потоки событий варианта использования «Снять деньги со счета» могут выглядеть следующим образом:

Основной поток событий

1. Вариант использования начинается, когда клиент вставляет свою карточку в банкомат.
2. Банкомат выводит приветствие и предлагает клиенту ввести свой персональный PIN-код.
3. Клиент вводит PIN-код.
4. Банкомат подтверждает введенный код.
5. Банкомат выводит список доступных действий: сделать вклад, снять деньги со счета, перевести деньги
6. Клиент выбирает пункт «Снять деньги со счета».
7. Банкомат спрашивает, сколько денег надо снять.
8. Клиент вводит требуемую сумму.
9. Банкомат определяет, имеется ли на счету достаточно денег.
10. Банкомат вычитает требуемую сумму из счета клиента.
11. Банкомат выдает клиенту требуемую сумму наличными.
12. Банкомат возвращает клиенту его карточку.
13. Банкомат печатает чек для клиента.
14. Вариант использования завершается.

Альтернативный поток событий 1. Ввод неправильного PIN-кода.

- 4a1. Банкомат информирует клиента, что код введен неправильно.
- 4a2. Банкомат возвращает клиенту его карточку.
- 4a3. Вариант использования завершается.

Альтернативный поток событий 2. Недостаточно денег на счете.

- 9a1. Банкомат информирует клиента, что денег на его счете недостаточно.
- 9a2. Банкомат возвращает клиенту его карточку.

9a3. Вариант использования завершается.

Альтернативный поток событий 3. Ошибка в подтверждении запрашиваемой суммы.

9б1. Банкомат сообщает пользователю, что при подтверждении запрашиваемой суммы произошла ошибка, и дает ему номер телефона службы поддержки клиентов банка.

9б2. Банкомат заносит сведения об ошибке в журнал ошибок. Каждая запись содержит дату и время ошибки, имя клиента, номер его счета и код ошибки.

9б3. Банкомат возвращает клиенту его карточку.

9б4. Вариант использования завершается.

Как видно из приведенного примера, хорошо написанный поток событий должен легко читаться и состоять из предложений, написанных в единой грамматической форме. На обучение его чтению не должно уходить больше нескольких минут. При написании основного потока событий нужно придерживаться следующих правил:

- использовать простые предложения;
- явно указывать в каждом пункте, кто выполняет действие — действующее лицо или система;
- не показывать слишком незначительные действия;
- не показывать детальные действия пользователя в процессе работы с пользовательским интерфейсом;
- не рассматривать возможные ошибочные ситуации (использовать действия «подтвердить», а не «проверить»).

При выявлении альтернативных потоков событий нужно в первую очередь обратить внимание на ситуации, связанные с:

- некорректными действиями пользователя (например, ввод неверного пароля);
- бездействием действующего лица (например, истечением времени

- ожидания пароля);
- внутренними ошибками в разрабатываемой системе, которые должны быть обнаружены и обработаны в обычном порядке (например, заблокирован автомат для выдачи наличных);
 - критически важными недостатками в производительности системы (например, время реакции не укладывается в 5 секунд).

Постусловия. Постусловиями называются такие условия, которые всегда должны быть выполнены после завершения варианта использования. Например, в конце варианта использования можно пометить флажком какой-нибудь переключатель. Информация такого типа входит в состав постусловий. Как и для предусловий, с помощью постусловий можно вводить информацию о порядке выполнения вариантов использования системы. Если, например, после одного из вариантов использования должен всегда выполняться другой, это можно описать как постусловие. Такие условия имеются не у каждого варианта использования.

Расширения. Этот пункт присутствует, если в основном потоке событий имеют место относительно редко встречающиеся ситуации (частные случаи). Описание таких ситуаций выносится в данный пункт.

В диаграммах вариантов использования может присутствовать несколько типов связей. Это связи коммуникации (communication), включения (include), расширения (extend) и обобщения (generalization).

Связь коммуникации — это связь между вариантом использования и действующим лицом, она изображается с помощью однонаправленной ассоциации (линии со стрелкой). Направление стрелки позволяет понять, кто инициирует коммуникацию.

Связь включения применяется в тех ситуациях, когда имеется какой-либо фрагмент поведения системы (часть потока событий), который повторяется более чем в одном варианте использования. С помощью таких связей обычно моделируют многократно используемую функциональность. В примере с банковской системой варианты использования «Снять деньги со счета» и

«Сделать вклад» должны аутентифицировать клиента и его PIN-код перед тем, как допустить осуществление самой транзакции. Вместо того чтобы подробно описывать процесс аутентификации для каждого из вариантов использования, можно поместить эту функциональность в свой собственный вариант использования под названием «Аутентифицировать клиента».

Связь расширения применяется при наличии изменений в нормальном поведении системы (описанных в пункте «Расширения»), которые также выносятся в отдельный вариант использования.

Связи включения и расширения изображаются в виде зависимостей, как показано на рис. 2.48.



Рис. 2.48. Связи включения и расширения

С помощью связи обобщения показывают, что у нескольких действующих лиц имеются общие черты и различия. Например, у служащих организации имеются как общие свойства, так и разные способы оплаты труда (рис. 2.49).

Нет необходимости всегда создавать связи этого типа. В общем случае они нужны, если отличия в поведении действующего лица одного типа от поведения другого затрагивают функции системы. Если оба подтипа используют одни и те же варианты использования, показывать обобщение действующего лица не требуется.

Варианты использования являются необходимым средством на стадии формирования требований к ПО. Каждый вариант использования — это потен-

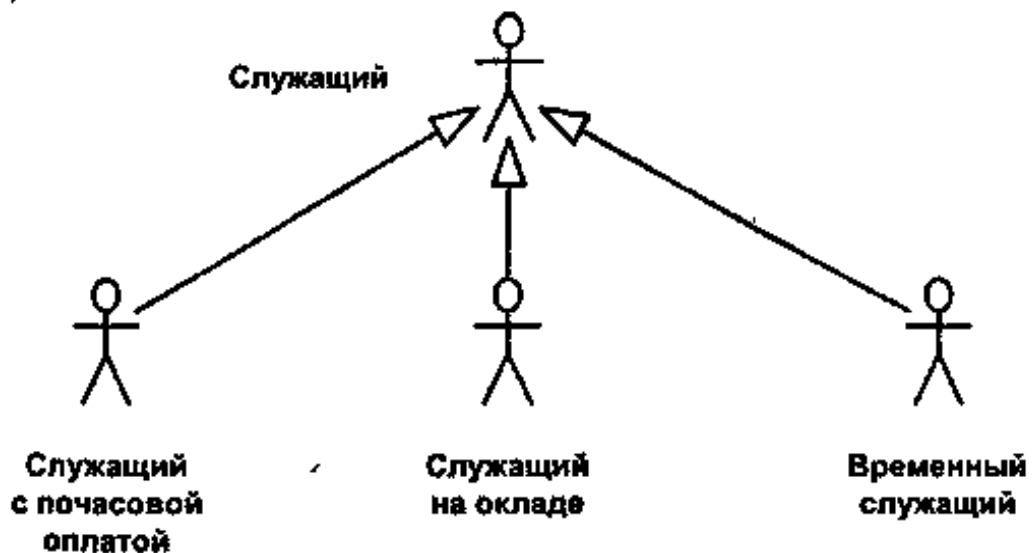


Рис. 2.49. Обобщение действующего лица

циальное требование к системе, и пока оно не выявлено, невозможно запланировать его реализацию.

Различные разработчики подходят к описанию вариантов использования с разной степенью детализации. Например, Ивар Якобсон утверждал, что для проекта с трудоемкостью 10 человеко-лет количество вариантов использования может составлять около 20 (не считая связей «включения» и «расширения»). Следует предпочитать небольшие и детализированные варианты использования, поскольку они облегчают составление и реализацию согласованного плана проекта.

Достоинства модели вариантов использования заключаются в том, что она:

- определяет пользователей и границы системы;
- определяет системный интерфейс;
- удобна для общения пользователей с разработчиками;
- используется для написания тестов;
- является основой для написания пользовательской документации;
- хорошо вписывается в любые методы проектирования (как объектно-ориентированные, так и структурные).