

Технологии разработки программного обеспечения

Составитель: Эверстов В.В.

Дата составления: 2.11.2010

Дата модификации: 25.11.2011

Модель взаимодействия

- Моделирование взаимодействия следует начинать с определения внешней границы системы. Затем следует выделить варианты использования и детально их проработать их в сценариях и на диаграммах последовательности. Для сложных и неочевидных вариантов использования нужно построить диаграммы деятельности. Далее нужно упорядочить варианты использования с помощью отношений. И, наконец, проверить модель взаимодействия по модели классов предметной области: несогласованности между ними быть не должно.

Этапы

- Модель взаимодействия строиться в несколько этапов:
 - 1) Определение границы системы,
 - 2) Выделение действующих лиц,
 - 3) Выделение вариантов использования,
 - 4) Выделение начальных и конечных событий,
 - 5) Подготовка типовых сценариев,
 - 6) Добавление сценариев, описывающих вариации и исключительные ситуации,
 - 7) Выделение внешних событий,
 - 8) Построение диаграммы деятельности для сложных вариантов использования,
 - 9) Структурирование действующих лиц и вариантов использования,
 - 10) Проверка по модели классов предметной области.

Определение границ

- Для определения функциональности нужно точно знать область приложения, т.е. границы системы. Вы должны решить, что будет входить в вашу систему, а что нет. Если граница определена корректно, во всех взаимодействиях вы сможете рассматривать систему как черный ящик – единый объект, внутренние особенности которого скрыты от вашего взгляда и могут быть реализованы по-разному.

Определение границ

- Обычно людей не следует рассматривать как часть системы, если только вы не занимаетесь моделированием организации как таковой. Люди – это действующие лица, которые должны взаимодействовать с системой.

Идентификация действующих лиц

- После того как вы определили внешние границы системы вы должны идентифицировать внешние объекты, непосредственно взаимодействующие с системой. Это и будут действующие лица. К их числу относятся люди, внешние устройства и другие программные продукты.
- Самым важным качеством действующих лиц является то, что они не контролируются приложением, а их поведение должно считаться непредсказуемым.

Идентификация действующих лиц

- Каждое действующее лицо представляет собой абстрактного пользователя, который задействует какое-либо подмножество функциональности системы.
- Изучите каждый внешний объект и выясните не характеризуется ли он существенно различными видами поведения. Действующее лицо это один вариант поведения по отношению к системе, а один внешний объект может соответствовать нескольким действующим лицам. С другой стороны, разные типы внешних объектов могут описываться одним действующим лицом.

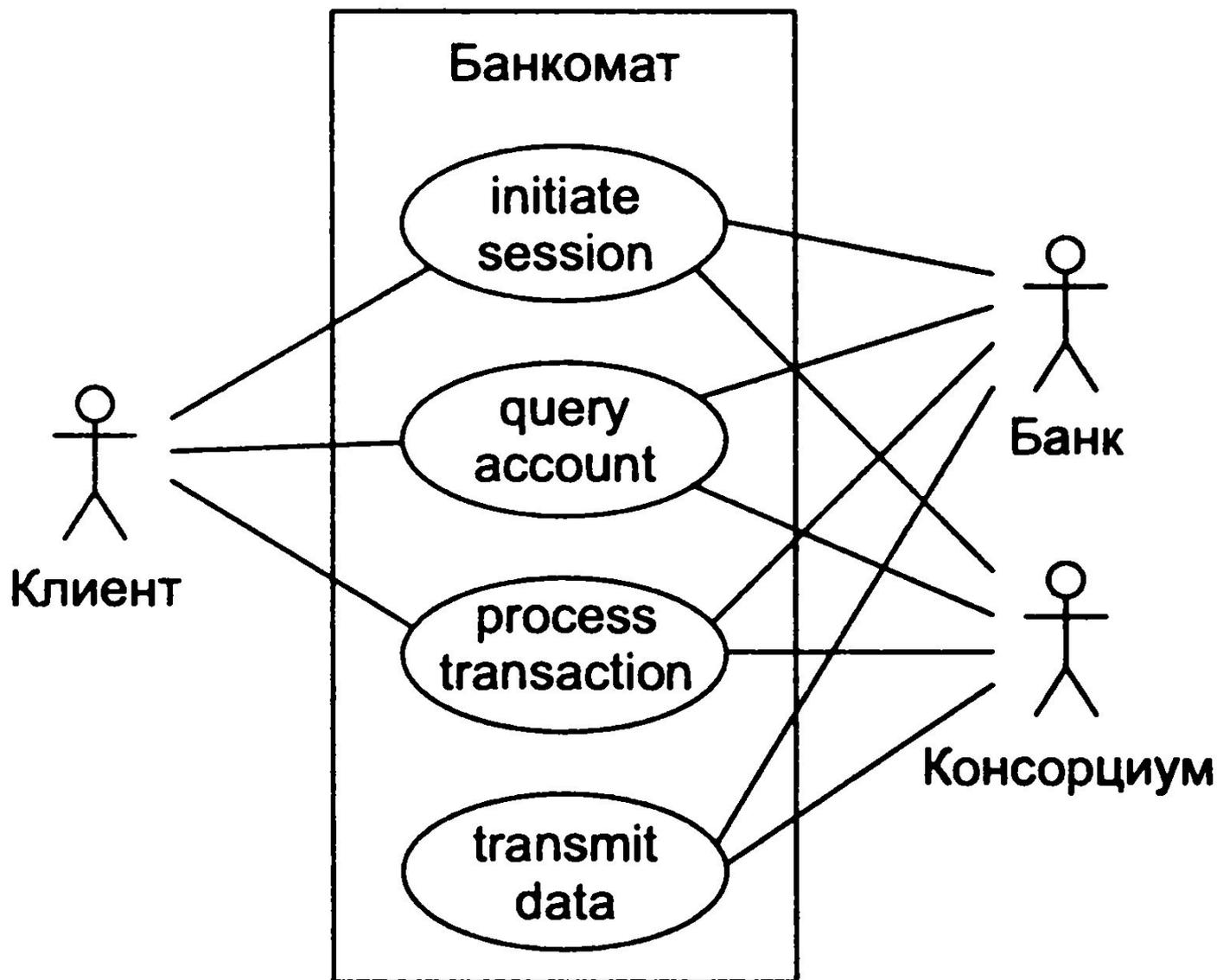
Варианты использования

- Для каждого действующего лица необходимо перечислить разные способы использования системы. Каждый из этих способов называется вариантом использования.
- Любое поведение системы должно принадлежать к тому или иному варианту использования.

Варианты использования

- Каждый вариант использования должен представлять собой некий сервис, представляемый системой, т.е. нечто ценное для действующего лица. Все варианты использования следует рассматривать на одном уровне детализации.
- На этом этапе вы можете нарисовать предварительную диаграмму вариантов использования. На ней следует показать действующие лица и варианты использования, соединив их между собой. Кроме того для каждого варианта использования полезно сформулировать краткое описание в одно-два предложения.

Пример



Начальные и конечные события

- Варианты использования разбивают функциональность системы на дискретные части и показывают действующие лица, использующие каждую из этих частей. Для понимания поведения необходимо знать последовательность выполнения соответствующие каждому из вариантов использования. Начать их анализ следует с поиска событий, инициирующих каждый из вариантов использования. Определите, какое лицо инициирует вариант использования, и определите событие, которое оно для этого передает системе.

Начальные и конечные события

- Во многих случаях начальным событием является запрос некоторой услуги, предоставляемой системой. В других случаях начальным событием является происшествие, запускающее цепочку действий. Дайте этому событию осмысленное название (не надо пока определять полный список его параметров).

Начальные и конечные события

- Кроме того, следует определить конечное событие или группу событий, а также общие рамки событий, которые должны быть включены в каждый из вариантов использования.
- Разработчики должны определить границы варианта использования, установив его конечное событие.

Пример

- **Initiate session (Инициализация сеанса).** Начальным событием является помещение клиентом банковской карты в банкомат. Конечных событий может быть два: либо система оставляет банковскую карту себе, либо возвращает ее обратно клиенту.
- **Query Account (Опрос счета).** Начальное событие: клиент запрашивает данные о состоянии счета. Конечное событие: выдача необходимых сведений клиенту.
- **Process transaction (Обработка транзакции).** Начальное событие: Клиент инициирует транзакцию. Конечных событий может быть два: завершение или откат транзакции.
- **Transmit Data (Передача данных).** Начальным событием может быть запрос данных о состоянии счета. Кроме того, передача данных может быть инициирована после устранения неполадок в сети или перебоев с питанием. Конечное событие: успешная передача данных.

Подготовка типовых сценариев

- Для каждого варианта использования нужно подготовить один или несколько типичных сценариев, чтобы почувствовать ожидаемое поведение системы. Эти сценарии описывают основные взаимодействия, форматы внешних отображаемых данных, а также обмен информацией.
- Сценарием называется последовательность событий на множестве взаимодействующих объектов.

Подготовка типовых сценариев

- Для большинства задач логическая корректность зависит от взаимной последовательности взаимодействий, а не конкретных временных промежутков между ними.
- Иногда описание задачи полностью задает последовательность взаимодействия, но в большинстве случаев вам придется придумывать ее (или по крайней мере конкретизировать ее).

Подготовка типовых сценариев

- Например, в постановке задачи о банкомате говорится о необходимости получения данных о транзакции от пользователя, но не указывается, какие конкретно параметры нужно у него спрашивать и в какой последовательности. Старайтесь не углубляться в подобные тонкости на этапе анализа. Для большинства приложений порядок ввода информации не имеет особой важности и может быть отложен до этапа проектирования.

Подготовка типовых сценариев

- Подготовьте сценарии для типовых ситуаций – взаимодействие без необычных параметров и ошибочных ситуаций. Событием является обмен информацией между объектом системы и внешним агентом (пользователем, датчиком и т.д.). Параметром события является передаваемая информация. События без параметров тоже важны и достаточно распространены.
- Для каждого события необходимо указать вызвавшее его лицо и параметры события.

*Инициация
сеанса*

Банкомат просит пользователя вставить кредитную карту.
Пользователь вставляет кредитную карту.
Банкомат принимает карту и считывает ее серийный номер.
Банкомат запрашивает пароль.
Пользователь вводит «1234».
Банкомат проверяет пароль, связываясь с консорциумом и банком.
Банкомат выводит меню действий над счетами и команд.
...
Пользователь выбирает команду завершения сеанса.
Банкомат печатает квитанцию, возвращает карту и просит пользователя взять ее из банкомата.
Пользователь берет квитанцию и карту.
Банкомат просит пользователя вставить кредитную карту.

*Запрос
счета*

Банкомат выводит меню счетов и команд.
Пользователь выбирает запрос счета.
Банкомат связывается с консорциумом и банком, которые предоставляют данные.
Банкомат выводит данные о счете.
Банкомат выводит меню счетов и команд.

*Обработка
транзакции*

Банкомат выводит меню счетов и команд.
Пользователь выбирает снятие денег со счета.
Банкомат запрашивает снимаемую сумму.
Пользователь вводит \$100.
Банкомат проверяет сумму на превышение лимита выдачи наличных денег.
Банкомат связывается с консорциумом и банком для проверки наличия достаточной суммы на счете.
Банкомат выдает наличные деньги и просит пользователя забрать их.
Пользователь берет наличные деньги.
Банкомат выводит меню счетов и команд.

*Передача
данных*

Банкомат запрашивает данные о счете, связываясь с консорциумом.
Консорциум принимает запрос и направляет его в соответствующий банк.
Банк принимает запрос и выдает требуемую информацию.
Банк отправляет информацию в консорциум.
Консорциум направляет информацию банкомату.

Нетипичные сценарии и исключительные ситуации

- После разработки типовых сценариев необходимо рассмотреть особые ситуации, такие как отсутствие введенных значений, ввод одинаковых значений и т. д. Затем нужно рассмотреть ошибочные ситуации, включая ввод неправильных значений и отсутствие отклика.
- Для многих интерактивных приложений обработка ошибок является наиболее сложной частью процесса разработки. Необходимо предоставлять пользователю возможность отменить операцию или откатиться к четко определенной начальной точке каждого этапа.

Внешние события

- Проанализируйте все разработанные сценарии и выделите все внешние события: ввод данных, принятие решений, прерывания и взаимодействия с другими пользователями и внешними устройствами.
- При помощи сценариев вы можете отыскать типовые сценарии, но не забывайте и про нетипичные события и ошибочные ситуации.

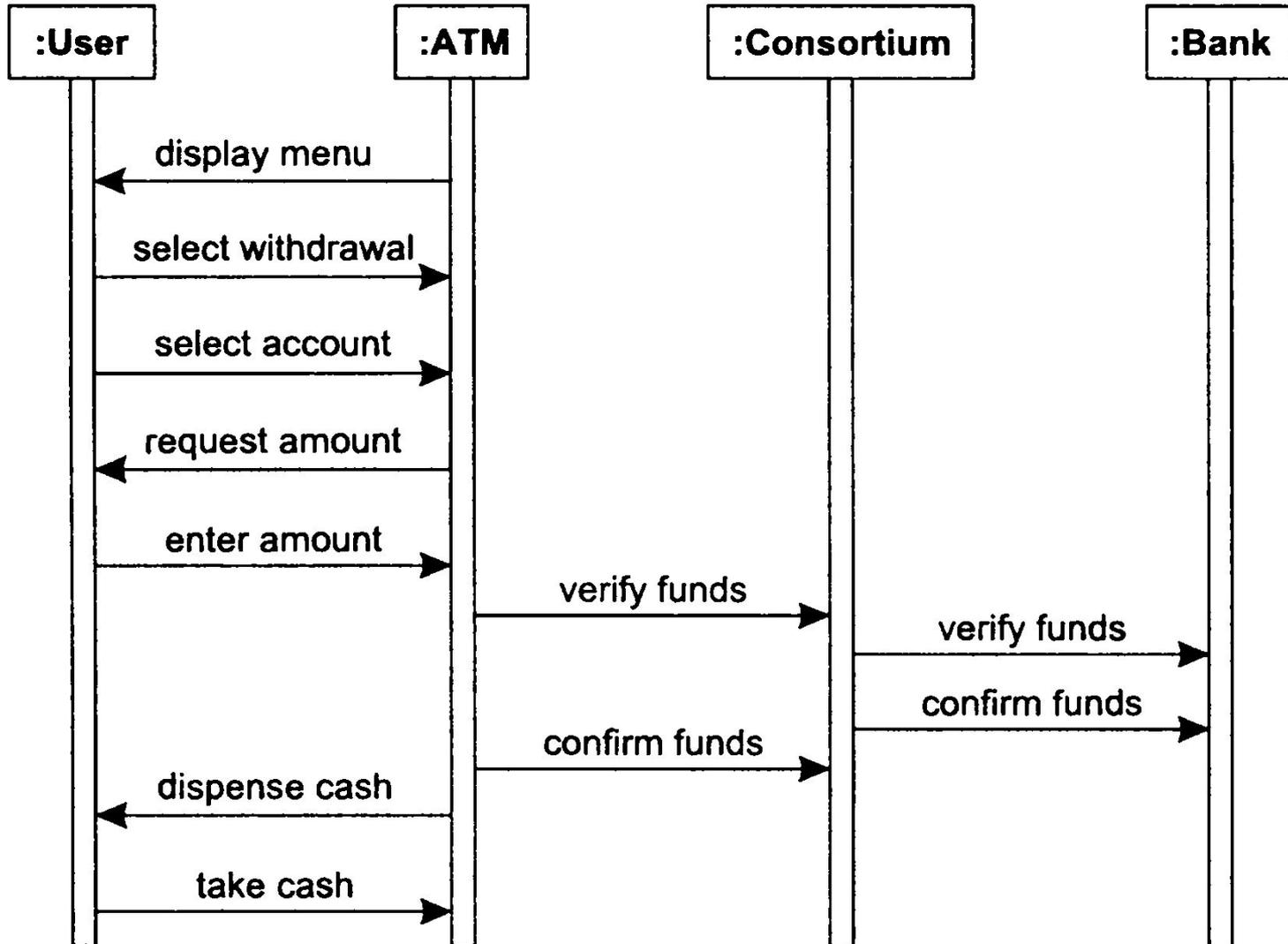
Внешние события

- Передача информации объекту является событием. Например, «введен пароль» - это сообщение, переданное от внешнего агента User (Пользователь) объекту приложения АТМ (Банкомат).
- Сгруппируйте под одинаковыми названием события, оказывающие одинаковое влияние на поток управления, даже если значения их параметров отличаются. Выбор значения пароля не влияет на поток управления, поэтому события в разными паролями являются экземплярами одного и того же типа событий. Аналогичным образом «выдача наличных» также является событием одного и того же типа независимо от суммы.
- Экземпляры событий, значения которых влияют на поток управления, должны быть отнесены к разным типам событий. «Правильный счет», «неверный счет» и «неверный пароль» - разные события, которые не следует группировать под названием «состояние карты»

Внешние события

- Подготовьте диаграмму последовательности для каждого сценария. Диаграмма последовательности показывает участников взаимодействия и последовательность сообщений, которыми они обмениваются. Для каждого участника выделяется свой столбец. Диаграмма показывает отправителя и получателя каждого сообщения. Если в объекте участвует несколько объектов одного и того же класса, им следует разные номера.
- Изучив один столбец таблицы, вы можете определить события, непосредственно влияющие на конкретный объект. После этого вы можете сгруппировать события, отправляемые и принимаемые каждым классом.

Диаграмма последовательности



Группировка событий

