

4. Моделирование функциональных требований к системе

4.1. Функциональная модель системы

- **Функциональная (use case) модель системы** описывает систему с точки зрения функциональных требований, которые пользователи требуют от системы.
- Акцент ставится на то, что система **должна** делать, а не то, как она это делает.

Структурный аспект функциональной модели системы

- Статический взгляд на функциональную модель системы описывается с помощью **диаграмм вариантов использования** (use case diagrams).
- Диаграмма вариантов использования может рассматриваться как **набор действий**, исполняемых системой и которые разбивают систему на **дискретные части**.

Динамический аспект функциональной модели системы

- Динамический аспект функциональной модели системы представляет собой описание:
 - взаимодействия пользователя с системой;
 - алгоритмов исполнения вариантов использования.
- Это описание может быть как словесным, так и графическим, используя диаграммы поведения.

- **Сценарием** называется последовательность действий, выполняемых пользователем и системой при их взаимодействии.
- На диаграмме вариантов графически отображаются действия одного или нескольких взаимосвязанных сценариев.
- Диаграмма вариантов использования **не описывает** последовательность этих действий, а только перечисляет их и указывает инициаторов этих действий.

4.2. Элементы диаграммы вариантов использования

Основные элементы

- Диаграмма вариантов использования содержит следующие основные элементы:
 - **актеры (actors);**
 - **варианты использования (use cases);**
 - **отношения между актерами и вариантами использования.**

Актёр

- **Актёр** обозначает роль, которую играет объект, взаимодействующий с системой.
- Этот объект может представлять любую сущность, например, пользователя системы или другую систему.

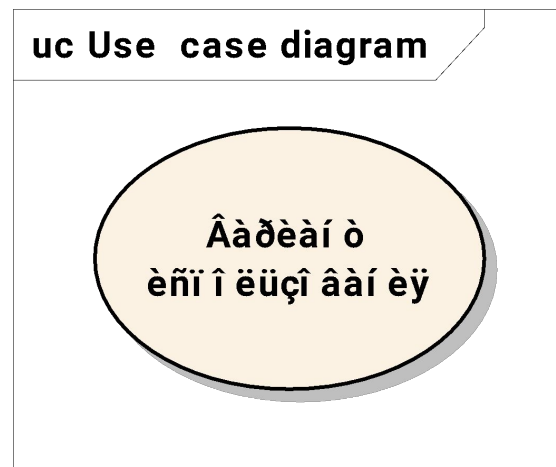
- Графическое обозначение актёра:



Вариант использования

- **Вариант использования** – это функциональное требование к системе с точки зрения пользователя этой системы.
- Вариант использования представляет некоторое действие или последовательность действий, которые исполняются системой.

- Графическое обозначение варианта использования:



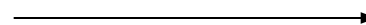
Отношения между актерами и вариантами использования

- Существует три типа отношений между актерами и вариантами использования:

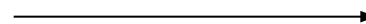
- **association** - ассоциация;
- **dependency** – зависимость;
- **generalization** – обобщение.

- Графическое обозначение отношений:

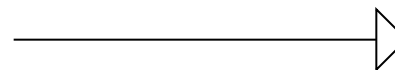
- ассоциация:



- зависимость:



- обобщение:

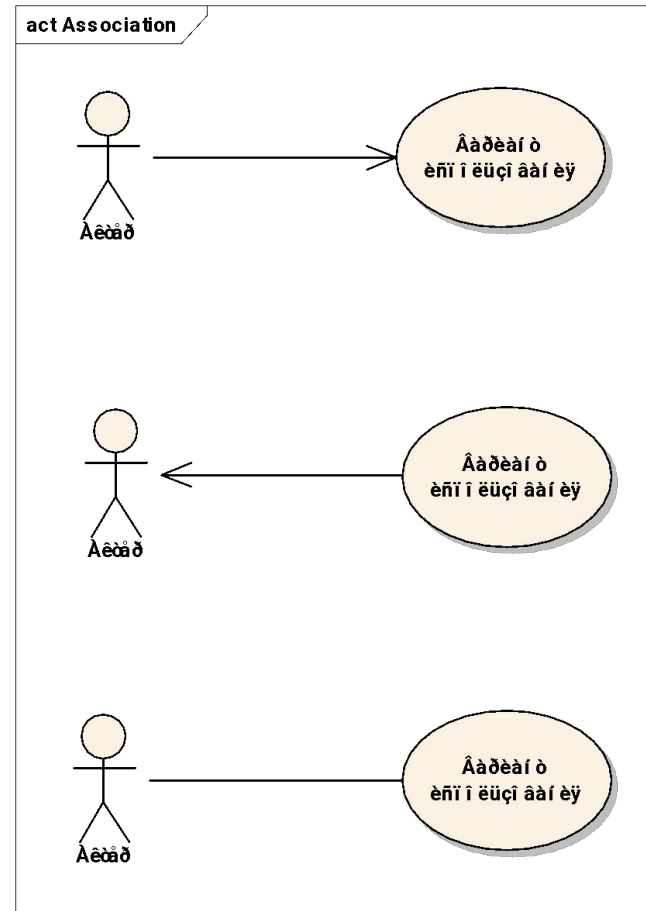


Отношение ассоциации

- **Ассоциация** (association) показывает следующие взаимодействие между актером и вариантом использования:
 - - актер инициирует взаимодействие с вариантом использования;
 - - система инициирует взаимодействие с актером;
 - - взаимодействие актера и системы.

Графическое обозначение взаимодействия актера и варианта использования

- - актер инициирует взаимодействие с вариантом использования;
- - система инициирует взаимодействие с актером;
- - взаимодействие актера и системы.



Отношение зависимости

- **Зависимость** (dependency) указывает на некоторую зависимость между вариантами использования или актерами, т.е. показывает, что изменение одного элемента модели воздействует (вызывает изменение) на другие элементы модели, зависящие от этого элемента.

Стереотипы отношения зависимости

- Чаще всего используют следующие три типа зависимости:
- <<***extend***>> - один элемент расширяет функциональность другого элемента;
- <<***include***>> - один элемент включает функциональность других элементов;
- <<***uses***>> - один элемент использует функциональность других элементов.

Примеры отношения
зависимости со стереотипом
<<extend>>

Пример отношения зависимости
со стереотипом <<include>>

Пример отношения зависимости
со стереотипом <<use>

Отношение обобщения

- **Обобщение**
(generalization)
показывает, что один элемент обобщает другой элемент относительно некоторой классификации.
- Пример отношения обобщения:

Другие элементы диаграммы вариантов использования

- ***Сотрудничество*** (collaborations) – определяет множество взаимодействующих объектов, исполняющих специфическую функциональность.
- Графическое обозначение сотрудничества:

- ***Граница системы***
(boundary) – включает варианты использования, которые составляют функционал системы или подсистемы.
- Графическое обозначение границы системы:

- **Примечание** (note) – используется для пояснения и комментариев к какому-либо элементу диаграммы.
- **Связь** примечания с элементом, к которому относится это примечание (anchor note to item).
- **Пакет** (package) - здесь это набор вариантов использования.

- графическое обозначение примечания и его связи с элементом диаграммы, который оно поясняет:

4.3. Варианты использования

- Вариант использования описывает **действие**, выполняемое системой, с точки зрения **пользователя** этой системы, например:
 - сделать оплату;
 - перевести средства со счета на счет.
- Вариант использования **не должен** описывать:
 - интерфейс с пользователем;
 - архитектуру системы;
 - не функциональные требования к системе, например, производительность, надежность.

Рецепты нахождения варианта использования

- Для нахождения вариантов использования нужно изучить:
 - **функции**, которые пользователь требует от системы;
 - **операции** типа create, read, write, update, delete, которые изменяют информацию, хранимую в системе;
 - описания того, как актер **информируется** об изменении состояния системы.

4.4. Функциональные уровни вариантов использования

- Функционально система может быть разделена на следующие уровни:
 - **component level** – диаграммы вариантов использования этого уровня описывают взаимодействие актеров с компонентами системы;
 - **application service level** – диаграммы вариантов использования этого уровня описывают взаимодействие актеров с сервисом (функциональной частью) системы;
 - **organization level** – диаграммы вариантов такого уровня описывают взаимодействие актеров со всей системой.

4.5. Актеры

- **Актер** это сущность, которая взаимодействует с системой и может быть как человеком, так и другой системой.

Классификация актеров

- Актеров, взаимодействующих с системой, делят на три группы:
 - **основные актеры** (primary actors) – это актеры, которые являются пользователями программной системы и вызывают её реакцию;
 - **вспомогательные актеры** (supporting actors) – это актеры, которые обслуживают систему;
 - **закулисные актеры** (offstage actors) – это актеры, которые связаны с исполнением варианта использования, но не являются основными или вспомогательными актерами.

Общий порядок нахождения актеров

1. Сначала рассмотрите пользователей системы;
2. Сгруппируйте пользователей по интересам (выполняемым задачам);
3. Дайте выделенным группам пользователей имена и определите их роли;
4. Определите системы, которые взаимодействуют с вашей системой;
5. Определите закулисных актеров.

Правила именования актеров

- Назовите актеров пользователей их ролями.
- Не называйте актера должностью, которую он занимает.
- Не тратьте время на споры об имени актера;
- Назовите вспомогательного актера именем системы, которую он представляют;
- Назовите закулисного актера именем процесса, который он исполняет.

4.6. Глоссарий

- **Глоссарий** – это словарь специфических терминов, использующихся в модели системы.
- **Цель глоссария** дать пояснения всем этим терминам.
- Глоссарий должен сопровождать функциональную модель системы и другую документацию на систему.

Информация, которую должен содержать глоссарий

- Определение ключевых концепций.
- Пояснение двусмысленных терминов.
- Объяснение жаргона.
- Определения бизнес событий (business events).
- Объяснение действий программного обеспечения.

4.7. Ошибки, допускаемые при разработке диаграммы вариантов использования

- Игнорируются требования к оформлению документации.
- Нет ясной цели при разработке диаграммы вариантов использования.
- Варианты использования моделируют систему на разных функциональных уровнях.
- В диаграмму вариантов использования включены нефункциональные требования и детали пользовательского интерфейса.

- В диаграмме вариантов использования часто используются стереотипы «includes» и «extends».
- Не обращают внимания на определение бизнес правил (business rules).
- При разработке диаграммы вариантов использования не консультируются с экспертами в прикладной области.
- При разработке диаграммы вариантов использования не консультируются с пользователями.
- Пытаются разработать диаграмму вариантов использования с первого раза. Необходимы улучшения.
- Не проверяют правильность разработанной диаграммы вариантов использования.