В нотации языка UML определены следующие виды канонических диаграмм (слайд 2):

* вариантов использования (use case diagram)
* классов (class diagram)
* кооперации (collaboration diagram)
* последовательности (sequence diagram)
* состояний (statechart diagram)
* деятельности (activity diagram)
* компонентов (component diagram)
* развертывания (deployment diagram)

Перечень этих диаграмм и их названия являются каноническими в том смысле, что представляют собой неотъемлемую часть графической нотации языка UML. Более того, процесс ООАП неразрывно связан с процессом построения этих диаграмм. При этом совокупность построенных таким образом диаграмм является самодостаточной в том смысле, что в них содержится вся информация, которая необходима для реализации проекта сложной системы.

Каждая из этих диаграмм детализирует и конкретизирует различные представления о модели сложной системы в терминах языка UML. При этом диаграмма вариантов использования представляет собой наиболее общую концептуальную модель сложной системы, которая является исходной для построения всех остальных диаграмм. Диаграмма классов, по своей сути, логическая модель, отражающая статические аспекты структурного построения сложной системы.

Диаграммы кооперации и последовательностей представляют собой разновидности логической модели, которые отражают динамические аспекты функционирования сложной системы. Диаграммы состояний и деятельности предназначены для моделирования поведения системы. И, наконец, диаграммы компонентов и развертывания служат для представления физических компонентов сложной системы и поэтому относятся к ее физической модели.

В целом интегрированная модель сложной системы в нотации UML может быть представлена в виде совокупности указанных выше диаграмм (слайд 3):

Кроме графических элементов, которые определены для каждой канонической диаграммы, на них может быть изображена текстовая информация, которая расширяет семантику базовых элементов. В языке UML предусмотрены три специальных механизма расширения, которые включают в себя следующие конструкции.

Стереотип (stereotype) — новый тип элемента модели, который расширяет семантику метамодели. Стереотипы должны основываться на уже существующих и описанных в метамодели языка UML типах или классах.

Стереотипы предназначены для расширения именно семантики, но не структуры уже описанных типов или классов. Некоторые стереотипы предопределены в языке UML, другие могут быть указаны разработчиком. На диаграммах изображаются в форме текста, заключенного в угловые кавычки. Предопределенные стереотипы являются ключевыми словами языка UML, которые используются на канонических диаграммах на языке оригинала без их перевода.

Помеченное значение (tagged value) — явное определение свойства как пары "имя – значение". В помеченном значении само имя называют тегом (tag).

Помеченные значения на диаграммах изображаются в форме строки текста специального формата, заключенного в фигурные скобки. При этом используется следующий формат записи: {тег = значение}. Теги встречаются в нотации языка UML, но их определение не является строгим, поэтому теги могут быть указаны самим разработчиком.

Ограничение (constraint) — некоторое логическое условие, ограничивающее семантику выбранного элемента модели.

Как правило, все ограничения специфицируются разработчиком. Ограничения на диаграммах изображаются в форме строки текста, заключенного в фигурные скобки. Для формальной записи ограничений предназначен специальный язык объектных ограничений (Object Constraint Language, OCL), который является составной частью языка UML.

Визуальное моделирование с использованием нотации UML можно представить как процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной бизнес-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 бизнес-система должна делать в процессе своего функционирования.

**Диаграмма вариантов использования (use case diagram) — диаграмма, на которой изображаются отношения между актерами и вариантами использования. (слайд 6)**

Диаграмма вариантов использования - это исходное концептуальное представление или концептуальная модель системы в процессе ее проектирования и разработки. Создание диаграммы вариантов использования имеет следующие 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

Назначение данной диаграммы состоит в следующем: проектируемая программная система представляется в форме так называемых вариантов использования, с которыми взаимодействуют внешние сущности или актеры. При этом актером или действующим лицом называется любой объект, субъект или система, взаимодействующая с моделируемой бизнес-системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая служит источником воздействия на моделируемую систему так, как определит разработчик. Вариант использования служит для описания сервисов, которые система предоставляет актеру. Другими словами каждый вариант использования определяет набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой и собственно выполнение вариантов использования.

Рассматривая диаграмму вариантов использования в качестве модели бизнес-системы, можно ассоциировать ее с "черным ящиком". Концептуальный характер этой диаграммы проявляется в том, что подробная детализация диаграммы или включение в нее элементов физического уровня представления на начальном этапе проектирования скорее имеет отрицательный характер, поскольку предопределяет способы реализации поведения системы. Эти аспекты должны быть сознательно скрыты от разработчика на диаграмме вариантов использования.

Базовыми элементами диаграммы вариантов использования являются вариант использования и актер (слайд 6).

**Вариант использования (use case)** — внешняя спецификация последовательности действий, которые система или другая сущность могут выполнять в процессе взаимодействия с актерами.

Несмотря на то, что каждый вариант использования определяет последовательность действий, которые должны быть выполнены проектируемой системой при взаимодействии ее с соответствующим актером, сами эти действия не изображаются на рассматриваемой диаграмме.

Цель спецификации варианта использования заключается в том, чтобы зафиксировать некоторый аспект или фрагмент поведения проектируемой системы без указания особенностей реализации данной функциональности. В этом смысле каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемая система по запросу актера, т. е. определяет один из способов применения системы. Сервис, который инициализируется по запросу актера, должен представлять собой законченную последовательность действий. Это означает, что после того как система закончит обработку запроса актера, она должна возвратиться в исходное состояние, в котором снова готова к выполнению следующих запросов.

Диаграмма вариантов использования содержит конечное множество вариантов использования, которые в целом должны определять все возможные стороны ожидаемого поведения системы. Для удобства множество вариантов использования может рассматриваться как отдельный пакет. Применение вариантов использования на всех этапах работы над проектом 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 их итеративного обсуждения со всеми заинтересованными специалистами.

**Актер (actor)** — согласованное множество ролей, которые играют внешние сущности по отношению к вариантам использования при взаимодействии с ними.

В некоторых случаях актер может обозначаться в виде прямоугольника класса со стереотипом <<actor>> и обычными составляющими элементами класса. Имена актеров должны начинаться с заглавной буквы и следовать рекомендациям использования имен для типов и классов модели. При этом символ отдельного актера связывает соответствующее описание актера с конкретным именем.

Имя актера должно быть достаточно информативным с точки зрения семантики. Для этой цели подходят наименования должностей в компании (например, продавец, кассир, менеджер, президент). Не рекомендуется давать актерам имена собственные или названия моделей конкретных устройств, 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.

Актеры взаимодействуют с системой посредством передачи и приема сообщений от вариантов использования. Сообщение представляет собой запрос актером сервиса от 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.

**Отношения на диаграмме вариантов использования (слайд 7):**

Между элементами 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.

В то же время два варианта использования, определенные в рамках одной моделируемой системы, также могут взаимодействовать друг с другом, однако характер этого взаимодействия будет отличаться от взаимодействия с актерами. Однако в обоих случаях способы взаимодействия элементов модели предполагают обмен сигналами или сообщениями, которые инициируют реализацию функционального поведения моделируемой системы.

***Отношение ассоциации (слайд 8):***

Отношение ассоциации – одно 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 ассоциация служит для обозначения специфической роли актера при его взаимодействии с отдельным вариантом использования. Другими словами, ассоциация специфицирует семантические особенности взаимодействия актеров и вариантов использования в графической модели системы. 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 некоторые дополнительные обозначения, например, имя и кратность.

***Включение (*include*)*** (слайд 9) в языке UML — это разновидность отношения зависимости между базовым вариантом использования и его специальным случаем. При этом отношением зависимости (dependency) является такое отношение между двумя элементами модели, при котором изменение одного элемента (независимого) приводит к изменению другого элемента (зависимого).

Отношение включения устанавливается только между двумя вариантами использования и указывает на то, что заданное поведение для одного варианта использования включается в качестве составного фрагмента 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.

***Отношение расширения*** (слайд 10) **(extend)** определяет взаимосвязь базового варианта использования с другим вариантом использования, функциональное поведение которого задействуется базовым не всегда, а только при выполнении дополнительных условий.

В языке UML отношение расширения является зависимостью, направленной к базовому варианту использования и соединенной с ним в так называемой точке расширения. Отношение расширения между вариантами использования обозначается как отношение зависимости в форме пунктирной линии со стрелкой, направленной от того варианта использования, который является расширением для базового варианта использования. Данная линия со стрелкой должна быть помечена стереотипом <<extend>>.

Семантика отношения расширения определяется следующим образом. Если базовый вариант 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 базового варианта, то проверяется логическое условие данного отношения. Если это условие выполняется, исходная последовательность действий расширяется посредством включения действий 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.

Два и более актера 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 отношения обобщения с другим, возможно, абстрактным актером, который моделирует соответствующую общность ролей.

Язык UML включает в себя специальные механизмы расширения, которые позволяют ввести в рассмотрение дополнительные графические обозначения, ориентированные для решения задач из определенной предметной области. Примеры подобных обозначений, которые используются для моделирования бизнес-систем и могут быть изображены на диаграммах вариантов использования: бизнес-актер, сотрудник и бизнес - вариант использования. (слайд 12)

Бизнес-актер (business actor) – индивидуум, группа, организация, компания или система, которые взаимодействуют с моделируемой бизнес-системой, но не входят в нее, т.е. не являются частью моделируемой системы.

Сотрудник (business worker) – индивидуум, который действует внутри моделируемой бизнес-системы, взаимодействует с другими сотрудниками и является участником бизнес-процесса моделируемой системы.

Бизнес-вариант использования (business use case) — вариант использования, определяющий последовательность действий моделируемой системы, направленных на выполнение отдельного бизнес-процесса.