Анализ и проектирование на UML

Направления подготовки «Информационные системы и технологии»

Максим Валерьевич Хлопотов, старший преподаватель кафедры ИС

UML — это язык моделирования

- UML имеет отношение прежде всего и главным образом к созданию и применению компьютерных программ.
- В отношении разработки программного обеспечения так сложилось, что результаты фаз анализа и проектирования, оформленные средствами определенного языка, принято называть моделью.

UML — это язык моделирования

- Деятельность по составлению моделей естественно назвать моделированием.
 Именно в этом смысле UML является языком моделирования.
- Модель UML это, прежде всего, основной артефакт фазы проектирования программной системы.

UML — это графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех артефактов, создаваемых при разработке программных систем.

Основное назначение <u>UML</u> предоставить, с одной стороны, достаточно формальное, с другой стороны, достаточно удобное, и, с третьей стороны, достаточно универсальное средство, позволяющее до некоторой степени снизить риск расхождений в толковании спецификаций.

Второе по важности назначение UML состоит в том, чтобы служить адекватным средством коммуникации между людьми.

Разумеется, наглядность визуализации моделей UML имеет значение, только если они должны составляться или восприниматься человеком — это назначение UML не имеет отношения к компьютерам.

UML предназначен не только для описания абстрактных моделей приложений, но и для непосредственного манипулирования артефактами, входящими в состав этих приложений, в том числе такими, как программный код.

Наконец, четвёртое назначение – документирование.

Модели UML являются документами, которые можно использовать самыми разными способами, начиная с печати картинок и заканчивая автоматической генерацией человекочитаемых текстовых описаний. В последних версиях UML с целью достижения более полного соответствия этому назначению сделано довольно много.

Стандарт UML

Чтобы подчеркнуть, что UML язык графический, авторы называют правила записи (рисования) моделей не синтаксисом, а *нотацией*.

Типов элементов нотации четыре:

- фигуры;
- линии;
- значки;
- тексты.

Модель UML

Модель UML — это конечное множество сущностей и отношений между ними.

Рассматривая модель UML с наиболее общих позиций, можно сказать, что это граф (точнее, нагруженный мульти-псевдо-гиперорграф), в котором вершины и ребра нагружены дополнительной информацией и могут иметь сложную внутреннюю структуру. Вершины этого графа называются сущностями, а ребра — отношениями.

Сущности

Для удобства обзора сущности в UML можно подразделить на четыре группы:

- структурные;
- поведенческие;
- группирующие;
- аннотационные.

Структурные сущности

- Класс описание множества объектов с общими атрибутами и операциями.
- Интерфейс множество операций, которое определяет набор услуг (службу), предоставляемых классом или компонентом.
- Действующее лицо сущность, находящаяся вне моделируемой системы и непосредственно взаимодействующая с ней.
- Вариант использования описание последовательности производимых системой действий, доставляющей значимый для некоторого действующего лица результат.
- Компонент физически заменяемый артефакт, реализующий некоторый набор интерфейсов.
- Узел физический вычислительный ресурс.

Отношения

В UML используются четыре основных типа отношений:

- зависимость;
- ассоциация;
- обобщение;
- реализация.

Отношения

Зависимость — это наиболее общий тип отношения между двумя сущностями. Отношение зависимости указывает на то, что изменение независимой сущности каким-то образом влияет на зависимую сущность. Графически отношение зависимости изображается в виде пунктирной стрелки, направленной от независимой сущности к зависимой.

Ассоциация — это наиболее часто используемый тип отношения между сущностями. Отношение ассоциации имеет место, если одна сущность непосредственно связана с другой (или с другими — ассоциация может быть не только бинарной). Графически ассоциация изображается в виде сплошной линии с различными дополнениями, соединяющей связанные сущности.

Отношения

Обобщение — это отношение между двумя сущностями, одна их которых является частным (специализированным) случаем другой. Графически обобщение изображается в виде сплошной стрелки с треугольником на конце, направленной от частного к общему. Отношение наследования между классами в объектно-ориентированных языках программирования является типичным примером обобщения.

Отношение *реализации* указывает, что одна сущность является реализацией другой. Например, класс является реализацией интерфейса. Графически реализация изображается в виде пунктирной стрелки с треугольником на конце, направленной от реализующей сущности к реализуемой.

Диаграммы UML

Диаграммы UML – основная накладываемая на модель структура, которая облегчает создание и использование модели.

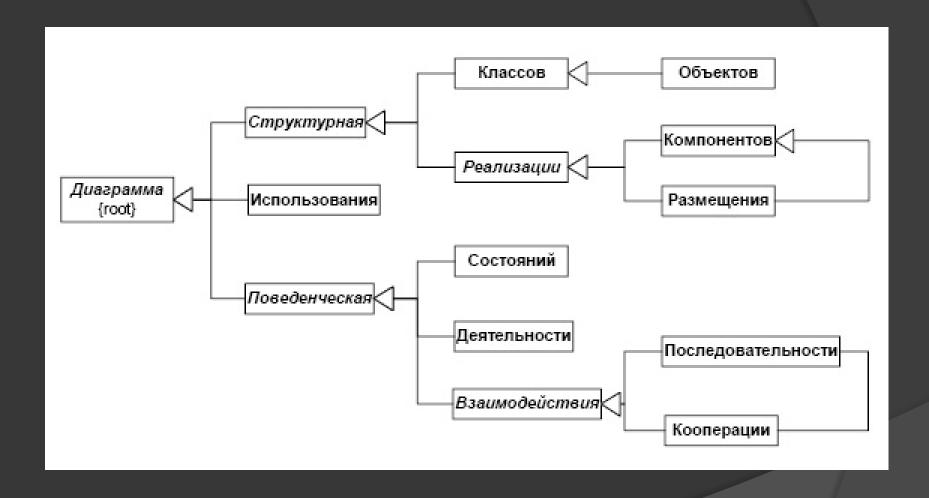
Диаграмма — это графическое представление некоторой части графа модели. Авторы UML определили набор рекомендуемых к использованию типов диаграмм, которые получили название канонических типов диаграмм.

Диаграммы UML

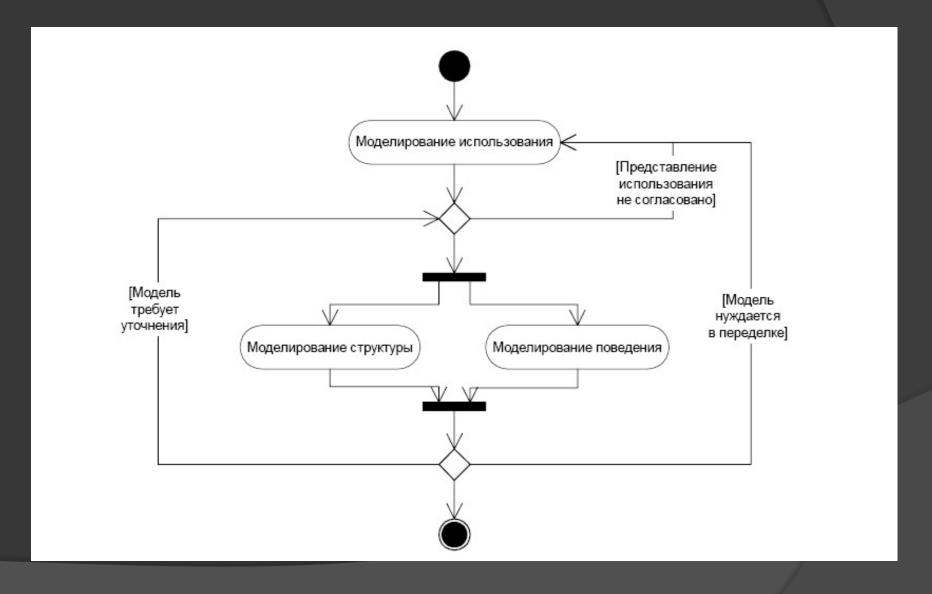
В UML 1.х всего определено 9 канонических типов диаграмм.

- Диаграмма использования
- Диаграмма классов
- Диаграмма объектов
- Диаграмма состояний
- Диаграмма деятельности
- Диаграмма последовательности
- Диаграмма кооперации
- Диаграмма компонентов
- Диаграмма размещения

Иерархия диаграмм UML



Процесс моделирования



Представления

Все аспекты моделируемой системы не удается описать с единой точки зрения. Моделировать сложную систему следует с нескольких различных точек зрения, каждый раз принимая во внимание один аспект

моделируемой системы и абстрагируясь от остальных.

Этот тезис является одним из основополагающих принципов UML.

Представления

Выделим три представления:

- представление использования (что делает система полезного?);
- представление структуры (из чего состоит система?);
- представление поведения (как работает система?).

Представления

Представление использования призвано отвечать на вопрос, что делает система полезного.

Определяющим признаком для отнесения элементов модели к представлению использования является, по нашему мнению, явное сосредоточение внимание на факте наличия у системы внешних границ, то есть выделение внешних действующих лиц, взаимодействующих с системой, и внутренних вариантов использования, описывающих различные сценарии такого взаимодействия.

Описывается диаграммой использования.

Диаграмма использования

Диаграмма использования является основным средством моделирования использования в UML.

На диаграмме использования применяются следующие типы сущностей:

- действующие лица;
- варианты использования;
- примечания;
- пакеты.

Диаграмма использования

Между этими сущностями устанавливаются следующие типы отношений:

- ассоциация между действующим лицом и вариантом использования;
- обобщение между действующими лицами;
- обобщение между вариантами использования;
- зависимости (двух стереотипов) между вариантами использования.

Моделирование использования

Наш язык и мышление устроены так, что самой простой, понятной и четкой формой изложения мыслей являются так называемые простые утверждения.

Простое утверждение имеет следующую грамматическую форму: подлежащее — сказуемое — прямое дополнение. В логических терминах: субъект — предикат — объект.

Например: начальник увольняет сотрудника, директор создает отдел.

Моделирование использования

По сути, именно простые утверждения и записаны на диаграмме использования. Действующее лицо — это субъект, а вариант использования — предикат (вместе с объектом). Моделирование использования предполагает явное формулирование требований к системе на самом начальном этапе разработки.

Диаграмма использования

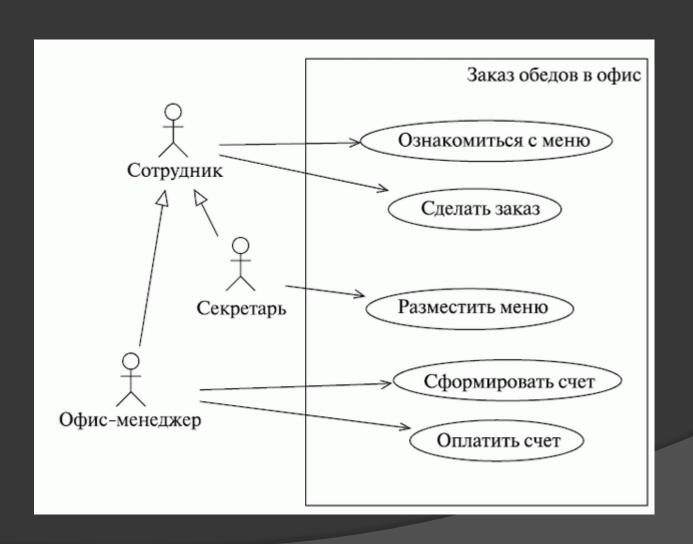
С синтаксической точки зрения действующее лицо — это стереотип классификатора, который обозначается специальным значком. Для действующего лица указывается только имя, идентифицирующее его в системе. Семантически действующее лицо — это множество логически взаимосвязанных ролей. С прагматической точки зрения главным является то, что действующие лица находятся вне проектируемой системы (или рассматриваемой части системы).

Действующие лица

В качестве имен действующих лиц рекомендуется использовать существительное (возможно с определяющим словом), а в качестве имен вариантов использования — глагол (возможно, с дополнением).



Пример нотации



Варианты использования

Семантически вариант использования — это описание множества возможных последовательностей действий (событий), приводящих к значимому для действующего лица результату.

Прагматика варианта использования состоит в том, что среди всех последовательностей действий, могущих произойти при работе приложения, выделяются такие, в результате которых получается явно видимый и достаточно важный для действующего лица результат.

Варианты использования

Выбор вариантов использования сильно влияет на качество модели. Формальные методы выбора предложить трудно — помогают только опыт и чутьё.

Некоторые пункты Т3 естественным образом переводятся в варианты использования.

Ассоциация

Ассоциация между действующим лицом и вариантом использования показывает, что действующее лицо тем или иным способом взаимодействует (предоставляет исходные данные, потребляет результат) с вариантом использования.

Ассоциация обозначает, что действующее лицо так или иначе, но обязательно непосредственно участвует в выполнении каждого из сценариев, описываемых вариантом использования.

Обобщение

Обобщение между действующими лицами показывает, что одно действующее лицо наследует все свойства (в частности, участие в ассоциациях) другого действующего лица.

С помощью обобщения между действующими лицами легко показать иерархию категорий пользователей системы, в частности, иерархию прав доступа к выполняемым функциям и хранимым данным.

Обобщение

Обобщение между вариантами использования показывает, что один вариант использования является частным случаем (подмножеством множества сценариев) другого варианта использования.

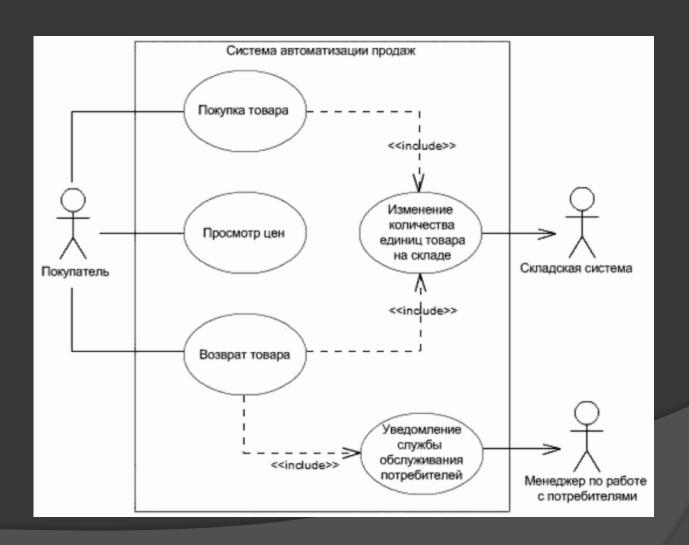
Зависимости

Зависимость между вариантами использования показывает, что один вариант использования зависит от другого варианта использования.

- 2 стандартных стереотипа зависимости:
- include показывает, что сценарий независимого варианта использования включает в себя в качестве подпоследовательности действий сценарий зависимого варианта использования;
- extend показывает, что в сценарий зависимого варианта использования может быть в определенном месте вставлен в качестве подпоследовательности действий сценарий независимого варианта использования.



Пример





Пример



Реализация вариантов использования

После того, как построено представление использования, то есть выделены действующие лица, варианты использования и установлены отношения между ними, встает вопрос: что дальше?

Представление использования, если оно тщательно продумано и детально прорисовано, является формой технического задания, содержащей достаточно информации для дальнейшего проектирования.

Реализация вариантов использования

Действующие лица находятся вне системы — с ними ничего делать не нужно.

Таким образом, переход от моделирования использования к другим видам моделирования состоит в уточнении, детализации и конкретизации вариантов использования.

В представлении использования мы показали, что делает система, теперь нужно определить, как это делается. Это обычно называется реализацией вариантов использования.

Диаграмма деятельности

- Реализация варианта использования
 диаграммой деятельности является
 компромиссным способом ведения разработки
 в сущности, это проектирование сверху вниз в
 терминах и обозначениях UML.
- Эту диаграмму можно показать заказчику, чтобы проверить, действительно ли проектируемая нами логика работы системы соответствует тому бизнес-процессу, который существует в реальности.

Диаграмма деятельности

Применение диаграмм деятельности для реализации вариантов использования не слишком приближает к появлению целевого артефакта — программного кода, однако может привести к более глубокому пониманию существа задачи и даже открыть неожиданные возможности улучшения приложения, которые было трудно усмотреть в первоначальной постановке задачи.

• А теперь тест. Ура!