

ЛЕКЦИЯ 6

История происхождения и развития языка UML
Диаграмма вариантов использования

Методологии, положенные в основу языка UML

- метод Гради Буча, получивший условное название Booch или Booch'91, Booch Lite (позже Booch'93)
- метод Джима Румбаха, получивший название Object Modeling Technique – OMT (позже OMT-2)
- метод Айвара Джекобсона, получивший название Object Oriented Software Engineering – OOSE

Требования к языку моделирования

- позволять моделировать не только программное обеспечение, но и более широкие классы систем и бизнес-приложений, с использованием объектно-ориентированных понятий;
- явным образом обеспечивать взаимосвязь между базовыми понятиями моделей концептуального, логического и физического уровней;
- обеспечивать масштабируемость моделей, что является важной особенностью сложных многоцелевых систем;
- быть понятным аналитикам, программистам и другим специалистам, входящим в состав проектных команд;
- поддерживаться специальными инструментальными средствами, реализованными на различных компьютерных платформах.

Основные этапы развития UML

Октябрь 1995 г.	Унифицированный метод 0.8
Июнь-октябрь 1996 г.	UML 0.9/0.91
Январь 1997 г.	UML 1.0
Ноябрь 1997 г.	UML 1.1 (поддержка OMG)
Июнь 1999 г.	UML 1.3
Сентябрь 2001 г.	UML 1.4
Март 2003 г.	UML 1.5
Октябрь 2004 г.	UML 2.0 (финальная спецификация)
Август 2005 г.	UML 2.0 (официальный стандарт)
2006 – 2007 гг.	UML 2.1
Февраль 2009 г.	UML 2.2
Май 2010 г.	UML 2.3
Март 2011 г.	UML 2.4
Июнь 2015 г.	UML 2.5

Назначение языка UML 2

- Предоставить в распоряжение всех пользователей легко воспринимаемую и выразительную нотацию для визуального моделирования, специально предназначенную для разработки и документирования моделей систем самого различного назначения.
- Снабдить исходные понятия языка UML 2 возможностью расширения и специализации для более точного представления моделей систем в конкретной предметной области.
- Описание языка UML 2 должно поддерживать такую спецификацию моделей, которая не зависит от конкретных языков программирования и инструментальных средств проектирования программных систем.
- Описание языка UML 2 должно включать в себя семантический базис для понимания общих особенностей ООАП.

Назначение языка UML 2

- Поощрять развитие рынка программных инструментальных средств, поддерживающих методологию ООАП и MDA.
- Способствовать распространению объектных технологий и соответствующих понятий ООАП.
- Интегрировать в себя новейшие и наилучшие достижения практики ООАП.

Общая структура UML 2

Семантика – система правил и соглашений, определяющих толкование и придание смысла конструкциям некоторого языка.

Нотация – система условных обозначений, принятая в некотором языке для изображения и визуализации модели.

Структурные модели – модели, предназначенные для описания статической структуры сущностей или элементов некоторой системы, включая их классы, интерфейсы, атрибуты и отношения.

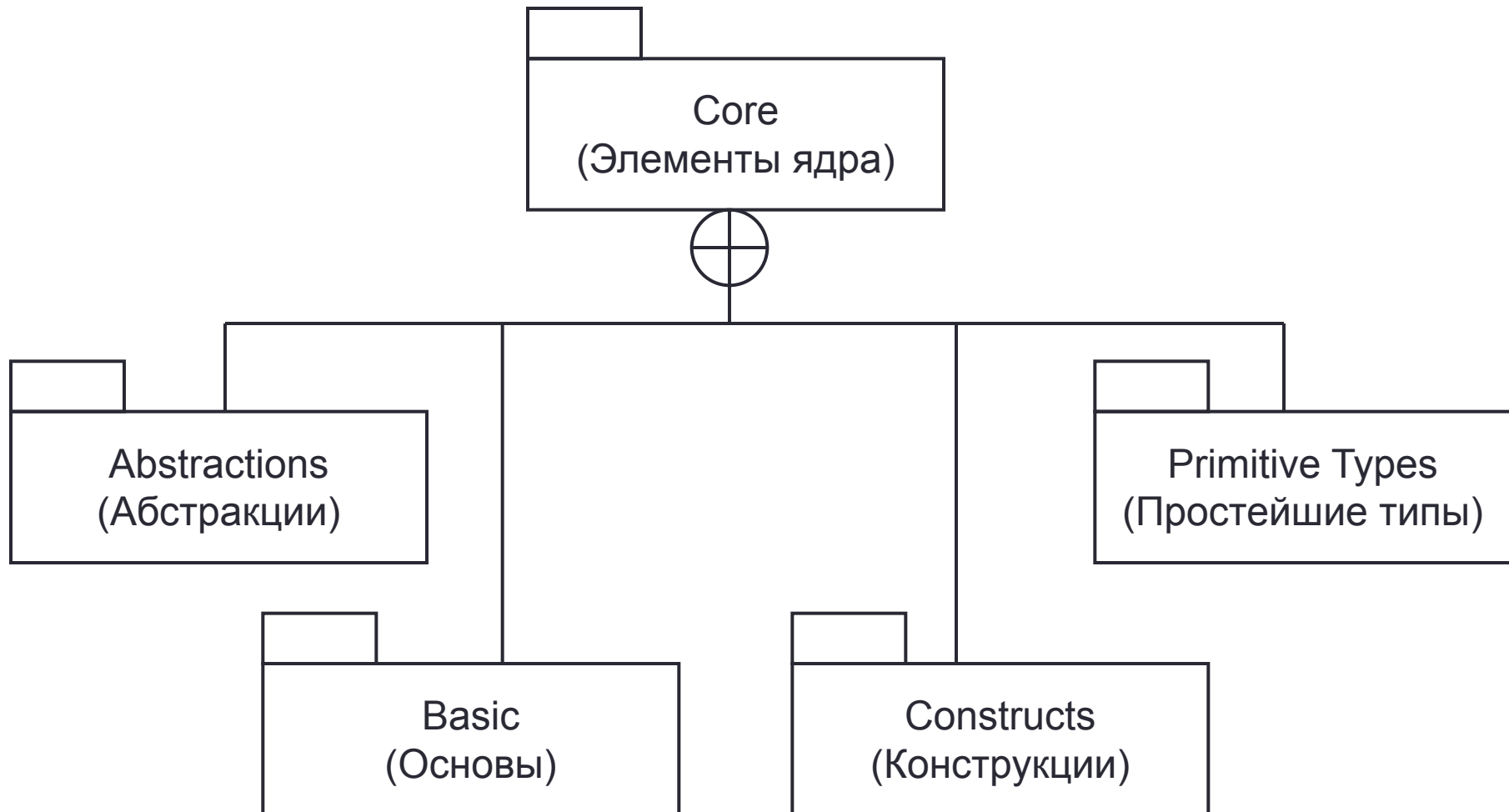
Модели поведения – модели, предназначенные для описания процесса функционирования элементов системы, включая их методы и взаимодействие между ними, а также процесс изменения состояний отдельных элементов и системы в целом.

Общая структура UML 2

Формальное описание самого языка UML 2 основывается на некоторой общей иерархической структуре модельных представлений, состоящей из четырех уровней:

- мета-метамодель (M3);
- метамодель (M2);
- модель (M1);
- объекты или экземпляры (M0).

Основные пакеты метамодели языка UML 2



Диаграммы языка UML 2

Диаграммы структуры:

- диаграмма классов;
- диаграмма композитной структуры;
- диаграмма пакетов;
- диаграмма объектов;
- диаграмма компонентов;
- диаграмма развертывания.

Диаграммы поведения:

- диаграмма вариантов использования;
- диаграмма деятельности;
- диаграмма конечного автомата.

Диаграммы взаимодействия:

- диаграмма последовательности;
- диаграмма коммуникации;
- диаграмма обзора взаимодействия;
- временная диаграмма.

Особенности изображения диаграмм в нотации UML 2

Три типа визуальных графических изображения:

- Геометрические фигуры на плоскости, играющие роль вершин графов соответствующих диаграмм.
- Графические пути, которые представляются различными линиями на плоскости.
- Специальные графические символы, изображаемые вблизи от тех или иных визуальных элементов диаграмм и имеющие характер дополнительной спецификации.

Рекомендации при построении диаграмм языка UML 2

1. Каждая диаграмма должна служить законченным представлением соответствующего фрагмента моделируемой предметной области.
2. Все сущности на диаграмме модели должны быть одного концептуального уровня.
3. Вся информация о сущностях должна быть явно представлена диаграммах.
4. Диаграммы не должны содержать противоречивой информации.
5. Диаграммы не следует перегружать текстовой информацией.
6. Каждая диаграмма должна быть самодостаточной для правильной интерпретации всех ее элементов и понимания семантики всех используемых графических символов.
7. Количество диаграмм различных типов для модели конкретного приложения не является строго фиксированным.
8. Любая из моделей должна содержать только те элементы, которые определены в нотации языка UML 2.

Рекомендации при построении диаграмм языка UML 2

Типы диаграмм UML 2, указываемые в заголовках фреймов:

- **activity (act)** – для фреймов диаграммы деятельности;
- **class (cs)** – для фреймов диаграммы классов;
- **component (cmp)** – для фреймов диаграммы КОМПОНЕНТОВ;
- **interaction (sd)** – для фреймов диаграммы взаимодействия;
- **package (pkg)** – для фреймов диаграммы пакетов;
- **state machine (stm)** – для фреймов диаграммы конечных автоматов;
- **use case (uc)** – для фреймов диаграммы вариантов использования.

Механизмы расширения UML 2

Предназначение механизмов расширения:

- уточнение или специализация общих элементов метамодели при разработке конкретных моделей в нотации языка UML 2;
- определения таких расширений языка UML 2, которые зависят от специфики моделируемого процесса или от языка реализации программного кода;
- присоединения произвольной семантической или несемантической информации к элементам модели.

Механизмы расширения UML 2

Стереотип – элемент модели, который расширяет семантику базового элемента метамодели языка UML 2.

Способы представления стереотипов:

- в форме текста, заключенного в двойные угловые кавычки и размещенного выше или перед именем элемента модели;
- в форме графической пиктограммы, которая заменяет текст имени соответствующего стереотипа;
- в форме прямоугольника класса с уменьшенной пиктограммой стереотипа внутри этого прямоугольника, расположенного, как правило, в верхнем правом углу.

Механизмы расширения UML 2

Ограничение – логическое условие, выраженное текстом на естественном языке или на языке, который может быть прочитан некоторой машиной, с целью декларирования семантики некоторого элемента модели.

`<ограничение> ::= '{' [<имя> ':'] <логическое-выражение> '}'`

Помеченное значение – явное определение некоторого семейства объектов в форме пары «имя - значение». В помеченном значении имя называется также тегом.

Диаграмма вариантов использования

Диаграмма вариантов использования – диаграмма, на которой изображаются варианты использования проектируемой системы, заключенные в границу субъекта, и внешние актеры, а также определенные отношения между актерами и вариантами использования.

Назначение диаграммы вариантов использования:

- определить общие границы функциональности проектируемой системы в контексте моделируемой предметной области;
- специфицировать требования к функциональному поведению проектируемой системы в форме вариантов использования;
- 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.

Субъект вариантов использования

Субъектом в контексте языка UML 2 называется любой элемент модели, который обладает функциональным поведением.

Основные графические элементы диаграммы вариантов использования:

- вариант использования;
- актер;
- примечание.

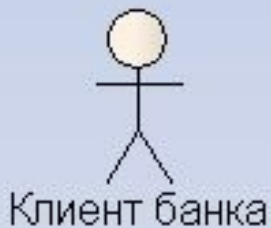
Вариант использования

Вариант использования представляет собой общую спецификацию совокупности выполняемых системой действий с целью представления некоторого наблюдаемого результата, который имеет значение для одного или нескольких актеров.



Актер

Актер представляет собой любую внешнюю по отношению к проектируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей или решения частных задач.



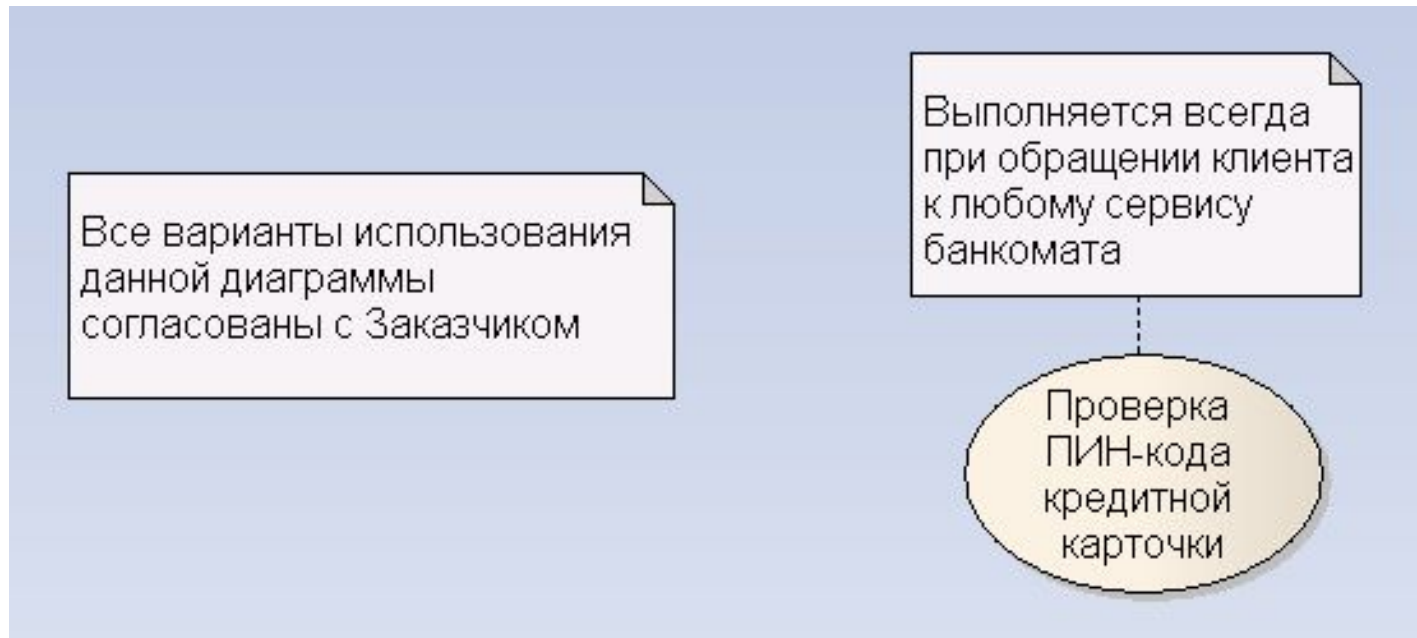
«actor»
Посетитель
Интернет-магазина



Удаленный пользователь

Комментарий

Комментарий в языке UML 2 предназначен для включения в модель произвольной текстовой информации в форме примечания, которое может быть присоединено к одному или нескольким элементам разрабатываемой модели.



Отношения на диаграмме вариантов использования

Под отношением в языке UML 2 понимается произвольная семантическая взаимосвязь между отдельными элементами модели.

На диаграммах вариантов использования могут применяться следующие отношения:

- ассоциации;
- включение;
- расширение;
- обобщение.

Отношение ассоциации

Ассоциация является одним из фундаментальных понятий в языке UML 2 и может использоваться на различных канонических диаграммах при построении визуальных моделей.

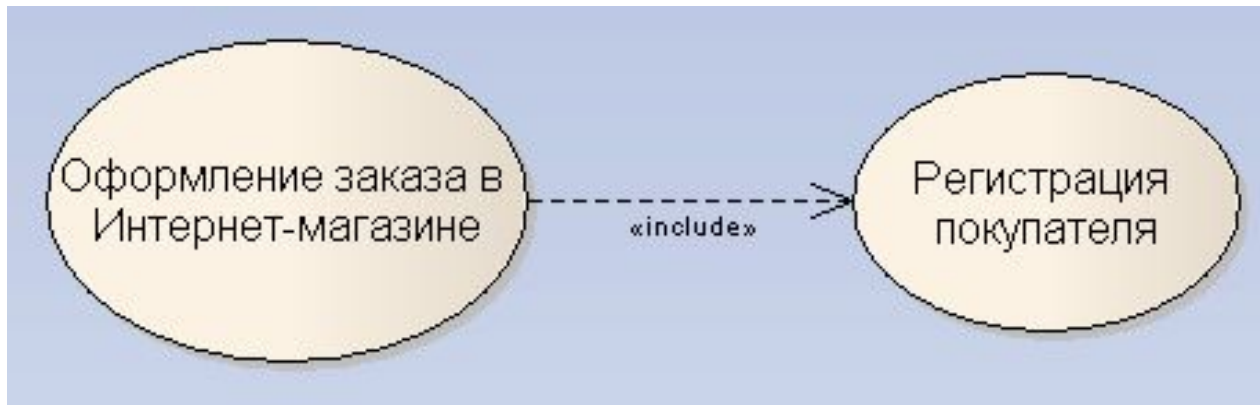
Применительно к диаграммам вариантов использования отношение ассоциации может служить только для обозначения взаимодействия актера с вариантом использования.



Отношение включения

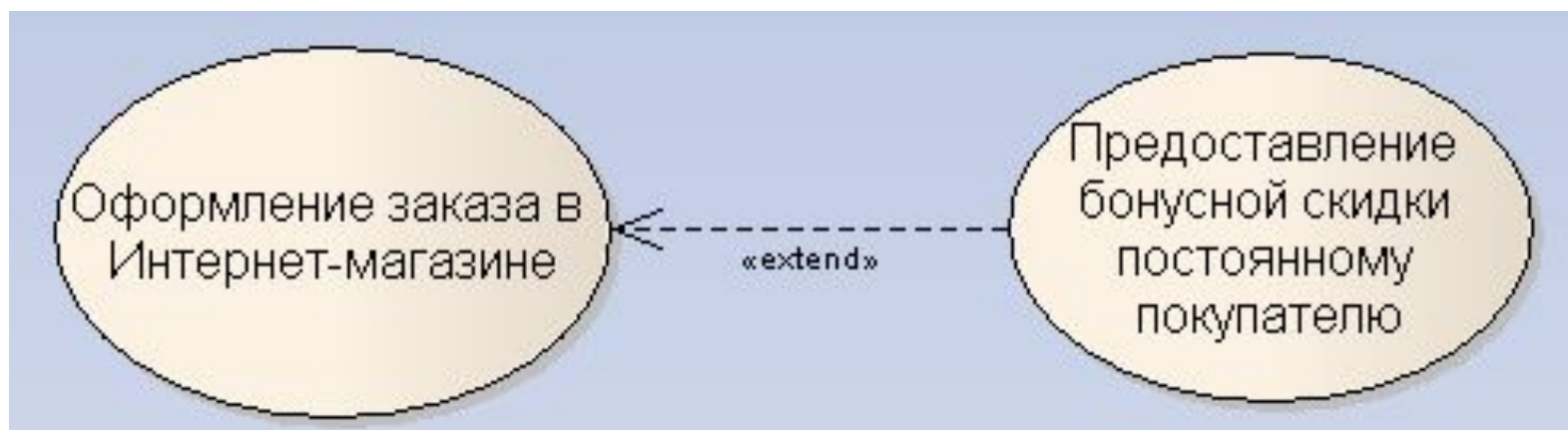
Отношение зависимости определяется как форма взаимосвязи между двумя элементами модели, предназначенная для спецификации того обстоятельства, что изменение одного элемента модели приводит к изменению некоторого другого элемента.

Отношение включения специфицирует тот факт, что некоторый вариант использования содержит поведение, определенное в другом варианте использования.



Отношение расширения

Отношение расширения определяет взаимосвязь одного варианта использования с некоторым другим вариантом использования, функциональность или поведение которого задействуется первым не всегда, а только при выполнении некоторых дополнительных условий.



Отношение расширения



Отношение обобщения

Отношение обобщения предназначено для спецификации того факта, что один элемент модели является специальным или частным случаем другого элемента модели.



