

Анализ и проектирование на UML

Максим Валерьевич Хлопотов

Темы лекционных занятий

1. Введение в UML
- 2. Моделирование использования**
3. Моделирование структуры
4. Моделирование поведения
- 5. Дисциплина моделирования**

План лекции

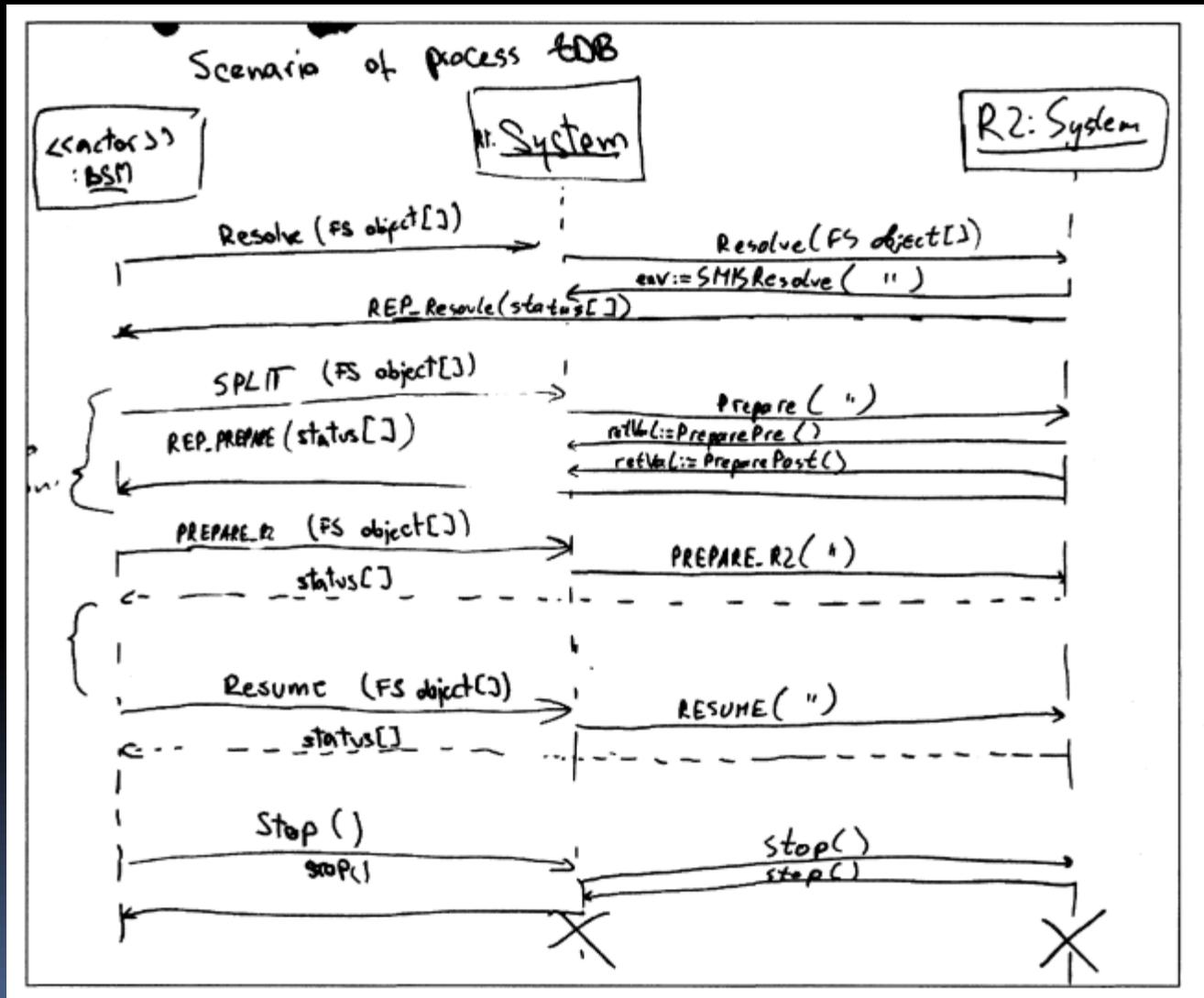
1. Диаграмма деятельности для моделирования использования
2. Общие механизмы и свойства модели
3. Использование UML в процессе разработки

Выделяют три режима использования UML разработчиками:

- режим эскиза,
- режим проектирования,
- режим языка программирования.

[Мартин Фаулер. UML. Основы.]

Режим Эскиза



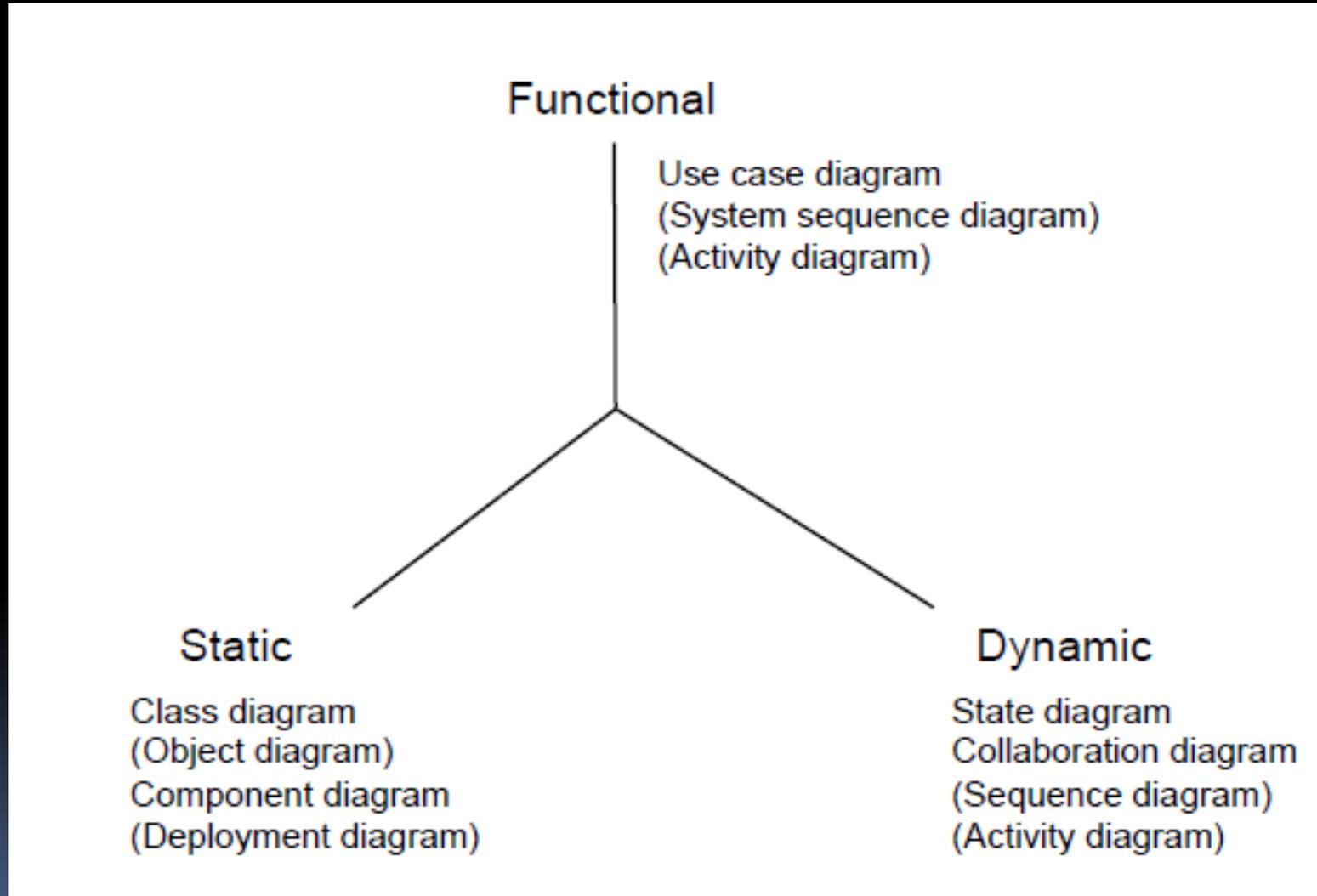
Режимы использования UML

В режиме эскиза разработчики используют UML для обмена информацией о различных аспектах системы.

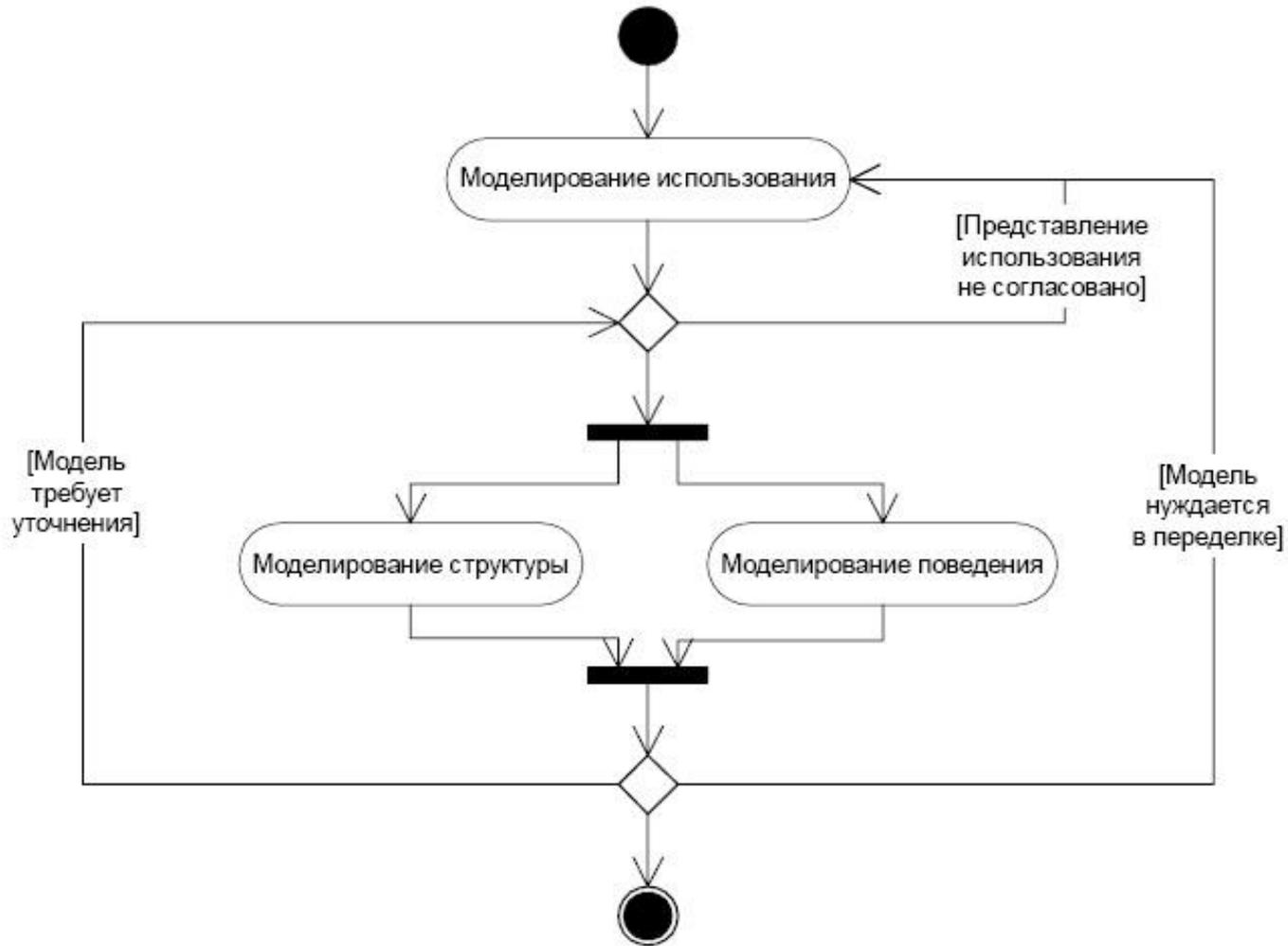
В режиме проектирования можно использовать эскизы при прямой и обратной разработке. При *прямой разработке* диаграммы рисуются до написания кода, а при *обратной разработке* диаграммы строятся на основании исходного кода, чтобы лучше понять его.

В режиме языка программирования разработчики рисуют диаграммы, которые компилируются прямо в исполняемый код, а UML становится исходным кодом.

Оси моделирования



Процесс моделирования



Моделирование использования

- Составление диаграмм использования – это первый шаг моделирования.
- Основное назначение диаграммы использования – показать, что делает система во внешнем мире.
- Идентификация действующих лиц и вариантов использования – ключ к дальнейшему проектированию.
- В зависимости от выбранной парадигмы проектирования и программирования применяются различные способы реализации вариантов использования.

Моделирование использования

- Реализация варианта использования диаграммой деятельности является компромиссным способом ведения разработки – в сущности, это проектирование сверху вниз в терминах и обозначениях UML.
- Другие способы реализации вариантов использования: текстовое описание прецедентов, диаграммы взаимодействия (диаграмма последовательности).

Диаграмма деятельности

Используются для моделирования процесса выполнения операций.

Частный случай диаграмм состояний.

Диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются *состояния действия* или *деятельности*, а дугами - переходы от одного *состояния действия* к другому.

Пример

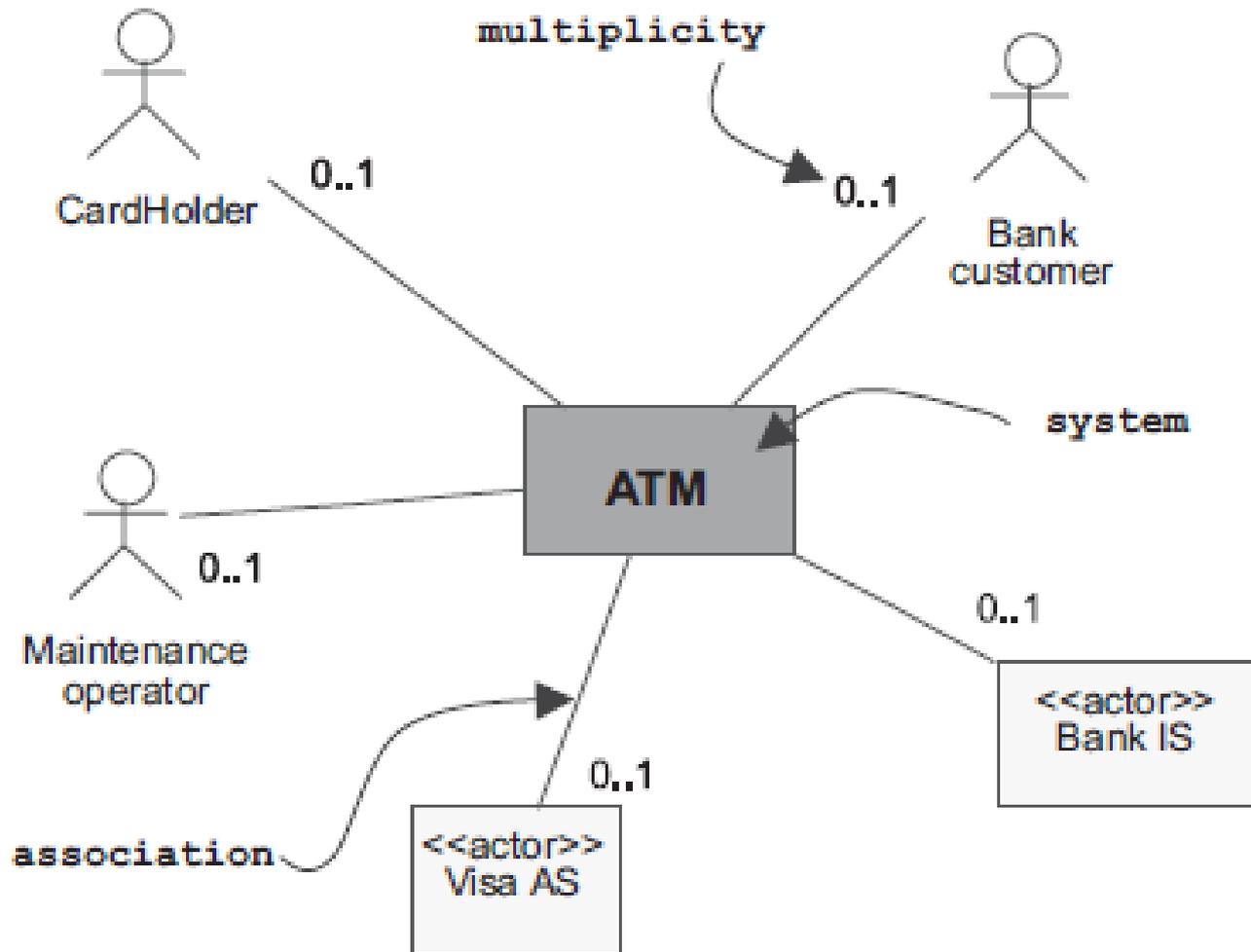
Упрощенная версия банкомата (automatic teller machine, АТМ) предлагает пользователю следующие сервисы:

1. Управление финансами владельцев карт при помощи карт ридера и диспенсера.
2. Информирование о состоянии счета, возможность пополнять счет и оплачивать услуги для владельцев карт банка, который обслуживает банкомат.

Не забудьте также:

3. Все транзакции должны быть безопасными. С этой целью происходит аутентификация через Visa authorisation system (для простоты ограничимся только этой системой).
4. Банкомат нуждается в регулярном обслуживании. Например, требуется пополнить диспенсер.

Идентификация действующих лиц



Идентификация вариантов использования

CardHolder:

- Withdraw money

Bank customer:

- Withdraw money.
- Consult the balance of one or more accounts.
- Deposit cash.
- Deposit cheques.

Maintenance operator:

- Refill dispenser.
- Retrieve cards that have been swallowed.
- Retrieve cheques that have been deposited.

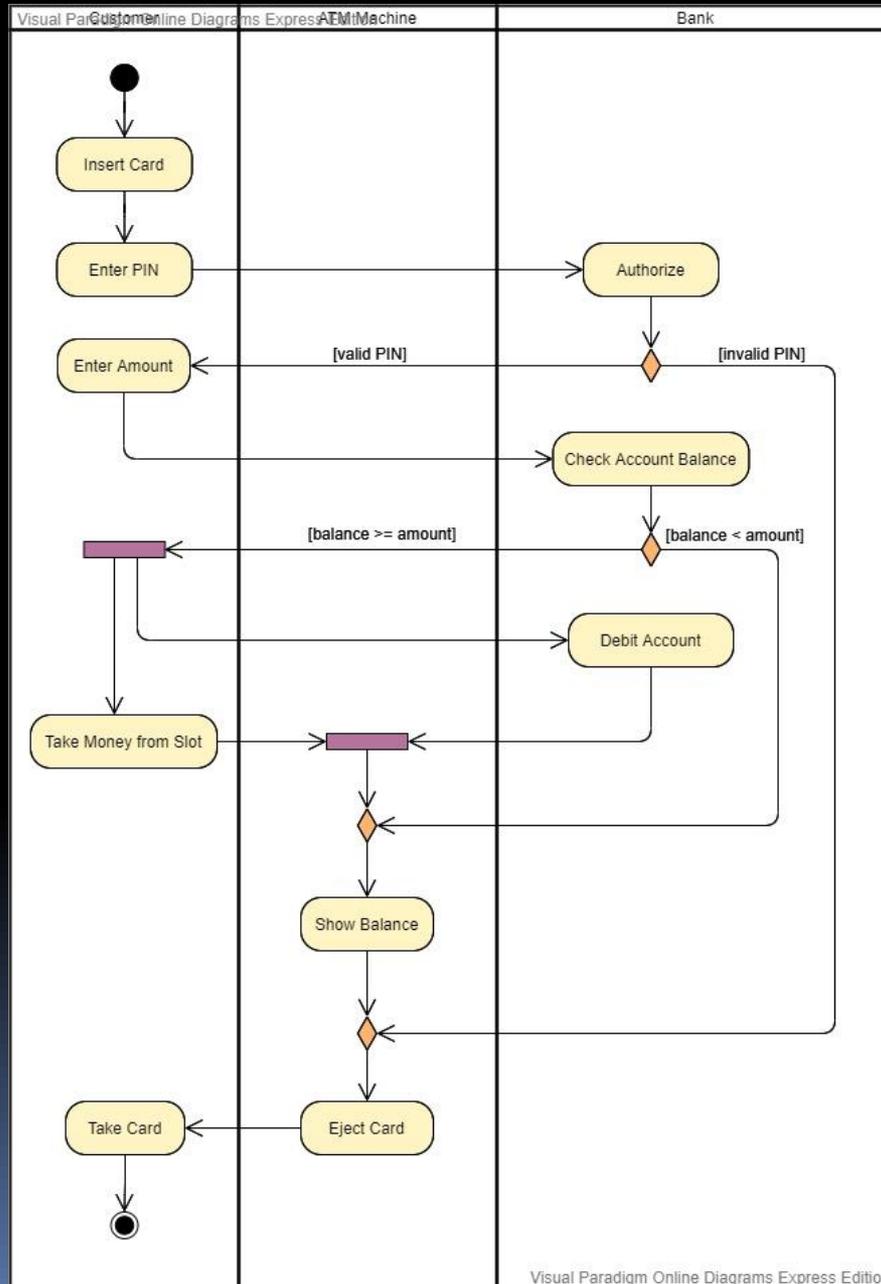
Visa authorisation system (AS):

- None.

Bank information system (IS):

- None.

Реализация варианта использования



Общие механизмы

В UML имеются общие правила и механизмы, которые относятся ко всему языку в целом.

Выделяют следующие общие механизмы:

- ▶ внутреннее представление модели;
- ▶ дополнения;
- ▶ подразделения;
- ▶ механизмы расширения.

Внутреннее представление

Модель имеет внутреннее представление

► Для графов используются способы представления: матрица смежности, списки смежности и др.

Разработчики инструментов для моделирования на UML придумать свое (что обычно и делается).

У каждого элемента модели есть «оборотная сторона», где записаны все свойства, даже те, которые в данном контексте не нужно или нельзя показывать на картинке.

Внутреннее представление

Внутреннее представление содержит список стандартных свойств, определенных для каждого элемента модели. Такое внутреннее представление может быть однозначно переведено во внешнее представление.

Внутреннее представление может быть переведено в текст в формате XML (конкретное приложение XML) без потери информации.

Текстовое представление моделей UML используется инструментами моделирования, например, для обмена моделями.

Дополнения



Базовая графическая нотация может быть расширена путем использования дополнительных текстовых и/или графических объектов, присоединяемых к базовой нотации.

Такие дополнительные объекты называются **дополнения**.

Дополнения позволяют показать на диаграмме те части внутреннего представления модели, которые не отображаются с помощью базовой графической нотации.

Подразделения



UML является объектно-ориентированным языком, поэтому базовые понятия объектно-ориентированного подхода имеют в языке сквозное действие.

Подразделения



Пример 1. Четко различается о чём идет речь: об общем описании некоторого множества однотипных объектов (т. е. о классе) или о конкретном объекте из некоторого множества однотипных объектов (т. е. об экземпляре класса).

Это различие передается единообразно: если это конкретный объект, то его имя подчеркивается; если это класс, то оно не подчеркивается.

Подразделения



Пример 2. Если абстрактный интерфейс, то при записи имени используется курсивное начертание шрифта, если конкретная реализация — используется прямое начертание.

Механизмы расширения

Механизмы расширения — это встроенный в язык способ изменить язык. Авторы UML при унификации различных методов постарались включить в язык как можно больше различных средств (удерживая объем языка в рамках разумного), так чтобы язык оказался применимым в разных контекстах и предметных областях. Но вполне могут возникать и возникают случаи, когда стандартных элементов моделирования не хватает или они не вполне адекватны.

Механизмы расширения



Механизмы расширения позволяют определять новые элементы модели на основе существующих управляемым и унифицированным способом. Таких механизмов три:

- помеченные значения;
- ограничения;
- стереотипы.

Механизмы расширения



Помеченное значение — это пара: имя свойства и значение свойства, которую можно добавить к любому стандартному элементу модели.

К любому элементу модели можно добавить любое помеченное значение, которое будет храниться также, как и все стандартные свойства данного элемента. Способ обработки помеченного значения, определенного пользователем, стандартом не определяется и отдается на откуп инструменту.

Механизмы расширения

Помеченные значения записываются в модели в виде строки текста, имеющей следующий синтаксис: в фигурных скобках указывается пара: имя и значение, разделенные знаком равенства. Можно указывать сразу список пар, разделяя элементы сл



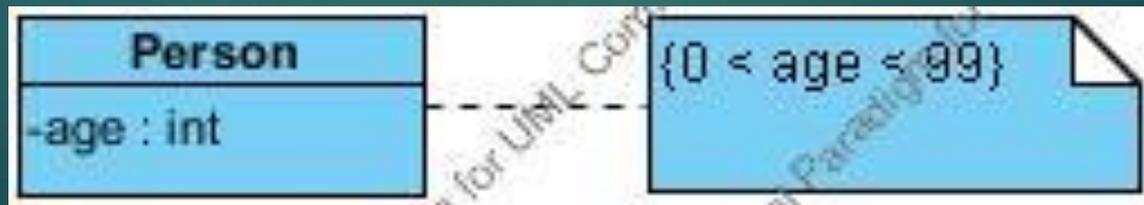
Механизмы расширения

Ограничение — это логическое утверждение относительно значений свойств элементов модели.

Логическое утверждение может иметь два значения: истина и ложь, то есть задаваемое им условие либо выполняется, либо не выполняется. Указывая ограничение для элемента модели, мы расширяем его семантику, требуя, чтобы ограничение выполнялось. Ограничение может относиться к отдельному элементу или к совокупности элементов модели или к совокупности элементов модели.

Механизмы расширения

Ограничения записываются в виде строки текста, заключенной в фигурные скобки. Это может быть неформальный текст на естественном языке, логическое выражение языка программирования, поддерживаемого инструментом или выражение на некотором другом формальном языке, специально включенного для этой цели в UML.



Механизмы расширения



Стереотип — это определение нового элемента моделирования в UML на основе существующего элемента моделирования.

Определение стереотипа производится следующим образом. Взяв за основу некоторый существующий элемент модели, к нему добавляют новые помеченные значения (расширяя тем самым внутреннее представление), новые ограничения (расширяя семантику) и дополнения, то есть новые графические элементы (расширяя нотацию).

Механизмы расширения



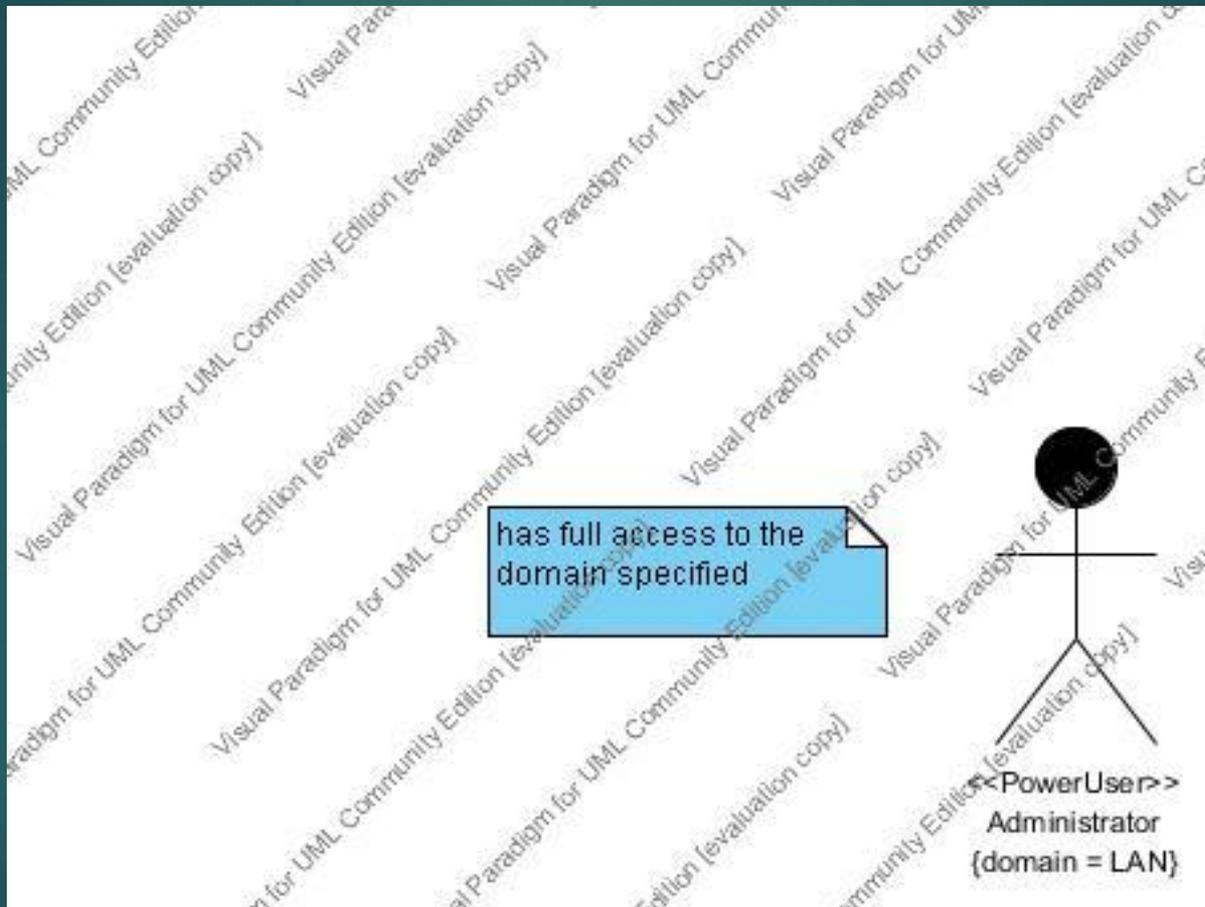
После того, как стереотип определен, его можно использовать как элемент модели нового типа. Если при создании стереотипа не использовались дополнения и графическая нотация взята от базового элемента модели, на основе которого определен стереотип, то стереотип элемента обозначается при помощи имени стереотипа, заключенного в двойные угловые скобки, которое помещается перед именем элемента модели. Если же для стереотипа определена своя нотация, например, новый графический символ, то указывается этот символ.

Механизмы расширения



В UML имеется большое количество predetermined стереотипов. Стереотипы используются очень часто, поэтому примеры их применения рассредоточены по всей книге. Стереотипы являются мощным механизмом расширения языка, однако им следует пользоваться умеренно. Определяя большое число нестандартных стереотипов, легко сделать модель непонятной никому, кроме автора.

Механизмы расширения



Общие свойства модели

Модель в целом может обладать (или не обладать) важными свойствами, которые оказывают значительное влияние на ее практическую применимость.

Основные свойства модели:

- ▶ правильность;
- ▶ непротиворечивость;
- ▶ полнота;
- ▶ вариации семантики.

Общие свойства модели

- ▶ Прежде всего, модель должна удовлетворять формальным требованиям к описанию сущностей, отношений и их комбинаций. Т.е. модель должна быть синтаксически **правильной**.
- ▶ Например, отношение (ребро в графе модели) всегда определяется между сущностями, на диаграмме линия должна начинаться и заканчиваться в фигуре, иначе это синтаксическая ошибка.

Общие свойства модели

- ▶ В некоторых случаях даже синтаксически правильная модель может содержать такие конструкции, семантика которых не определена или неоднозначна.
- ▶ Такая модель называется **противоречивой** (ill formed), а модель, в которой все в порядке и семантика всех конструкций определяется однозначно, называется **непротиворечивой** (well formed).

Общие свойства модели

- ▶ Например, пусть мы определим в модели, что класс А является подклассом класса В, класс В — подкласс С, а класс С — подкласс А. Каждое из этих отношений обобщения в отдельности допустимо и синтаксически правильно, а все вместе они образуют противоречие.
- ▶ Ответственность за непротиворечивость модели лежит на ее авторе.

Общие свойства модели

- ▶ Модель не создается мгновенно — она появляется в результате многочисленных итераций и на каждой из них не полна.
- ▶ В некоторых случаях оказывается достаточно одной диаграммы использования, а в других необходимы диаграммы всех типов, прорисованные до мельчайших деталей. Все зависит от прагматики, т. е. от того, для чего составляется модель.
- ▶ **Полнота** – критерий субъективный.

Общие свойства модели

- ▶ В описании семантики UML определено некоторое количество точек **вариации семантики**.
- ▶ Стандарт говорят: «мы понимаем это так-то и так-то, но допускаем, что другие могут это понимать иначе».
- ▶ При реализации языка в конкретном инструменте разработчики в точке вариации семантики вправе выбрать альтернативный вариант, если он не противоречит семантике остальной части языка.
- ▶ Рядовому пользователю точки вариации семантики не заметны и он может о них не думать.

Процесс разработки

- Начальная фаза — это краткий период формирования общего видения и рамок проекта. Он включает анализ примерно 10% прецедентов, осмысление основных нефункциональных требований, создание бизнес-плана и подготовку среды разработки, чтобы уже на следующей стадии развития можно было приступить к программированию.
- Use case driven development

Процесс разработки

- Для большинства проектов необходим небольшой начальный этап, на котором нужно сформулировать ответы на следующие вопросы.
- Каково ваше видение проекта?
- Реально ли осуществить задуманное?
- Что лучше: купить или разработать?
- В какую сумму (примерно) обойдется реализация проекта: 10-100 тысяч или миллионы долларов?
- Стоит ли браться за этот проект?

Процесс разработки

- Определив свое видение задачи и приблизительно оценив необходимые затраты, можно приступать к изучению требований. Однако задачей начальной фазы не является определение всех требований, составление правдоподобного бюджета или плана реализации проекта.
- Большая часть анализа требований приходится на стадию развития, при этом анализ требований выполняется параллельно с созданием окончательного кода и его тестирования.

Процесс разработки

- С точки зрения UML на начальной фазе создаются следующие артефакты:
 - Модель вариантов использования (прецедентов)
 - Детализация самых приоритетных вариантов использования

Выводы

- Составление диаграмм использования — это первый шаг моделирования.
- Реализация варианта использования диаграммой деятельности является компромиссным способом ведения разработки — в сущности, это проектирование сверху вниз.
- В UML имеются общие правила и механизмы, которые относятся ко всему языку в целом. Выделяют следующие общие механизмы: внутреннее представление модели; дополнения; подразделения; механизмы расширения.

Выводы

- Основные свойства модели: правильность; непротиворечивость; полнота; вариации семантики.
- UML не накладывает никаких ограничений на процесс разработки. Однако, наиболее естественным является «use case driven» подход. Это способ итеративной разработки, в котором в первую очередь реализуются наиболее важные варианты использования.