

# Лекция 4: Моделирование и анализ

Юрий Литвинов  
y.litvinov@spbu.ru

22.09.2022

# Computer-Aided Software Engineering

- ▶ В 80-е годы термином CASE называли всё, что помогает разрабатывать ПО с помощью компьютера
  - ▶ Даже текстовые редакторы
- ▶ Теперь — прежде всего средства для визуального моделирования (UML-диаграммы, ER-диаграммы и т.д.)
- ▶ Отличаются от графических редакторов тем, что “понимают”, что в них рисуют
- ▶ Нынче чаще используются термины “MDE tool”, “UML tool” и т.д.

# Типичная функциональность CASE-инструментов

- ▶ Набор визуальных редакторов
- ▶ Репозиторий
- ▶ Набор генераторов
- ▶ Текстовый редактор
- ▶ Редактор форм
- ▶ Средства обратного проектирования (reverse engineering)
- ▶ Средства верификации и анализа моделей
- ▶ Средства эмуляции и отладки
- ▶ Средства обеспечения командной разработки
- ▶ API для интеграции с другими инструментами
- ▶ Библиотеки шаблонов и примеров

# Примеры CASE-инструментов

- ▶ “Рисовалки”
  - ▶ Visio
  - ▶ Dia
  - ▶ SmartDraw
  - ▶ LucidChart
  - ▶ <http://plantuml.com/>
- ▶ Полноценные CASE-системы
  - ▶ Enterprise Architect
  - ▶ Rational Software Architect
  - ▶ MagicDraw
  - ▶ Visual Paradigm
  - ▶ GenMyModel
- ▶ Браузерные инструменты
  - ▶ <https://www.websequencediagrams.com/>
  - ▶ <http://yuml.me/>
  - ▶ <https://mermaid-js.github.io/>

# Моделирование требований

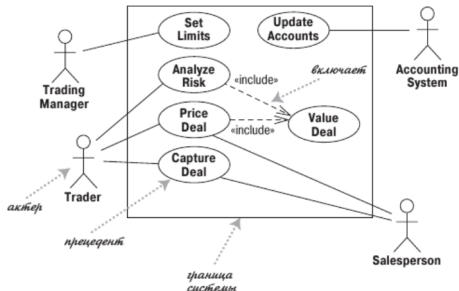
Первый этап разработки любой системы — сбор и анализ требований

- ▶ Понимание разработчиками решаемой задачи
- ▶ Соглашение между разработчиками, заказчиками и пользователями
  - ▶ Заказчики и пользователи часто разные люди с разными потребностями
- ▶ Чёткое обозначение границ системы
- ▶ Основа для планирования проекта
- ▶ Чаще всего словесное описание требований, реже формальные модели

# Диаграмма случаев использования UML

## Диаграмма прецедентов

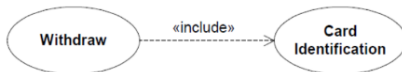
- ▶ Ивар Яacobсон, 1992 год
- ▶ Акторы (или актёры, роли) — внешние сущности, использующие систему
  - ▶ Люди или другие программные системы
- ▶ Случаи использования (прецеденты) — цель использования системы актором
  - ▶ Раскрываются в набор сценариев, описываемых чаще текстом



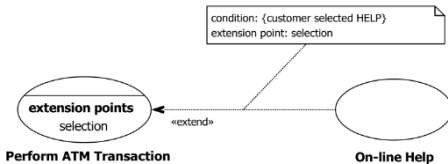
© М. Фаулер, UML. Основы

# Include и Extend

Include:



Extend:



© OMG, UML 2.5 Specification





## Сценарий использования, типичная структура

- ▶ Заголовок (цель основного актора)
- ▶ Заинтересованные лица, акторы, основной актор
- ▶ Предусловия
- ▶ Триггеры (активаторы)
- ▶ Основной порядок событий
- ▶ Альтернативные пути и расширения
- ▶ Постусловия

## Моделирование и анализ

# Контекстная диаграмма IDEF0

- ▶ Обозначает границы системы и способы её взаимодействия с внешним миром
- ▶ Используется для моделирования не только ПО
- ▶ Каждая сторона имеет свой смысл
  - ▶ Слева — входные данные или материалы
  - ▶ Сверху — управление
  - ▶ Снизу — механизмы
  - ▶ Справа — выходные данные или продукты

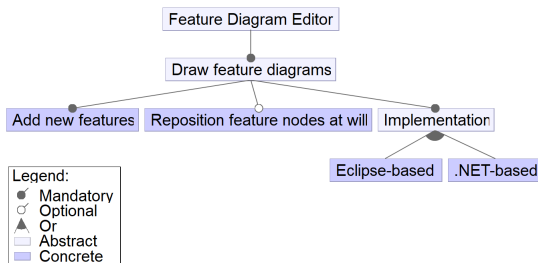


© <http://ecm-journal.ru>

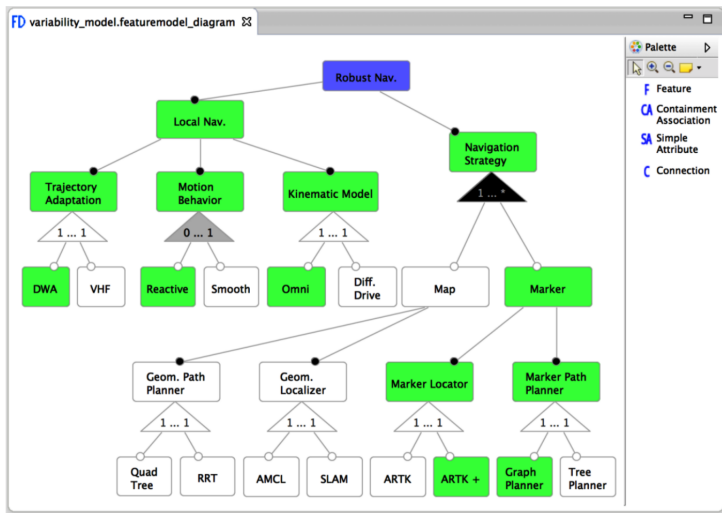
# Диаграмма характеристик

## Feature Diagram

- ▶ Представляет функциональность системы в виде дерева
- ▶ Используется в основном для моделирования семейств программных продуктов (Product lines)

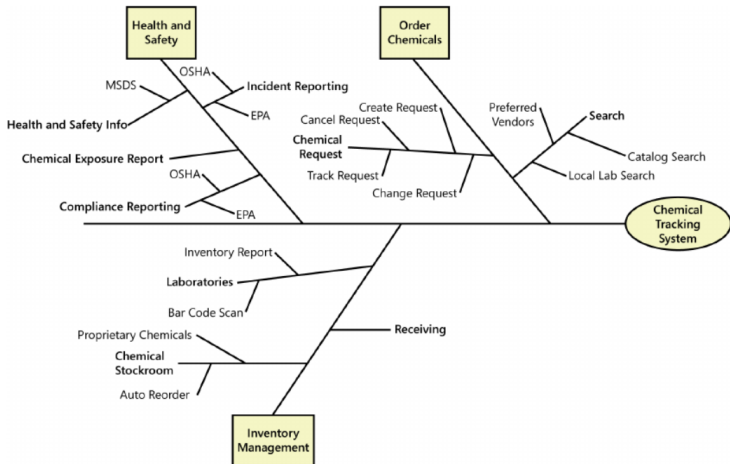


# Диаграмма характеристик, пример



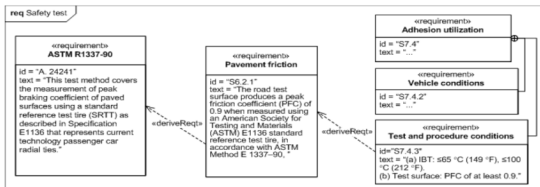
© D. Brugali

# Feature Tree



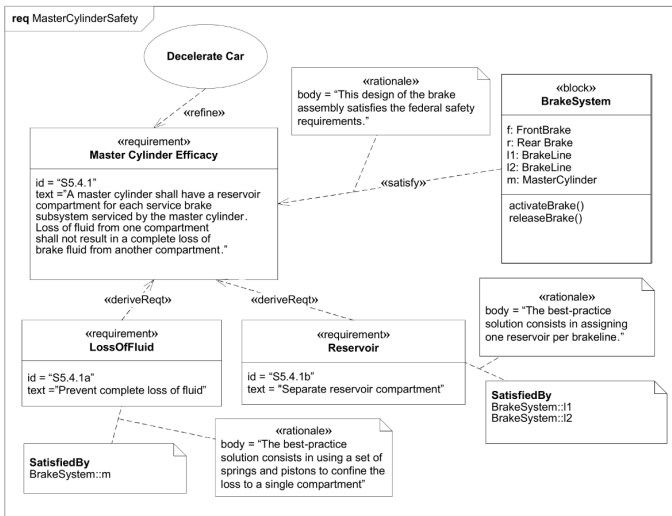
# Диаграмма требований, SysML

- ▶ Более формальная нотация дерева фич



© OMG SysML 1.4 Specification

# Диаграмма требований SysML, пример

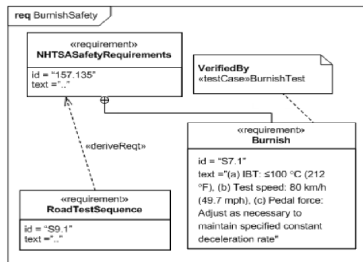


© OMG SysML 1.4 Specification

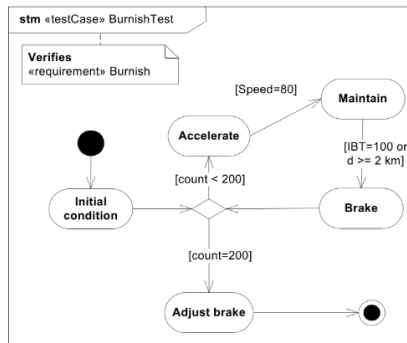


# Диаграмма требований SysML и тесты

## Требования:



## Сценарий тестирования:

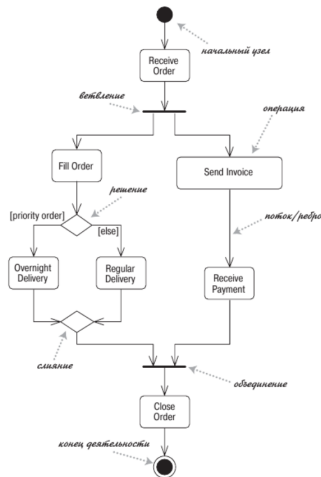


© OMG SysML 1.4 Specification

# Диаграмма активностей UML

## Диаграммы деятельности

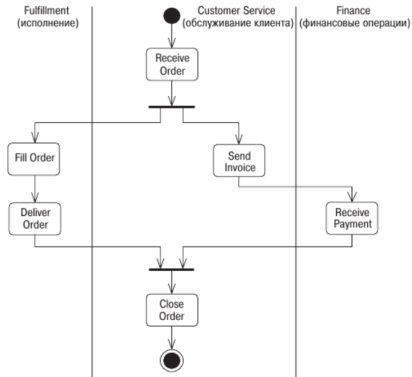
- ▶ Используются для моделирования бизнес-процессов, тоже на первых этапах
  - ▶ Может быть визуализацией сценария использования
- ▶ Иногда — для моделирования алгоритма
- ▶ Расширенные блок-схемы
- ▶ Семантика на основе сетей Петри



© М. Фаулер, UML. Основы

# Диаграмма активностей, разделы

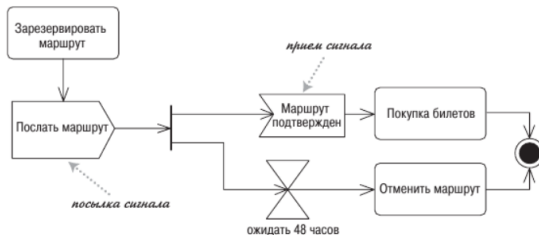
- ▶ Раздел представляет отдел организации (или организацию), отвечающий за часть работы
- ▶ Визуализирует поток работ между отделами



© М. Фаулер, UML. Основы

## Диаграмма активностей, сигналы

- ▶ Для визуализации асинхронных процессов
- ▶ Сигналом может быть посылка документа, запрос и т.д.

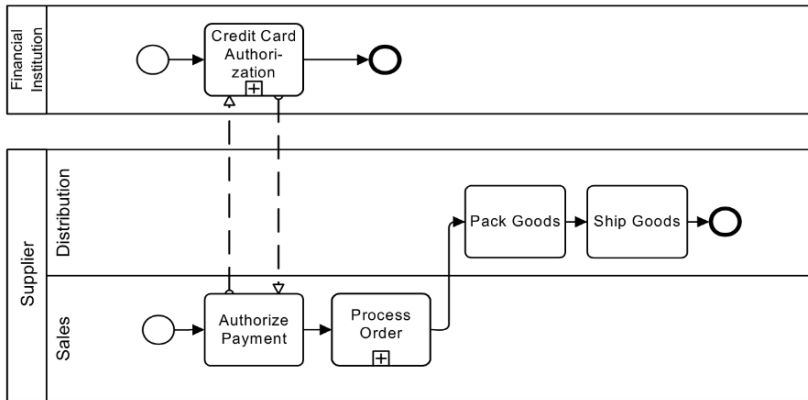


© М. Фаулер, UML. Основы

# Business Process Model and Notation

- ▶ Версия 1.0 в 2004 году, текущая (2.0) — в 2011
- ▶ Для описания бизнес-процессов
  - ▶ Сильно продвинутые диаграммы активностей
  - ▶ Позволяют описывать группы взаимодействующих процессов
  - ▶ Исполнимая семантика
  - ▶ Правила генерации в BPEL
    - ▶ Business Process Execution Language

# Пример диаграммы



© OMG BPMN 2.0 Specification

# События

	Начальные	Промежуточные	Завершающие	
	Обработка		Генерация	
Простое				
Сообщение				
Таймер				
Ошибка				
Отмена				
Компенсация				
Условие				
Сигнал				
Составное				
Ссылка				
Останов				

© <https://ru.wikipedia.org/wiki/BPMN>

# Операторы ветвления



Оператор исключающего ИЛИ, управляемый данными



Оператор исключающего ИЛИ, управляемый событиями



Оператор включающего ИЛИ



Оператор И

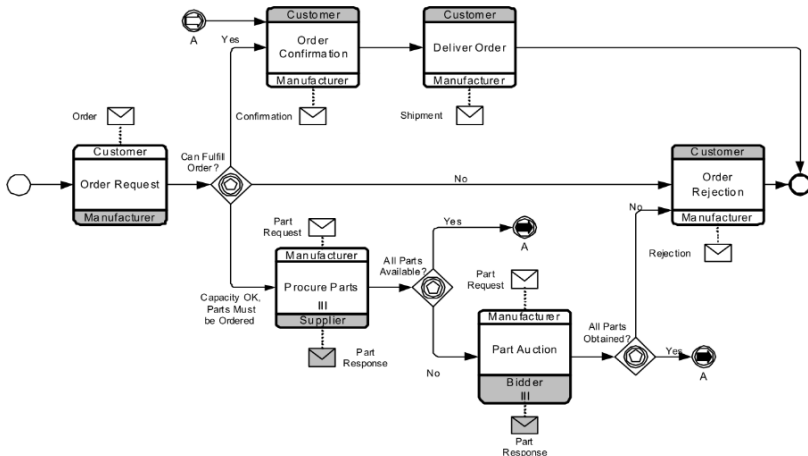


Сложный оператор

© <https://ru.wikipedia.org/wiki/BPMN>

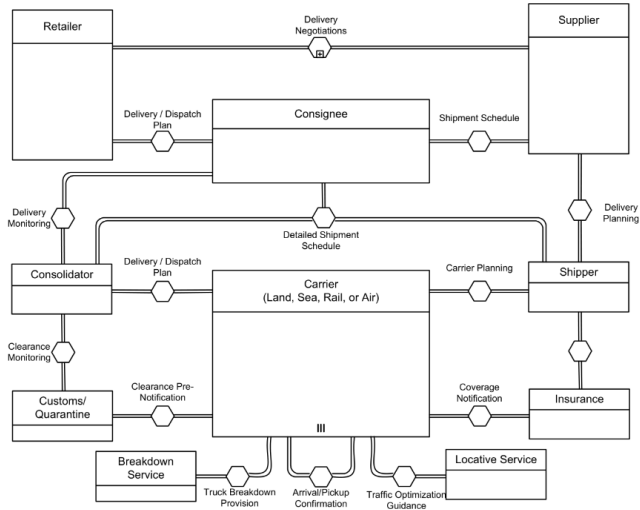


# Диаграмма хореографии



© OMG BPMN 2.0 Specification

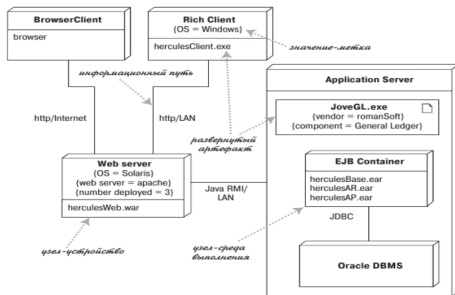
# Диаграмма диалогов



© OMG BPMN 2.0 Specification

# Диаграмма развёртывания UML

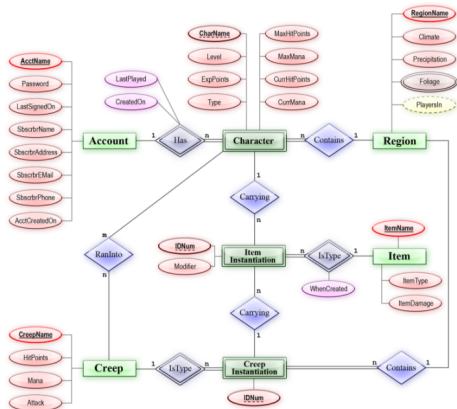
- ▶ Показывает отображение компонентов и физических артефактов на реальные (или виртуальные) устройства
- ▶ Бывает полезна на начальных этапах проектирования, даже до диаграмм компонентов



© М. Фаулер, UML. Основы

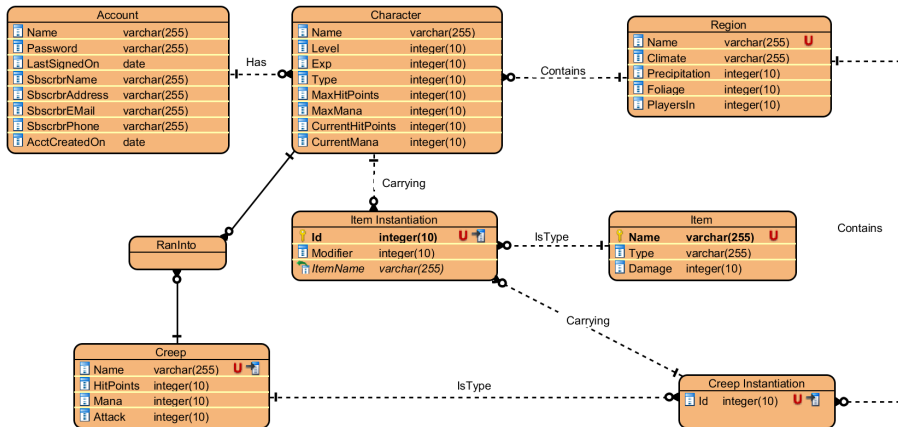
# Диаграммы “Сущность-связь”

- ▶ Описывают концептуальную модель предметной области
- ▶ Идеальны для моделирования схем реляционных баз данных
- ▶ 1976 год, Питер Чен

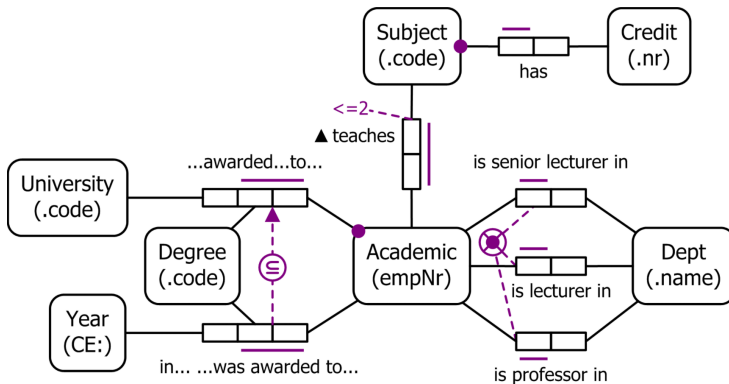


© <https://ru.wikipedia.org>

# Нотация “Вороньей лапки”



# Object-Role Modeling



© <http://www.orm.net>