

# Лекция 4: Моделирование и анализ

Юрий Литвинов  
yurii.litvinov@gmail.com

28.02.2022

# 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

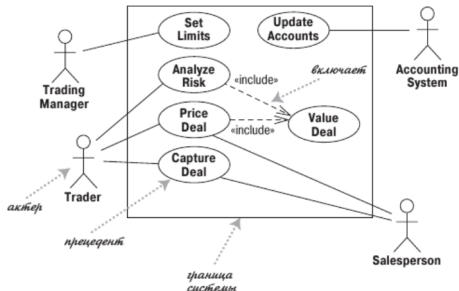
Первый этап разработки любой системы — сбор и анализ требований

- ▶ 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

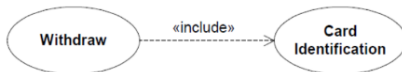
- ▶ Ивар Якобсон, 1992 год
- ▶ Акторы (или актёры, роли) — внешние сущности, использующие систему
  - ▶ Люди или другие программные системы
- ▶ Случаи использования (прецеденты) — цель использования системы актором
  - ▶ Раскрываются в набор сценариев, описываемых чаще текстом



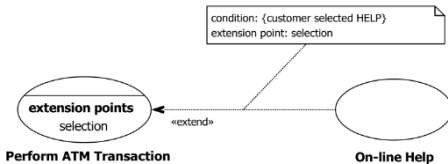
© М. Фаулер, UML. Основы

# Include и Extend

Include:

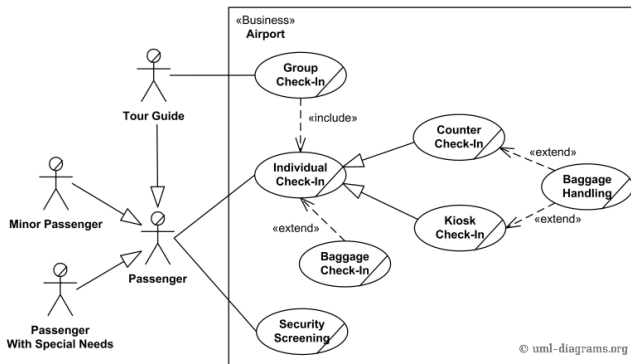


Extend:



© OMG, UML 2.5 Specification

# Пример, check-in в аэропорту



© <http://www.uml-diagrams.org>



## Сценарий использования, типичная структура

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

Use Case Name: Request a chemical	ID: UC-2	Priority: High
Actor: Lawn Chemical Applicator (LCA)		
Description: The Lawn Chemical Applicator (LCA) specifies the lawn chemical needed for a job by entering its name or ID number. The system satisfies the request by reserving the quantity requested or the quantity available and notifying the Chemical Supply Warehouse of the pick-up.		
Trigger: A Lawn Chemical Applicator (LCA) needs a chemical for a job.		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions: <ol style="list-style-type: none"> <li>1. The LCA identity is authenticated.</li> <li>2. The LCA has necessary training and credentials on file.</li> <li>3. The Chemical Supply datastore is up-to-date and on-line.</li> </ol>		
Normal Course: <ol style="list-style-type: none"> <li>1.0 Request a lawn chemical from the chemical supply warehouse.               <ol style="list-style-type: none"> <li>1. The LCA specifies a chemical needed and the quantity needed</li> <li>2. The system lists chemical and quantity on hand from Chemical Supply datastore                   <ol style="list-style-type: none"> <li>a. If the quantity on hand is less than the quantity needed, the LCA specifies the quantity he will take</li> <li>b. Purchasing is notified of chemical shortage</li> </ol> </li> <li>3. The system gives the LCA a Chemical Pick-up Authorization for the quantity requested</li> <li>4. The system notifies the Chemical Supply Warehouse of the chemical pick-up</li> <li>5. The system stores the Lawn Chemical Request in the Chemical Request datastore</li> </ol> </li> </ol>		
Postconditions: <ol style="list-style-type: none"> <li>1. The Lawn Chemical Request is stored in the Chemical Management System.</li> <li>2. The Chemical Pick-up Authorization is produced for the LCA.</li> <li>3. The Chemical Supply Warehouse is notified of the chemical pick-up.</li> <li>4. Purchasing is notified of chemical outage.</li> </ol>		
Exceptions: <p>E1: Chemical is no longer approved for use (occurs at step 1)</p> <ol style="list-style-type: none"> <li>1. The system displays message, "That chemical is no longer approved for use"</li> <li>2. The system asks the LCA if he wants to request another chemical or to exit               <ol style="list-style-type: none"> <li>3a. The LCA asks to request another chemical</li> <li>4a. The system starts Normal Course again</li> <li>3b. The LCA asks to exit</li> <li>4b. The system terminates the use case</li> </ol> </li> </ol>		

# 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

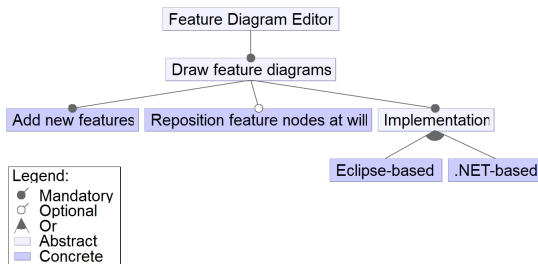


© <http://ecm-journal.ru>

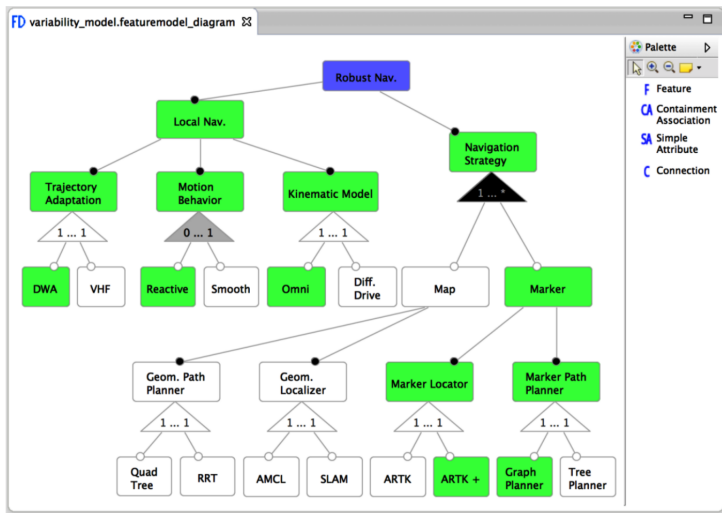
# Диаграмма характеристик

## Feature Diagram

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

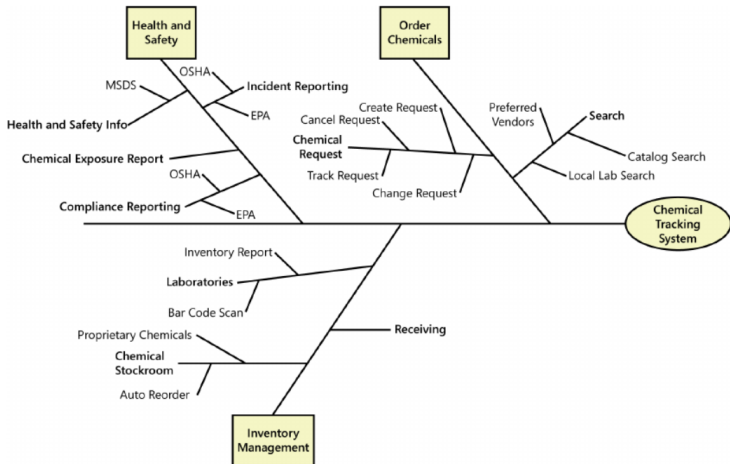


# Диаграмма характеристик, пример



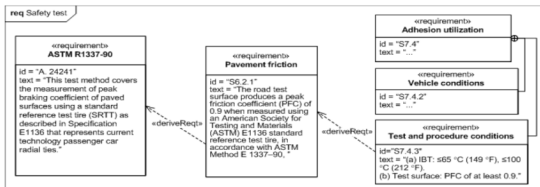
© D. Brugali

# Feature Tree



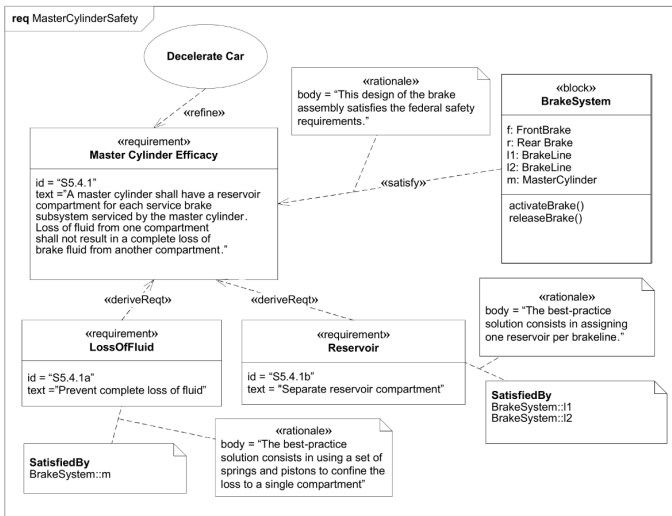
# Диаграмма требований, SysML

- ▶ Более формальная нотация дерева фич



© OMG SysML 1.4 Specification

# Диаграмма требований SysML, пример

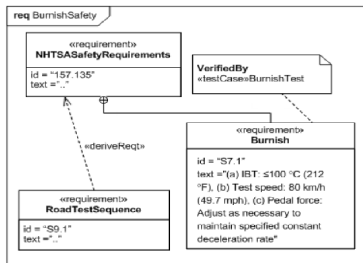


© OMG SysML 1.4 Specification

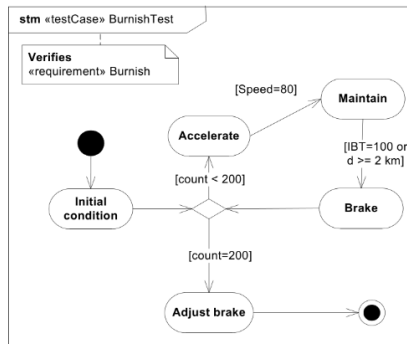


# Диаграмма требований SysML и тесты

## Требования:



## Сценарий тестирования:

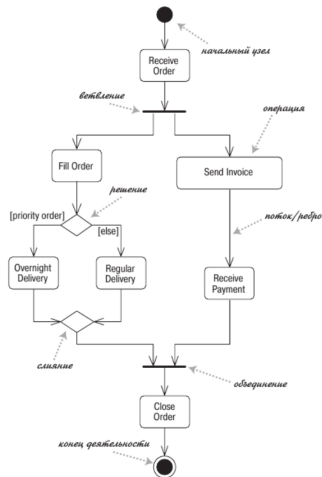


© OMG SysML 1.4 Specification

# Диаграмма активностей UML

## Диаграммы деятельности

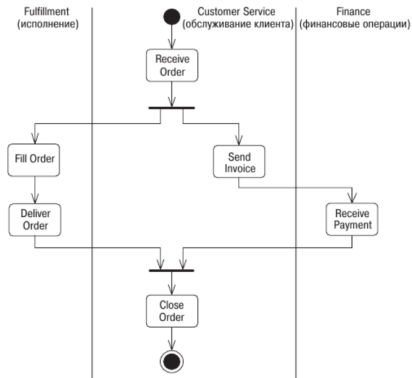
- ▶ Используются для моделирования бизнес-процессов, тоже на первых этапах
  - ▶ Может быть визуализацией сценария использования
- ▶ Иногда — для моделирования алгоритма
- ▶ Расширенные блок-схемы
- ▶ Семантика на основе сетей Петри



© М. Фаулер, UML. Основы

# Диаграмма активностей, разделы

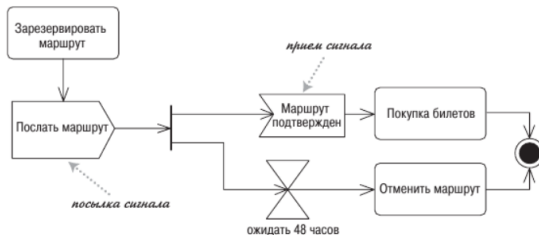
- ▶ Раздел представляет отдел организации (или организацию), отвечающий за часть работы
- ▶ Визуализирует поток работ между отделами



© М. Фаулер, UML. Основы

# Диаграмма активностей, сигналы

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

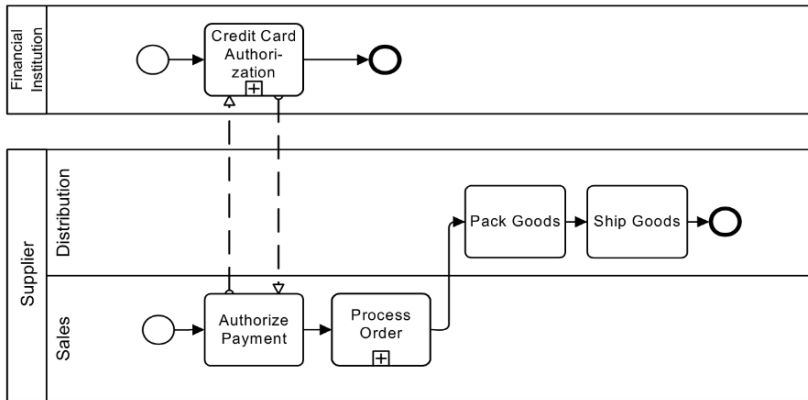


© М. Фаулер, UML. Основы

# 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

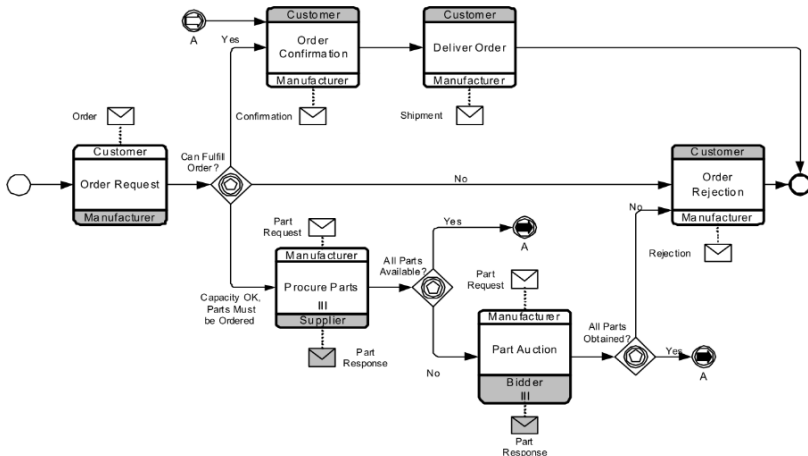


Сложный оператор

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

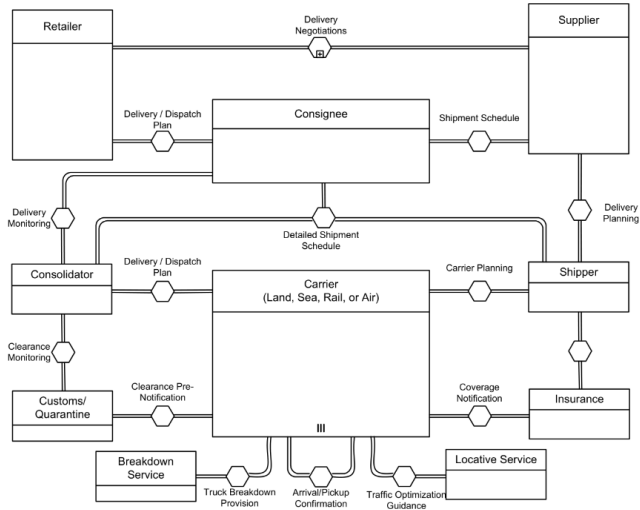


# Диаграмма хореографии



© OMG BPMN 2.0 Specification

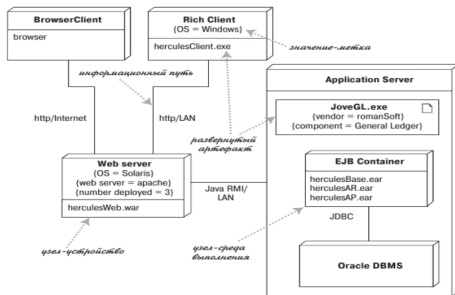
# Диаграмма диалогов



© OMG BPMN 2.0 Specification

# Диаграмма развёртывания UML

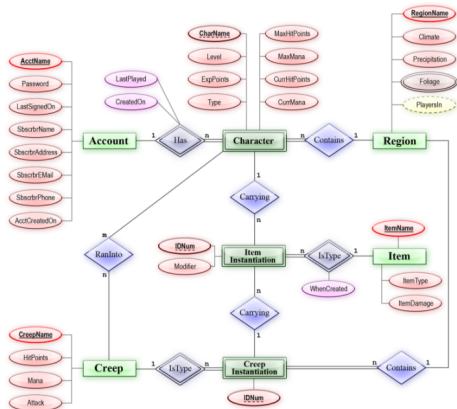
- ▶ Показывает отображение компонентов и физических артефактов на реальные (или виртуальные) устройства
- ▶ Бывает полезна на начальных этапах проектирования, даже до диаграмм компонентов



© М. Фаулер, UML. Основы

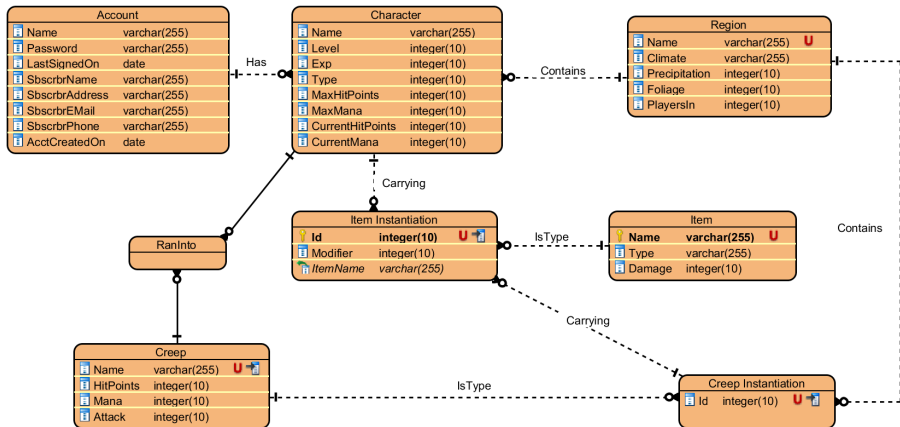
# Диаграммы “Сущность-связь”

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

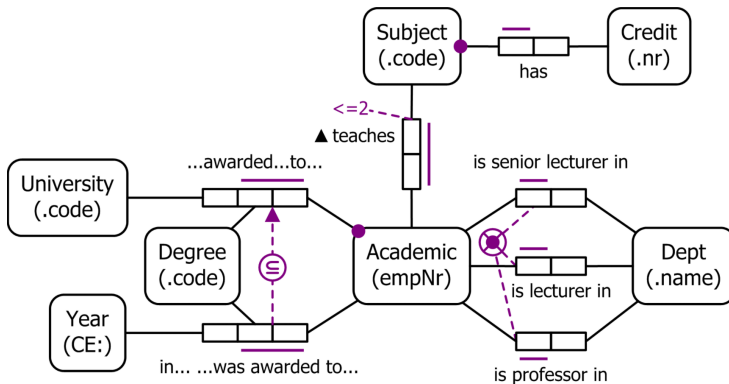


© <https://ru.wikipedia.org>

# Нотация “Вороньей лапки”



# Object-Role Modeling



© <http://www.orm.net>