

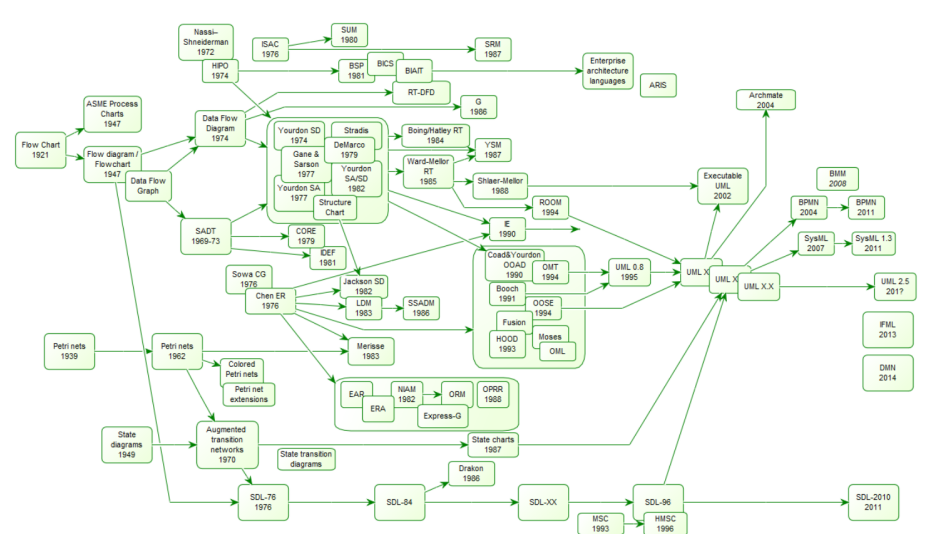
Обзор UML

Юрий Литвинов
yurii.litvinov@gmail.com

28.02.2019г

Unified Modeling Language

- ▶ Семейство графических нотаций
 - ▶ 14 видов диаграмм
- ▶ Общая метамодель
- ▶ Стандарт под управлением Object Management Group
 - ▶ UML 1.1 — 1997 год
 - ▶ UML 2.0 — 2005 год
 - ▶ UML 2.5.1 — декабрь 2017 года
- ▶ Прежде всего, для проектирования ПО
 - ▶ После UML 2.0 стали появляться нотации и для инженеров
- ▶ Расширяем
 - ▶ Профили — механизм легковесного расширения
 - ▶ Метамоделирование



Виды диаграмм

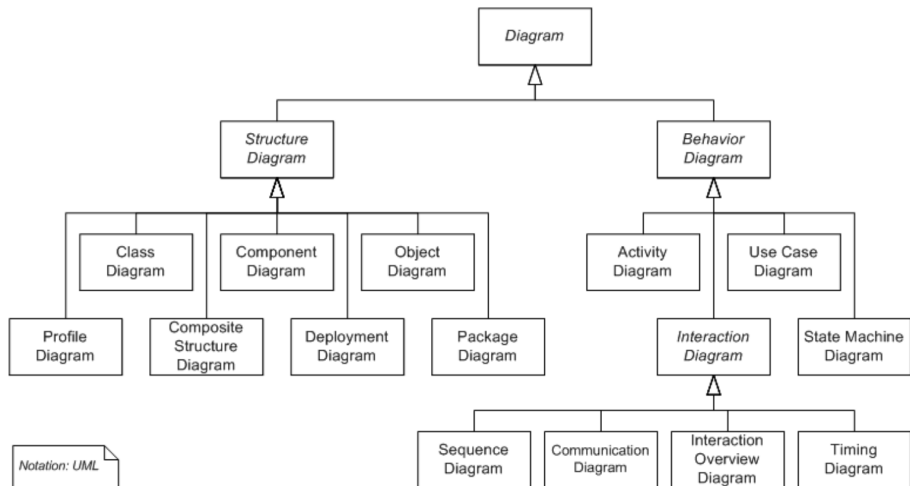
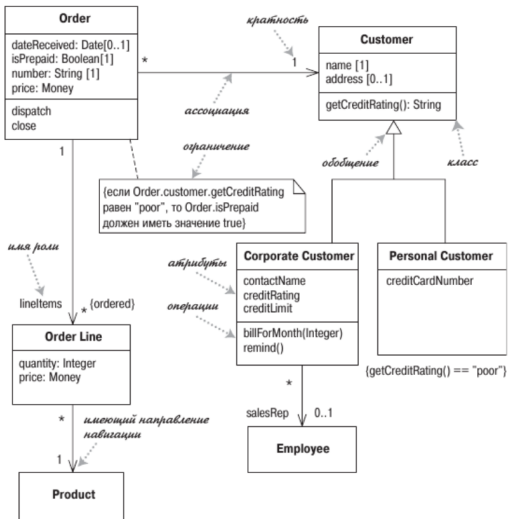
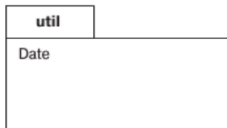
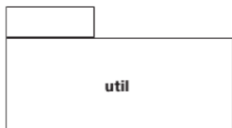


Диаграмма классов

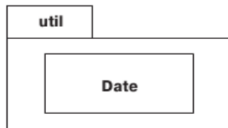


© М. Фаулер. "UML. Основы"

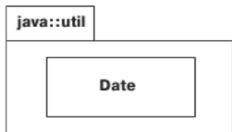
Диаграммы пакетов



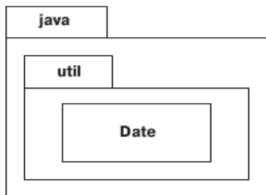
Содержимое, перечисленное
в прямоугольнике



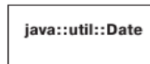
Содержимое в виде диаграммы
в прямоугольнике



Полностью определенное
имя пакета



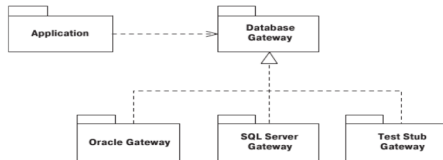
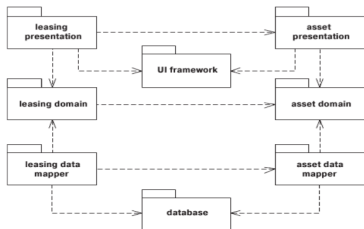
Вложенные пакеты



Полностью определенное
имя класса

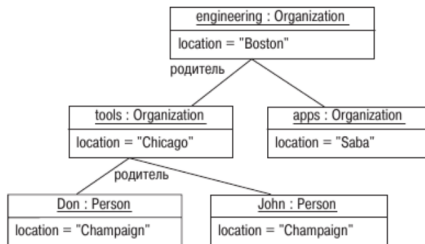
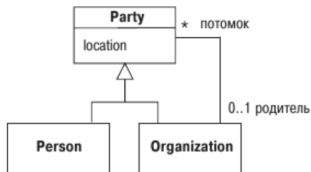
© М. Фаулер. “UML. Основы”

Диаграммы пакетов, зависимости



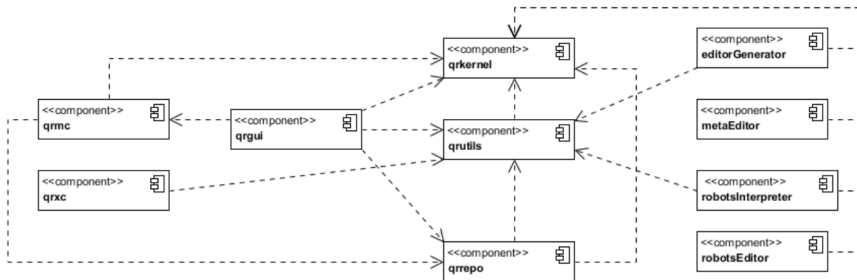
© М. Фаулер. "UML. Основы"

Диаграммы объектов

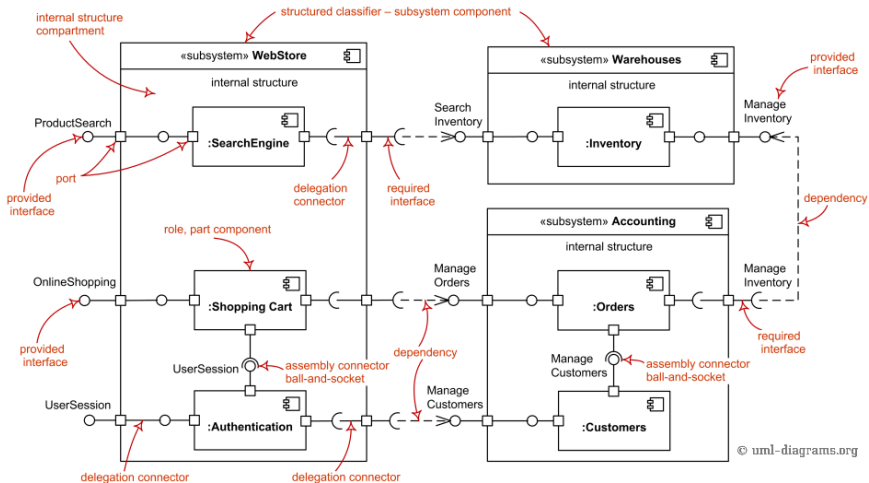


© М. Фаулер. "UML. Основы"

Диаграммы компонентов



Более подробно

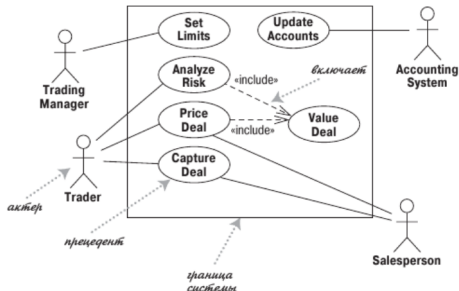


© <http://www.uml-diagrams.org>

Диаграмма случаев использования UML

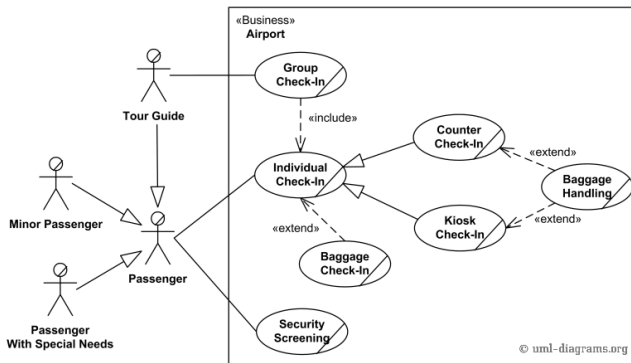
Диаграмма прецедентов

- ▶ 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

Пример, check-in в аэропорту



© uml-diagrams.org

© <http://www.uml-diagrams.org>

Случай использования, типичная структура

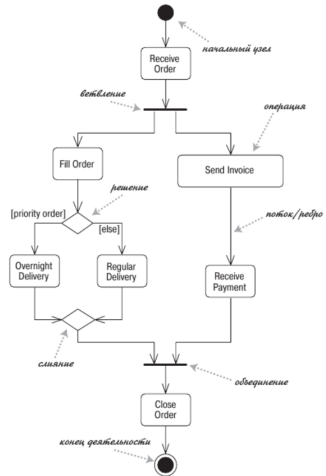
- ▶ Заголовок (цель основного актора)
- ▶ Заинтересованные лица, акторы, основной актор
- ▶ Предусловия
- ▶ Триггеры (активаторы)
- ▶ Основной порядок событий
- ▶ Альтернативные пути и расширения
- ▶ Постусловия

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"> 1. The LCA identity is authenticated. 2. The LCA has necessary training and credentials on file. 3. The Chemical Supply datastore is up-to-date and on-line. 		
Normal Course: <ol style="list-style-type: none"> 1.0 Request a lawn chemical from the chemical supply warehouse. <ol style="list-style-type: none"> 1. The LCA specifies a chemical needed and the quantity needed 2. The system lists chemical and quantity on hand from Chemical Supply datastore <ol style="list-style-type: none"> a. If the quantity on hand is less than the quantity needed, the LCA specifies the quantity he will take b. Purchasing is notified of chemical shortage 3. The system gives the LCA a Chemical Pick-up Authorization for the quantity requested 4. The system notifies the Chemical Supply Warehouse of the chemical pick-up 5. The system stores the Lawn Chemical Request in the Chemical Request datastore 		
Postconditions: <ol style="list-style-type: none"> 1. The Lawn Chemical Request is stored in the Chemical Management System. 2. The Chemical Pick-up Authorization is produced for the LCA. 3. The Chemical Supply Warehouse is notified of the chemical pick-up. 4. Purchasing is notified of chemical outage. 		
Exceptions: <p>E1: Chemical is no longer approved for use (occurs at step 1)</p> <ol style="list-style-type: none"> 1. The system displays message. "That chemical is no longer approved for use" 2. The system asks the LCA if he wants to request another chemical or to exit <ol style="list-style-type: none"> 3a. The LCA asks to request another chemical 4a. The system starts Normal Course again 3b. The LCA asks to exit 4b. The system terminates the use case 		

Диаграмма активностей UML

Диаграммы деятельности

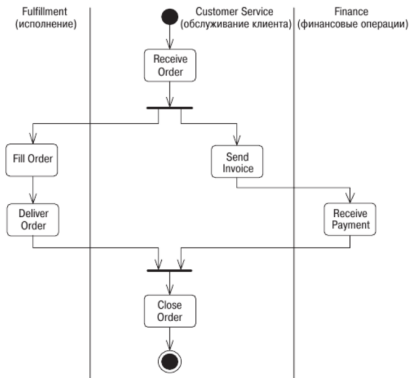
- ▶ Используются для моделирования бизнес-процессов, тоже на первых этапах
 - ▶ Может быть визуализацией сценария использования
- ▶ Иногда — для моделирования алгоритма
- ▶ Расширенные блок-схемы
- ▶ Семантика на основе сетей Петри



© М. Фаулер, UML. Основы

Диаграмма активностей, разделы

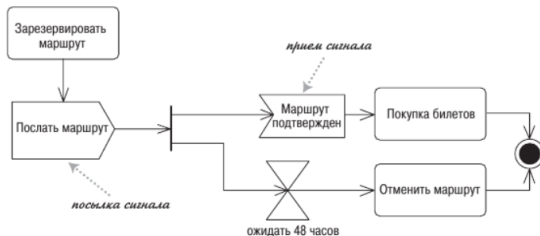
- ▶ Раздел представляет отдел организации (или организацию), отвечающий за часть работы
- ▶ Визуализирует поток работ между отделами



© М. Фаулер, UML. Основы

Диаграмма активностей, сигналы

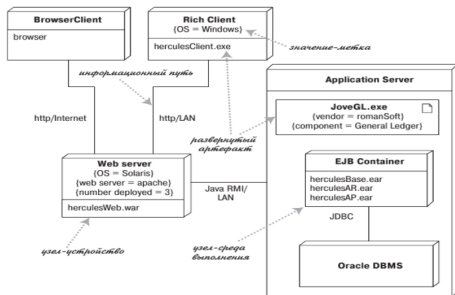
- ▶ Для визуализации асинхронных процессов
- ▶ Сигналом может быть посылка документа, запрос и т.д.



© М. Фаулер, UML. Основы

Диаграмма развёртывания UML

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

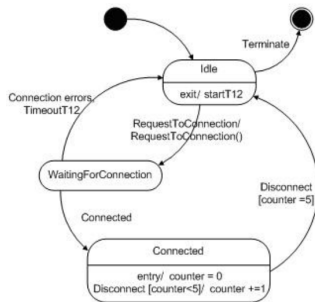


© М. Фаулер, UML. Основы

Диаграммы конечных автоматов

Диаграммы состояний

- ▶ Состояния объекта как часть жизненного цикла
- ▶ Моделирование реактивных объектов
 - ▶ Например, сетевое соединение
 - ▶ Или знакомый пример с торговым автоматом
- ▶ Имеют исполнимую семантику
- ▶ Д. Харел, 1987



Диаграммы конечных автоматов, синтаксис

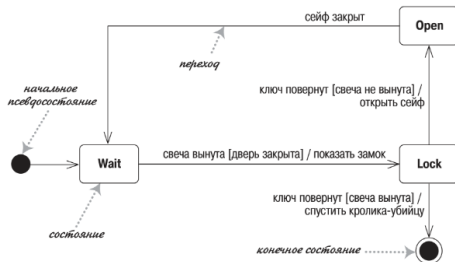
► Состояние

- entry activity
- exit activity
- do activity
- внутренний переход

► Событие

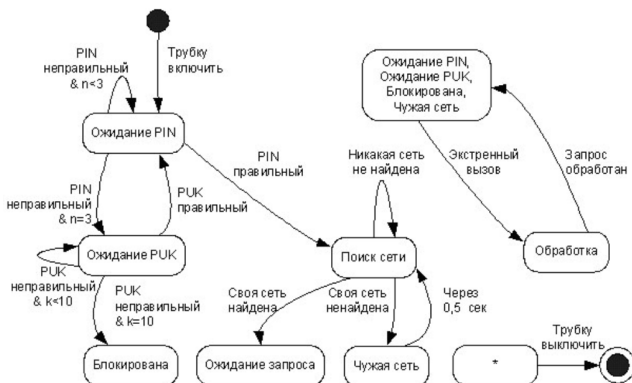
► Переход

- имя события (список параметров) [сторожевое условие] выражение действия



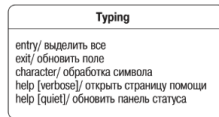
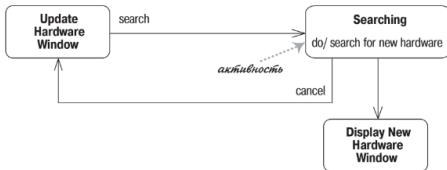
© М. Фаулер, UML. Основы

Пример, мобильный телефон

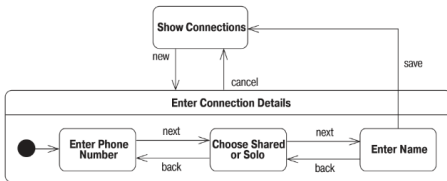


Диаграммы конечных автоматов, прочие вещи

Активности:

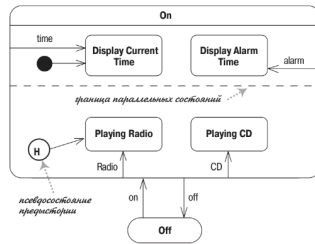


Вложенные состояния:



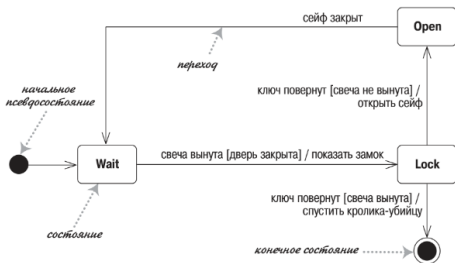
© М. Фаулер, UML. Основы

Параллельные состояния, псевдосостояние истории:



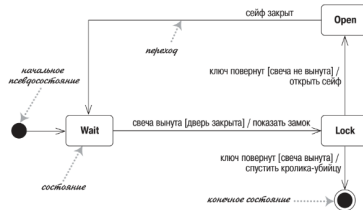
Генерация кода

```
public void handleEvent(PanelEvent anEvent) {
    switch (currentState) {
        case PanelState.Open:
            switch (anEvent) {
                case PanelEvent.SafeClosed:
                    currentState = PanelState.Wait;
            }
            break;
        case PanelState.Wait:
            switch (anEvent) {
                case PanelEvent.CandleRemoved:
                    if (isDoorOpen) {
                        revealLock();
                        currentState = PanelState.Lock;
                    }
            }
            break;
        case PanelState.Lock:
            switch (anEvent) {
                case PanelEvent.KeyTurned:
                    if (isCandleIn) {
                        openSafe();
                        currentState = PanelState.Open;
                    } else {
                        releaseKillerRabbit();
                        currentState = PanelState.Final;
                    }
            }
            break;
    }
}
```



© М. Фаулер, UML. Основы

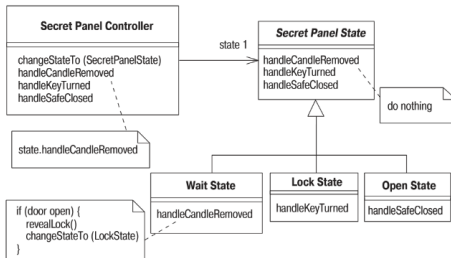
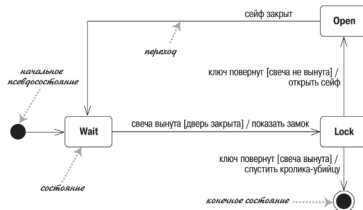
Таблица состояний



Исходное состояние	Целевое состояние	Событие	Защита	Процедура
Wait	Lock	Candle removed (свеча удалена)	Door open (дверца открыта)	Reveal lock (показать замок)
Lock	Open	Key turned (ключ повернут)	Candle in (свеча на месте)	Open safe (открыть сейф)
Lock	Final	Key turned (ключ повернут)	Candle out (свеча удалена)	Release killer rabbit (освободить убийцу-кролика)
Open	Wait	Safe closed (сейф закрыт)		

© М. Фаулер, UML. Основы

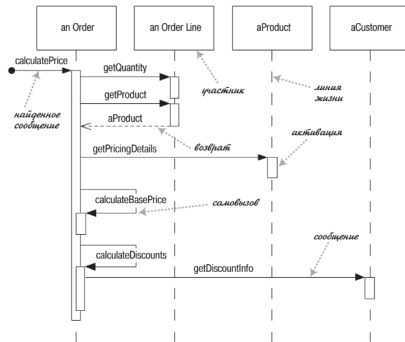
Паттерн “Состояние”



© М. Фаулер, UML. Основы

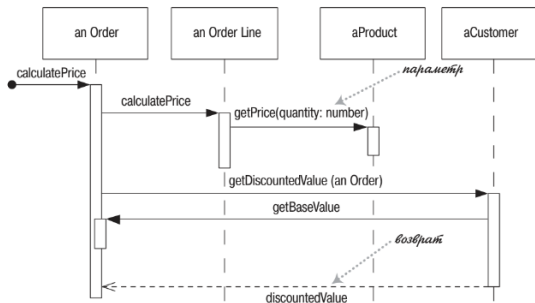
Диаграммы последовательностей

- ▶ Применяются для визуализации взаимодействия между объектами
 - ▶ Особо удобно для асинхронных вызовов
 - ▶ Телекоммуникационные протоколы
- ▶ Могут применяться на этапе анализа предметной области
- ▶ Могут применяться для составления плана тестирования
- ▶ И даже для визуализации логов работающей системы



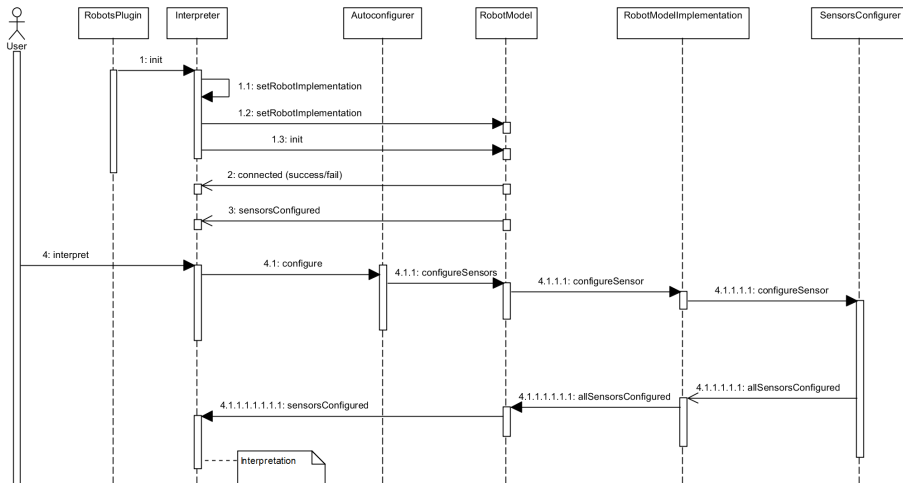
© М. Фаулер, UML. Основы

Ещё немного о синтаксисе

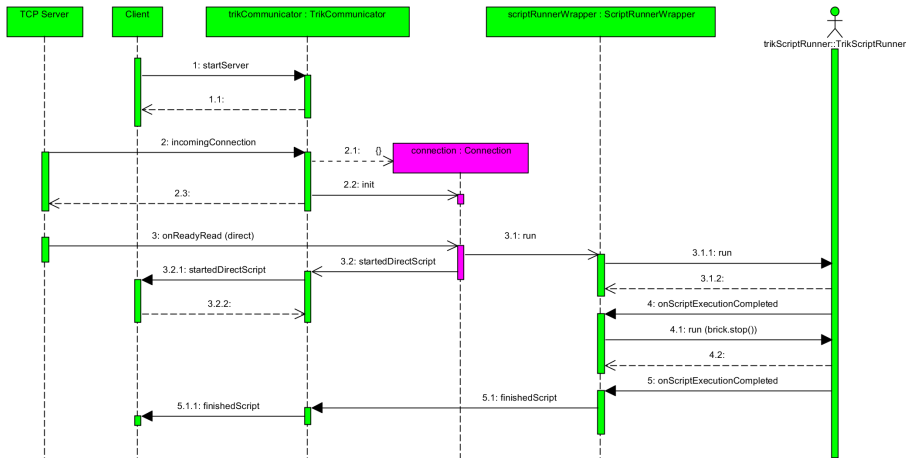


© М. Фаулер, UML. Основы

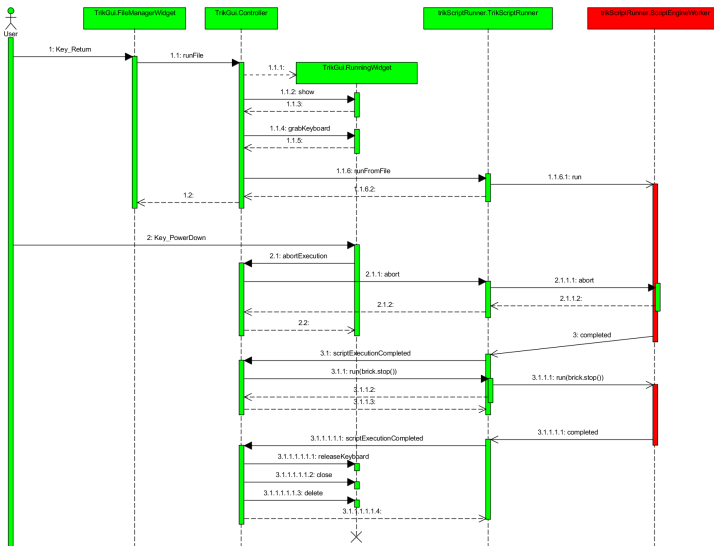
Пример



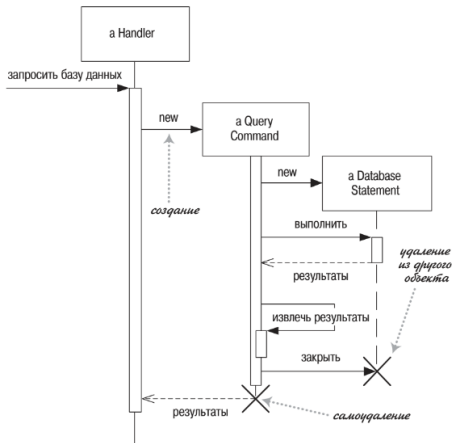
Ещё пример



И ещё пример



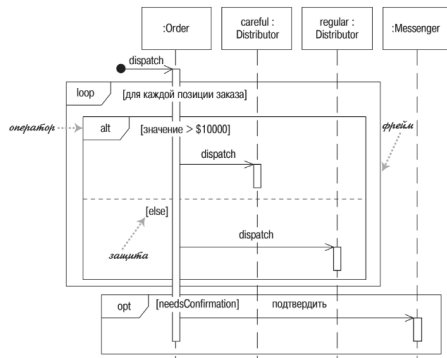
Создание и удаление объектов



© М. Фаулер, UML. Основы

Фреймы

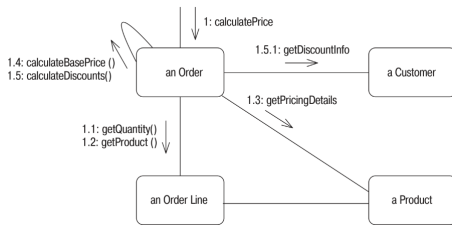
```
foreach (lineitem)
  if (product.value > $10K)
    careful.dispatch
  else
    regular.dispatch
  end if
end for
if (needsConfirmation)
  messenger.confirm
```



© М. Фаулер, UML. Основы

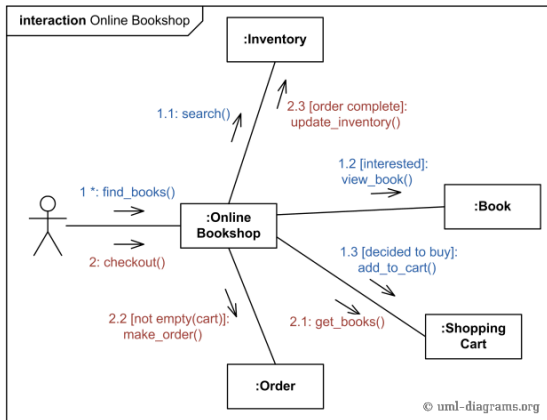
Коммуникационные диаграммы

- ▶ Применяются для визуализации взаимодействия между объектами
 - ▶ Более легковесный аналог последовательностей
 - ▶ Тоже отображают один сценарий взаимодействия



© М. Фаулер, UML. Основы

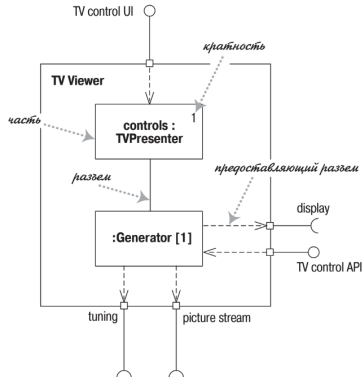
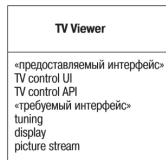
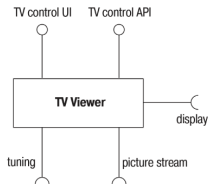
Коммуникационные диаграммы, пример



© <http://www.uml-diagrams.org/>

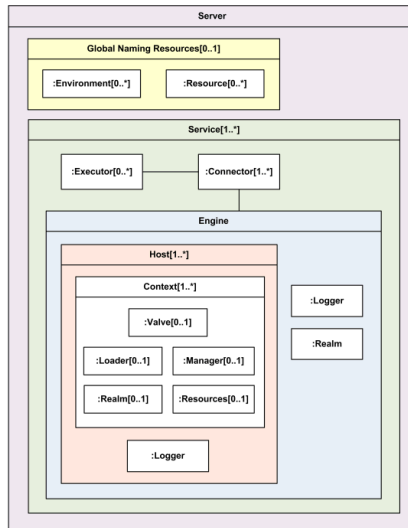
Диаграммы составных структур

- ▶ По сути, продвинутые диаграммы компонентов
- ▶ Внутри компоненты не другие компоненты, а части (роли)



© М. Фаулер, UML. Основы

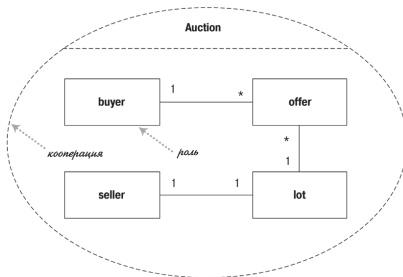
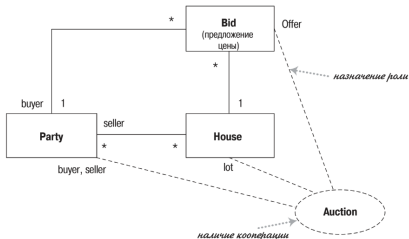
Диаграммы составных структур, пример



© <http://www.uml-diagrams.org/>

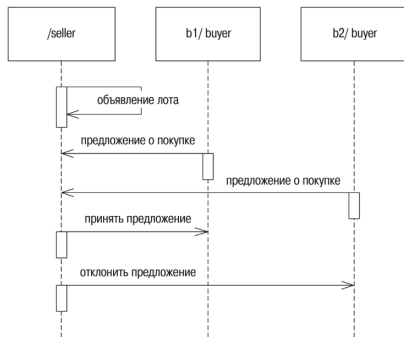
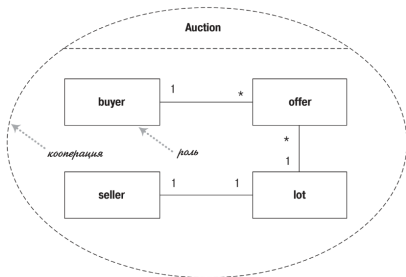
Диаграммы коопераций

- Показывают взаимодействие между объектами (ролями) в рамках одного сценария использования



© М. Фаулер, UML. Основы

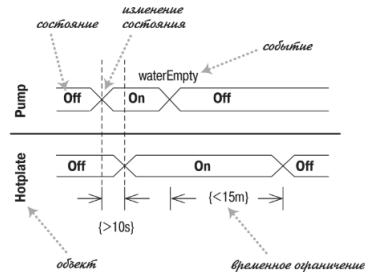
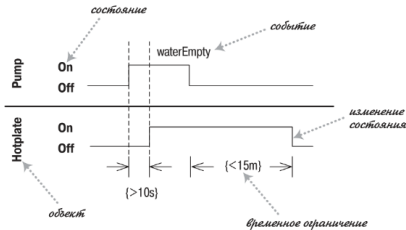
Диаграммы коопераций, последовательности



© М. Фаулер, UML. Основы

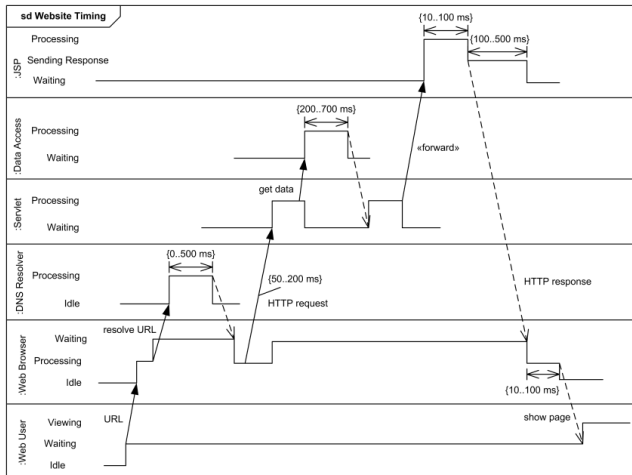
Временные диаграммы

- ▶ Для моделирования временных ограничений в системах реального времени



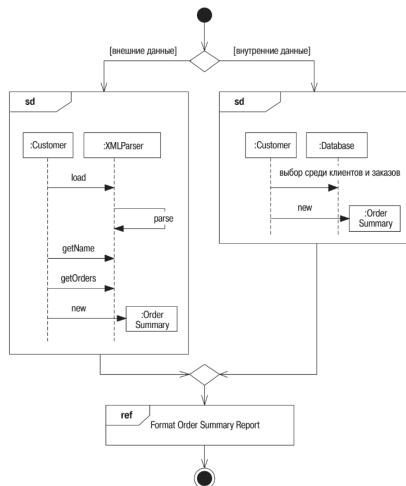
© М. Фаулер, UML. Основы

Временная диаграмма, пример

© <http://www.uml-diagrams.org/>

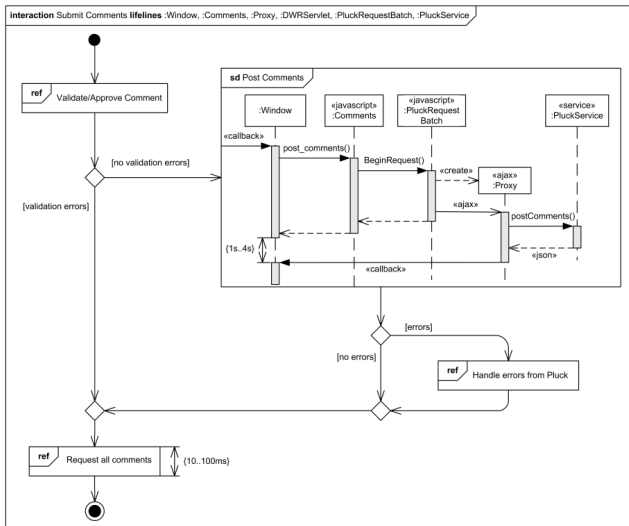
Диаграммы обзора взаимодействия

- ▶ Диаграммы активностей + диаграммы последовательностей
- ▶ Применяются при наличии взаимодействия со сложной логикой, когда фреймы неудобны



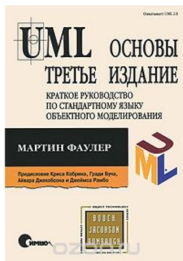
© М. Фаулер, UML. Основы

Диаграмма обзора взаимодействия, пример



© <http://www.uml-diagrams.org/>

Книжка



М. Фаулер, UML. Основы. Краткое руководство по стандартному языку объектного моделирования. СПб., Символ-Плюс, 2011. 192 С.

Домашнее задание, вариант 1

1. Нарисуйте с помощью диаграмм классов UML следующий фрагмент предметной области. Вот перечень учреждений, которые могут выдавать общегражданские загранпаспорта в РФ: в пределах РФ — органы МВД по месту жительства; МИД РФ и его представительства на территории РФ; вне пределов РФ — дипломатическими представительствами или консульскими учреждениями РФ.
2. Нарисуйте с помощью диаграмм классов UML следующий фрагмент предметной области. Есть лес, в нем растут деревья — сосны, березы, ивы. Березы бывают следующих видов: береза бумажная, береза вишневая, береза даурская. Сосны бывают следующих видов: сосна чешуйчатая, сосна уэмацу, сосна юньнаньская. У каждого дерева есть ствол, ветви, коневая система. Еще в лесу живут птицы — синицы, дрозды, совы (ушастая и болотная).
3. Нарисуйте с помощью диаграмм классов UML следующий фрагмент предметной области. «Учебный курс в университете».

Домашнее задание, вариант 2

1. Нарисуйте с помощью диаграмм классов UML следующий фрагмент предметной области. Органы власти в России подразделяются на государственные и муниципальные. Государственные органы власти бывают центральными и региональными, а также субъектов федерации. Субъекты бывают следующих видов — республики, края, области, города федерального значения, автономная область и автономные округа.
2. Нарисуйте с помощью диаграмм классов UML следующий фрагмент предметной области. Общегражданский загранпаспорт РФ бывает следующих видов — обычный и нового поколения. Владельцами загранпаспорта могут быть дети и взрослые (владельцы агрегируют свои загранпаспорта). Другими видами документов для выезда из РФ являются дипломатический паспорт, служебный паспорт, паспорт моряка.
3. Нарисуйте с помощью диаграмм классов UML фрагмент предметной области. «Сдача экзамена».

Домашнее задание, вариант 3

1. Нарисуйте с помощью диаграмм классов UML следующий фрагмент предметной области. Деревья в лесу бывают следующих видов — сосны, березы, ивы. Березы бывают следующих видов: береза бумажная, береза вишневая, береза даурская. Сосны бывают следующих видов: сосна чешуйчатая, сосна уэмацу, сосна юньнаньская.
2. Нарисуйте с помощью диаграмм классов UML следующий фрагмент предметной области. Банк состоит из различных филиалов, а также головного офиса. Все подразделения банка состоят из департаментов. Департаменты бывают производственными и административными. В департаментах работают сотрудники.
3. Нарисуйте с помощью диаграмм классов UML фрагмент предметной области «Поселение в общежитие».