

# Анализ и проектирование на UML

**Новиков Федор Александрович**

[fedornovikov@rambler.ru](mailto:fedornovikov@rambler.ru)

Курс подготовлен по заказу  
ООО Сан Майкросистемс СПб

Курс подготовлен при поддержке Sun Microsystems  
Правила использования материалов опубликованы на [www.sun.ru](http://www.sun.ru)

# План курса с упражнениями

- ✓ Введение в UML
  - Обзор языка
  - Моделирование использования
  - Моделирование структуры
  - Моделирование поведения
  - Управление моделями
  - UML и процесс разработки
- Действующие лица и варианты использования
- Классы и отношения между ними
- Автоматы, блок-схемы и кооперации
- Итоговый семинар

# Цели и задачи курса

- **Ясное понимание основных идей UML**
- **Знание базовых конструкций UML 1.x + UML 2.0**
- **Приобретение практических навыков моделирования на UML**
- **Знакомство с конкретным инструментом моделирования (**Sun Java Studio Enterprise 8**)**
- **Осознание тенденций развития UML**

# Проверяемые результаты курса

- **Индивидуально созданный артефакт – синтаксически правильные модели использования, структуры и поведения реального приложения**
- **Содержательная презентация модели с использованием технических средств продолжительностью полчаса**

# **Анализ и проектирование на UML**

## **Часть 1**

# План лекций

- ✓ **Введение в UML**
  - **Обзор языка**
  - **Моделирование использования**
  - **Моделирование структуры**
  - **Моделирование поведения**
  - **Управление моделями**
  - **Тенденции развития языка**
  - **UML и процесс разработки**

# 1. Введение в UML

- 1.1. Что такое UML?
- 1.2. Назначение UML
- 1.3. Способы использования языка
- 1.4. Структура определения языка
- 1.5. Терминология, литература, нотация

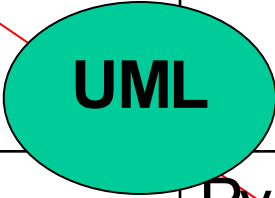
# 1.1. Что такое UML?

- **UML = Unified Modeling Language**
- **Язык**
  - Нотация
  - Семантика
  - Прагматика
- **Моделирования**
  - Modeling vs. Simulation
- **Унифицированный**
  - Универсальный? Единый?



# UML – это язык

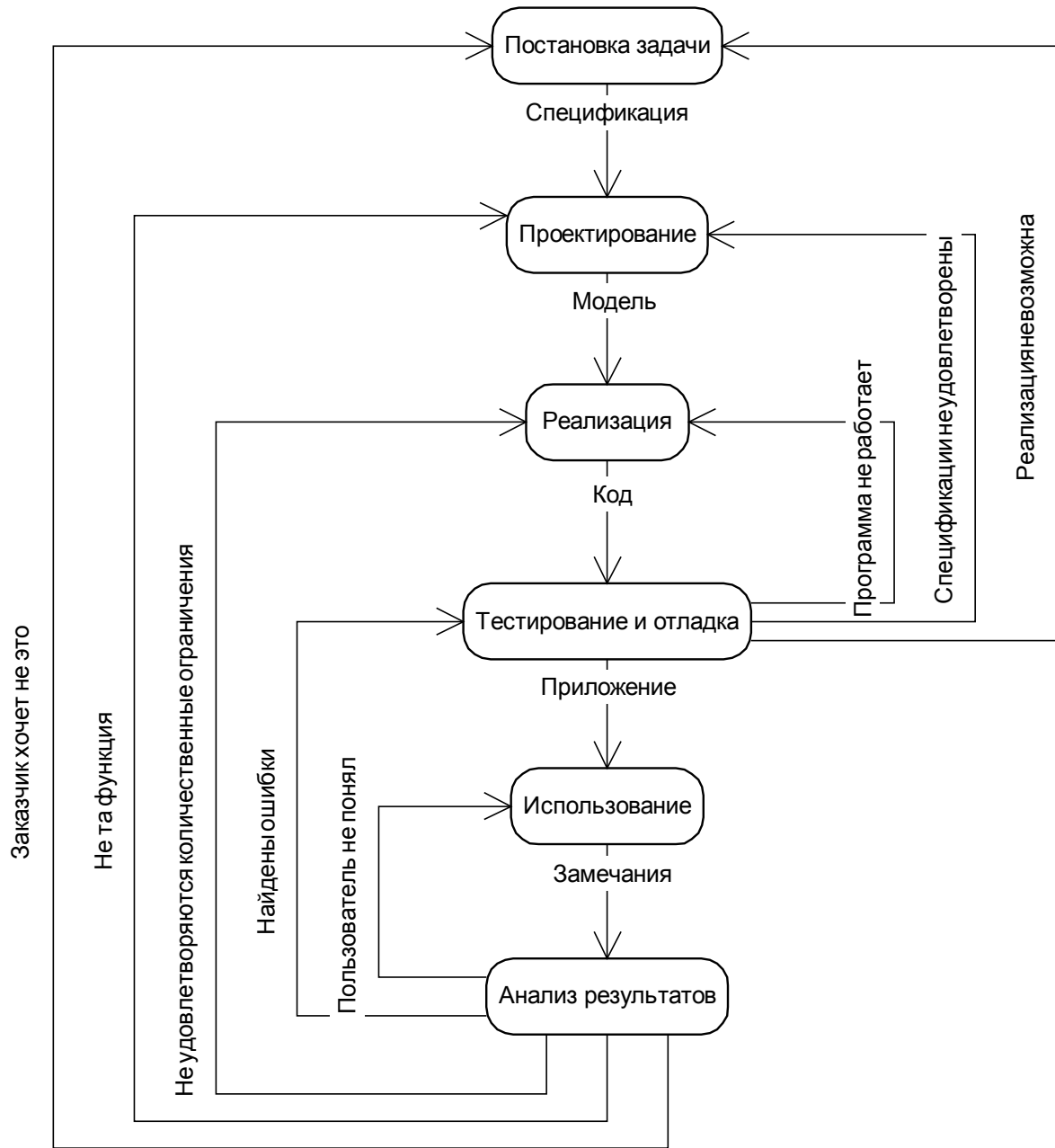
	Формальный	Неформальный
Искусственный	Паскаль	Эсперанто
Естественный	Математика	Русский



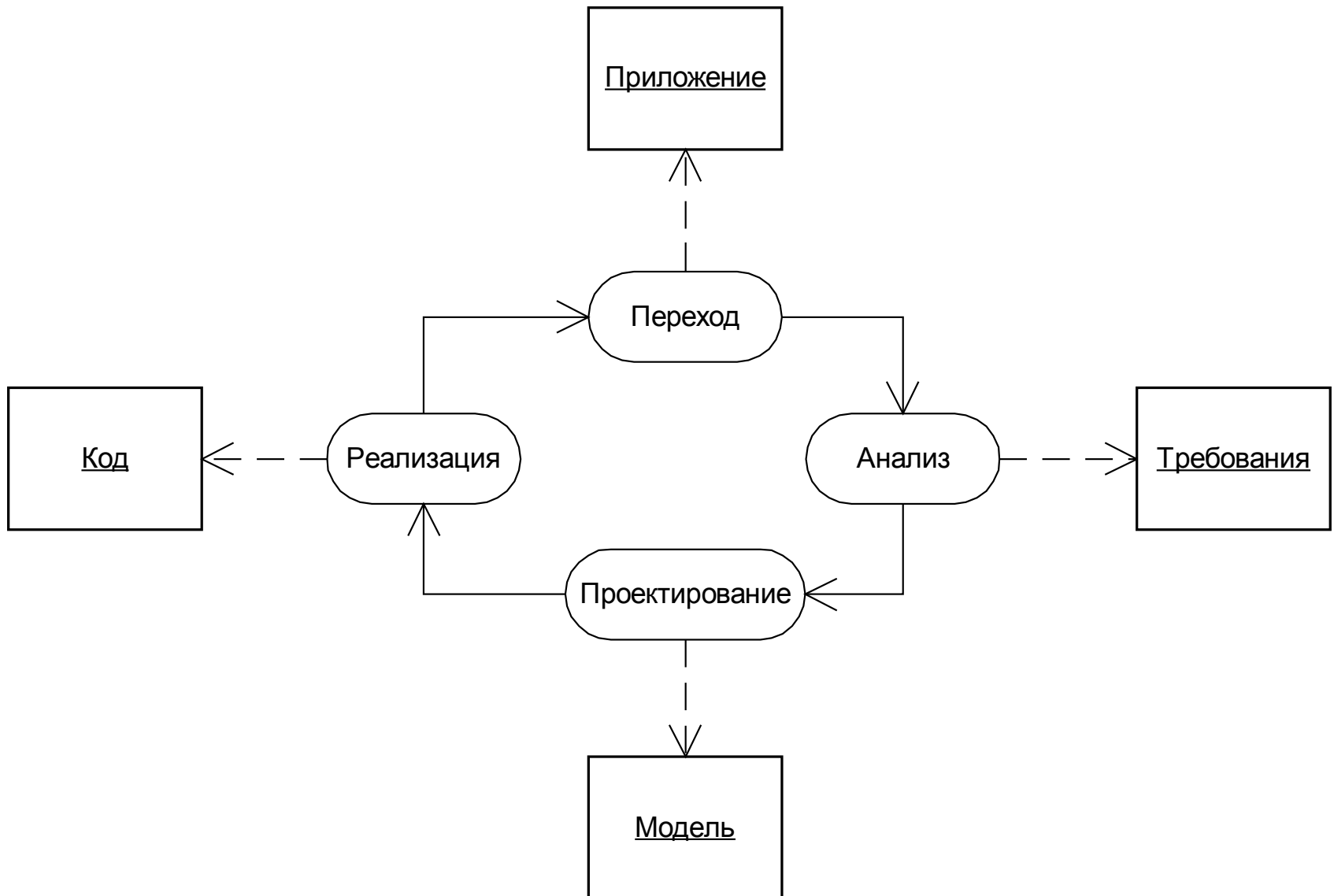
# **UML – это язык моделирования**

- **Жизненный цикл приложения**
- **Модель процесса разработки**
- **Модель – главный артефакт фазы проектирования итеративного процесса разработки**
- **Моделирование – составление и использование моделей**

# Жизненный цикл приложения



# Итеративный процесс разработки



# **UML – это унифицированный язык моделирования**

- **Исторический контекст**
- **Развитие путем объединения и унификации**
- **Авторы и международное сообщество**
- **Современные тенденции**

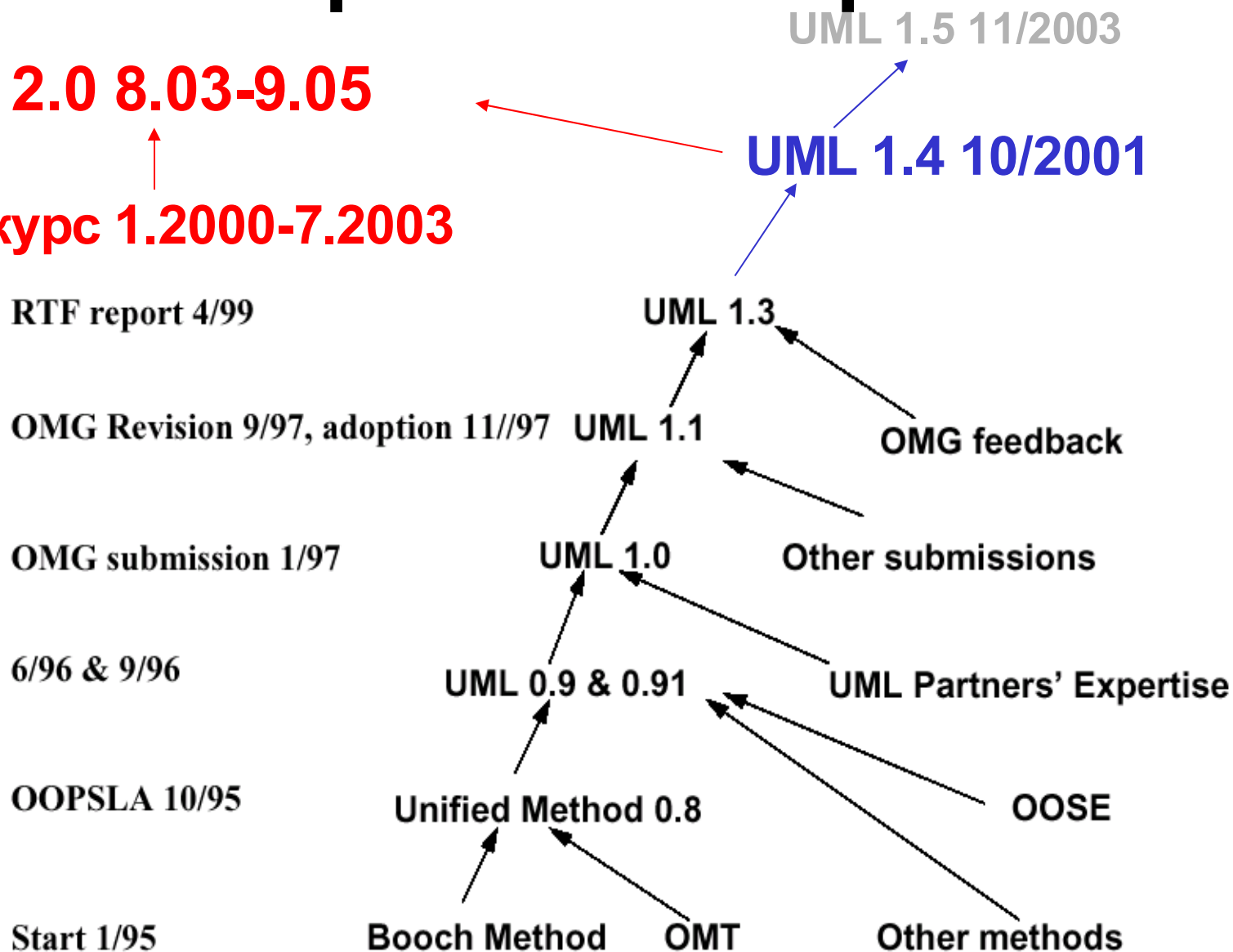
# Исторический контекст

- ...
- Петроглифы
- Блок-схемы
- Р-технология
- Диаграммы потоков данных (DFD)
- Диаграммы «сущность-связь» (ERD)
- Методология структурного анализа и проектирования (SADT)
- ...

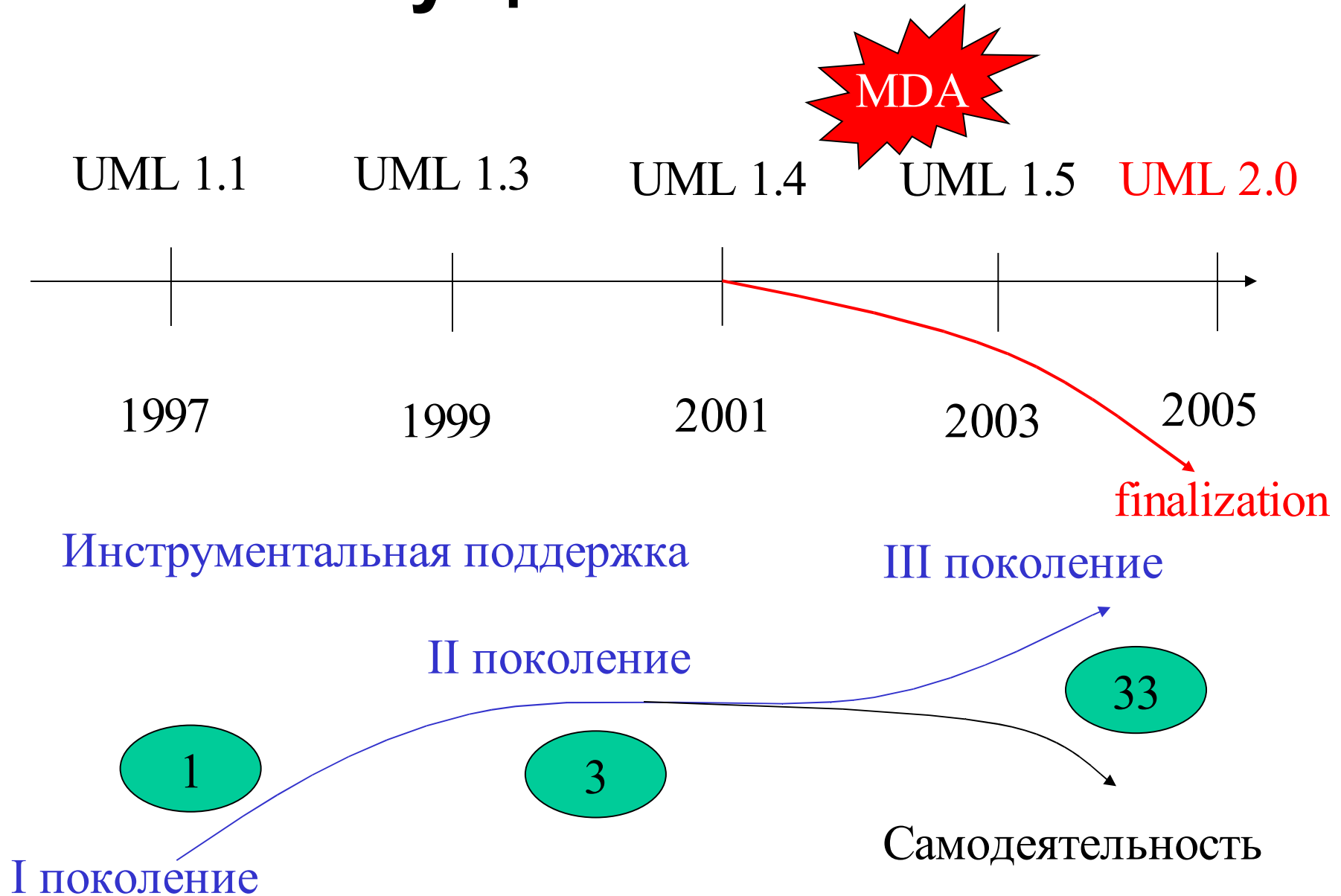
# Историческая справка

**UML 2.0 8.03-9.05**

**Конкурс 1.2000-7.2003**

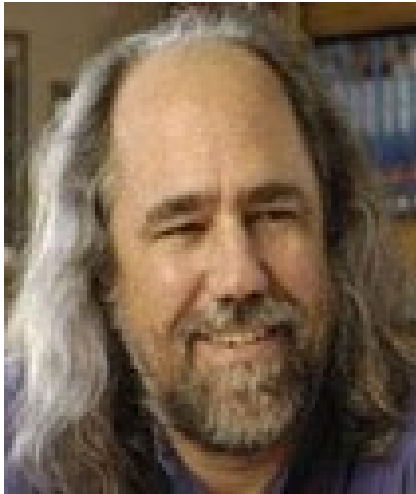


# Текущее состояние





# Авторы UML



**Grady Booch**

**Грэди Буч**



**James Rumbaugh**

**Джеймс Рамбо**



**Ivar Jacobson**

**Айвар Якобсон**

# 1.2. Назначение UML

**Моделирование =**

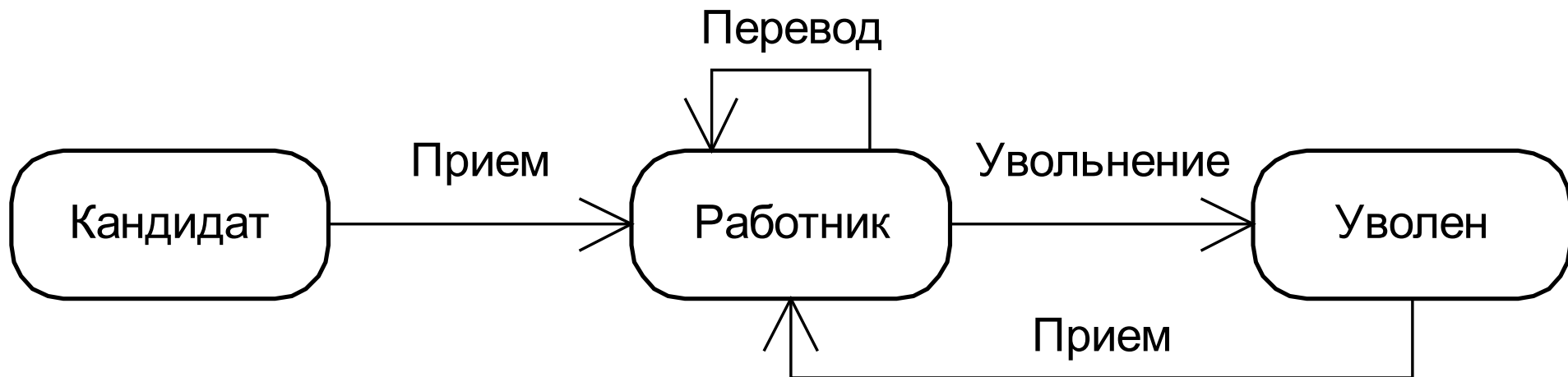
- **+ Спецификация (specification)**
  - формальная (но наглядная)
- **+ Визуализация (visualization)**
  - для общения
- **+ Проектирование (construction)**
  - архитектуры (конструирование?)
- **+ Документирование (documenting)**
  - всех артефактов

# Формальные спецификации

- **Спецификация = описание (программы)**
  - В понимании заказчика  $\neq$
  - В понимании разработчика  $\neq$
  - На самом деле
- **Полная формальная (математическая) спецификация возможна, но...**
  - Объем спецификации больше объема кода
  - Математики дороже программистов
  - Наилучшая спецификация программы – код
  - Формализация дороже автоматизации
- **Формализованная спецификация**

# Визуализация

- Сухой текст < текст с картинками < картинки с текстом = комиксы = UML



# Проектирование (construction)

- Автоматический синтез программ алгоритмически неразрешим (как массовая проблема)
  - Известны разрешимые подклассы
- Частичная генерация кода возможна
  - Опора на стереотипные компоненты
  - Уменьшение количества ошибок
- Reverse engineering (???обратное проектирование) !инженерный анализ программ
  - не путать с декомпиляцией
  - восстановление модели по коду
  - round-trip engineering  $\approx$  construction + reverse engineering

# Документирование

- Все элементы моделей UML могут содержать текстовые описания
- Почти все инструменты умеют собирать из них осмысленные документы
- Почти никто из программистов не пишет текстов

# 1.3. Способы использования языка




- Рисование картинок **+ 3**
- Обмен информацией **+ 2**
- Спецификация систем **+ 1**
- Повторное использование архитектурных решений **0**
- Генерация кода **- 1**
- Simulation **- 2**
- Верификация моделей **- 3**

# Инструментальная поддержка





# Чем НЕ является UML

- **Языком программирования**
  - хотя генерация кода не возбраняется
- **Спецификацией инструмента (CASE)**
  - хотя инструменты подразумеваются и имеются (Sun  , Together  , Rose, Visio  , Argo, ...)
- **Моделью процесса**
  - хотя модель необходима и имеется (Rational Unified Process → Unified Software Development Process)

# 1.4. Структура определения языка

- Точность vs. Понятность: компромисс
- Четыре уровня моделирования
  - мета метамодель      КС грамматика
  - метамодель              язык
  - модель                    программа
  - объекты                  выполнение
- Раскрытие
  - Семантика = Диаграммы классов (abstract syntax) + Ограничения (OCL) + Текст (English)
  - Нотация = Отображение семантики в картинки (а не наоборот!)
  - OCL = ОО язык первого порядка с равенством и ограниченными кванторами

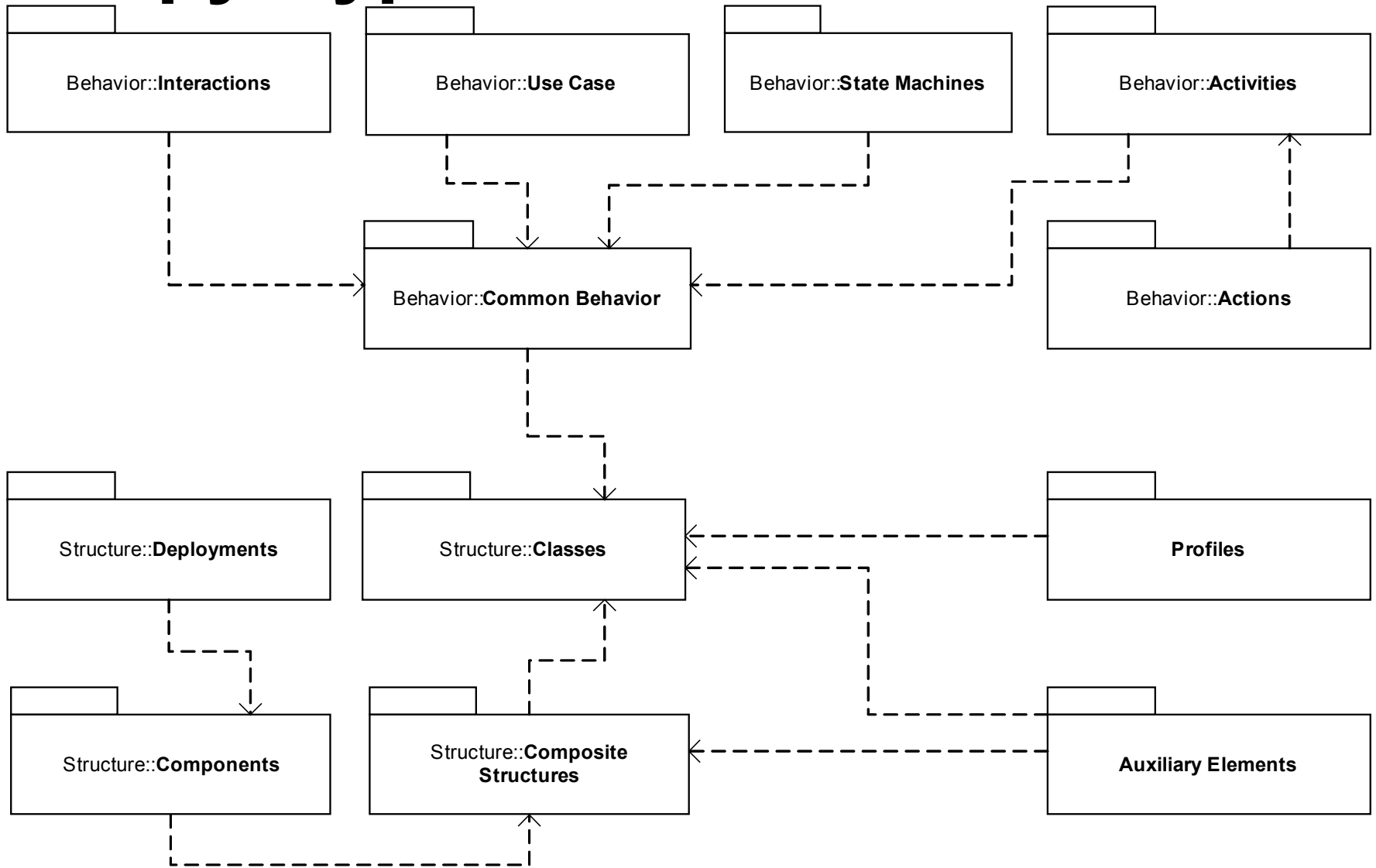
# OMG UML Specification 1.4

■ UML Summary	14
■ UML Semantics	183
■ UML Notation Guide	166
■ UML Extensions	14
■ UML CORBA	155
■ UML XMI DTD	362
■ OCL	52
■ Glossary	24
■ всего страниц	≈1000

# OMG UML Specification 2.0

- UML 2.0 Superstructure 640
- UML 2.0 Infrastructure 199
- UML 2.0 Diagram Interchange 034
- UML Extensions ...
- UML MOF Metamodel ...
- UML CORBA ...
- UML XMI DTD ...
- OCL ...
- всего страниц **ОЧЕНЬ МНОГО**

# Структура метамодели UML 2.0



# Способ объяснения понятий

- Теория – откуда это взялось
  - Прагматика – зачем это нужно
  - Семантика – на сквозном примере
  - Нотация – все элементы по разу
  - Метамодель – упрощенный абстрактный синтаксис
  - Упражнение – по принципу 80%:20%
- 20% языка достаточно для 80% случаев**

# 1.5. Терминология

- **Перевод терминов UML на русский**
  - **А.М. Вендров (UML Distilled)**
    - Неплохо для начала
  - **А.А. Слинкин (UML User Guide)**
    - Переведено по словарю
  - **А.В. Леоненков (Самоучитель)**
    - Средне
  - **К. и А. Максимовы (Справочник)**
    - Редактор Ф.А. Новиков
  - **Д. Солнышков (UML 2)**
    - Не успел повлиять ☹️

# Доступная и рекомендуемая литература

- UML Language Reference (Three Amigos)
  - Специальный справочник (Питер, 2002)
  - **UML 2-е издание (Питер, 2006)**
- UML Distilled (**Martin Fowler** & Kendall Scott)
  - Основы UML второе издание (Символ, 2002)
  - В кратком изложении (Мир, 1999)
- UML User Guide (Three Amigos)
  - Руководство пользователя (ДМК, 2000, Питер 2003)
- Самоучитель (Леоненков)
  - (БХВ, 2001, 2004)






# Нотация

## ■ Нотация

- Фигуры – двумерные с внутренностью
- Линии – одномерные с наружностью
- Значки – двумерные с наружностью
- Тексты
- **Рамки (UML 2.0)**

## ■ Инструменты рисования

- Sun (используется в этих слайдах) 
- Visio (отлично РИСУЕТ – не мешает) 
- Together (красиво РИСУЕТ – иногда слишком) 
- Rose (хорошо РИСУЕТ – старается)
- Argo UML (неплохо РИСУЕТ, но устает и падает)

## ■ Стандартный стиль и вариации

- Черно-белый без излишеств для печати
- **Синий – мой комментарий**

# Выводы

- **UML – еще один формальный язык, который необходимо быстро освоить**
- **Знание UML является необходимым, но не является достаточным условием построения разумных моделей**
- **UML имеет синтаксис, семантику и прагматику, которые нужно использовать с учетом особенностей фактического инструмента**