

重庆大学本科课程

面向对象技术与 UML

Object-Oriented Technique and UML



重庆大学软件学院

教材目录 《面向对象技术UML教程》

第 1 章 面向对象技术概述

第 2 章 [UML概述](#)

第 3 章 用例和用例图

第 4 章 顺序图和协作图

第 5 章 类图和对象图

第 6 章 数据建模

第 7 章 包

第 8 章 状态图和活动图

第 9 章 构件图

第 10 章 部署图

第 11 章 对象约束语言

第 12 章 业务建模

第 13 章 Web建模

第 14 章 UML与设计模式

第 15 章 面向对象实现技术

第 16 章 RUP 软件开发工程

第 17 章 UML开发工具

第 18 章 实例应用分析



第2章 UML概述

- 2.1 [为什么要学习UML?](#)
- 2.2 [UML的历史](#)
- 2.3 [UML的特点](#)
- 2.4 [UML的构成](#)
- 2.5 [UML中的视图](#)
- 2.6 [UML在系统开发各阶段的应用](#)



第1章 面向对象技术概述

- 本章需要掌握的内容：
 - 什么UML？它的英文全称？
 - 了解UML的发展历史
 - UML的重点图（9种）
 - UML的“4+1”视图指的是什么？



UML概述

◇ 什么是UML?

UML是**U**nified **M**odeling **L**anguage（统一建模语言）的简称。

UML是对软件密集型系统中的制品（软件开发过程中产生的各种各样的产物，如模型、源代码、测试用例等）进行可视化、详述、构造和文档化的语言。



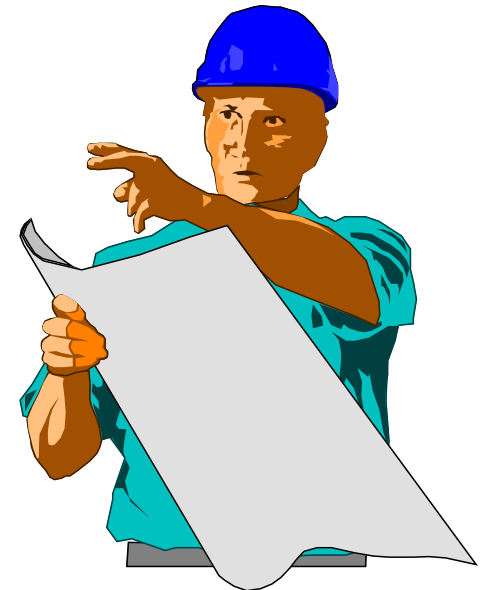
Visualizing - *a picture is worth a thousand words; a graphical notation articulates and unambiguously communicates the overall view of the system (problem-domain).*

“一幅图顶得上一千个字”

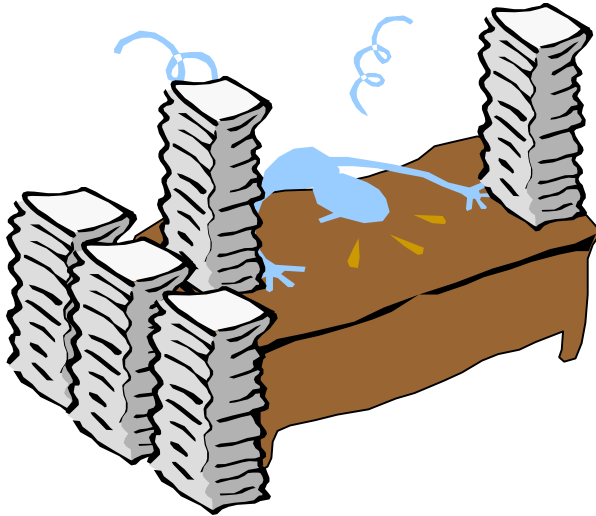


Specifying - *UML provides the means to model precisely, unambiguously and completely, the system in question.*

Constructing - *models built with UML have a “design” dimension to it; these are language independent and can be implemented in any programming language.*



Documenting - every software project involves a lot of documentation - from the inception (初期) phase to the deliverables (交付) .



Documentation is (among others) for:

- Requirements
- Design
- Tests

UML provides the notations (符号) for documenting some of these artifacts

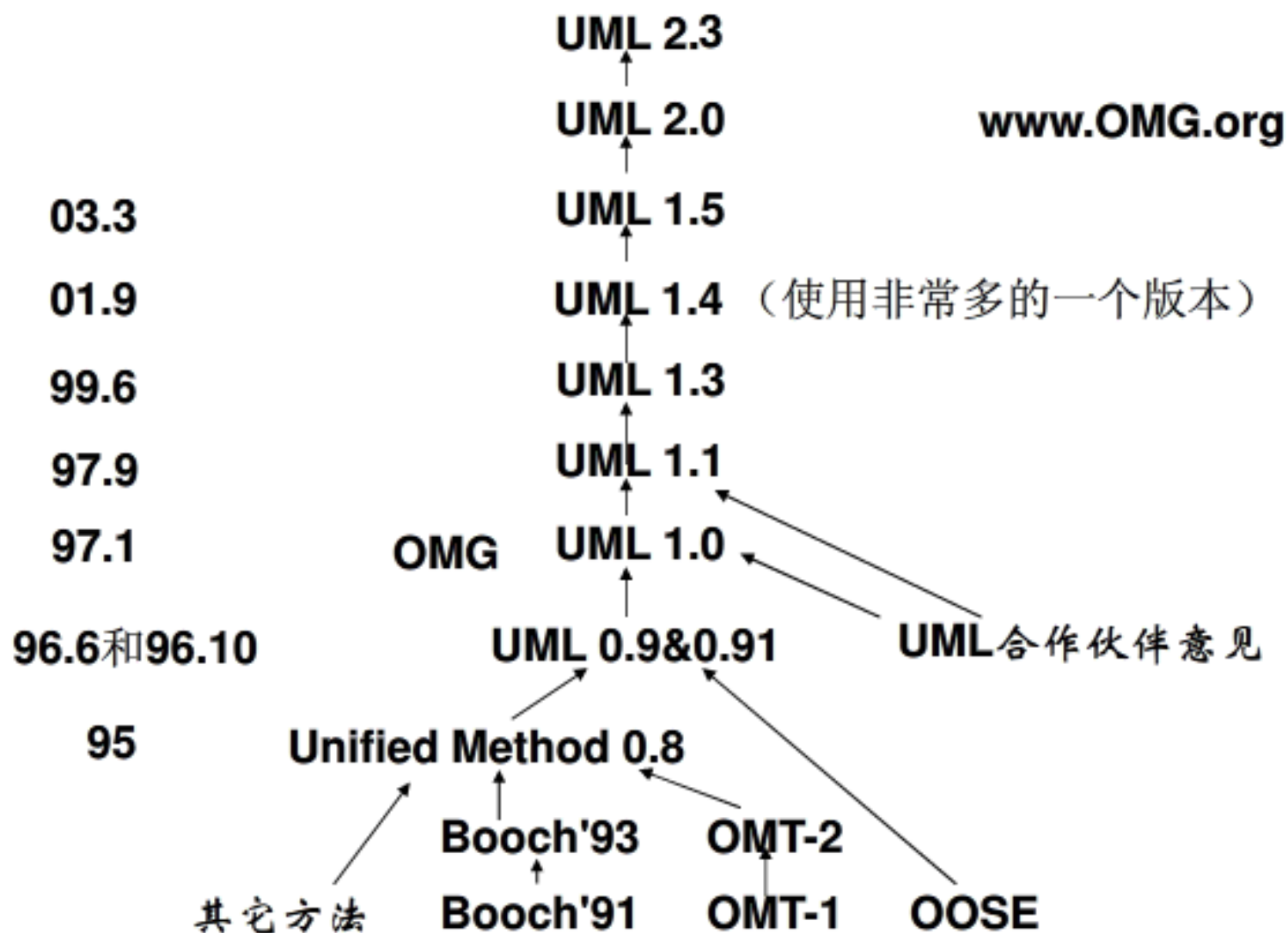
为什么要学习UML

◇ 为什么要建立模型

- * 使用模型可以更好地理解问题 (To understand the problem better)
- * 使用模型可以加强人员之间的沟通 (To communicate with other persons)
- * 使用模型可以更早地发现错误或疏漏的地方 (To find errors or omissions)
- * 使用模型可以获取设计结果 (To plan out the design)
- * 模型为最后的代码生成提供依据 (To generate code)



UML的历史



UML的历史

九十年代中，面向对象方法已经成为软件分析和设计方法的主流。

1994年10月Jim Rumbaugh和Grady Booch共同合作把他们的OMT和Booch方法统一起来，到1995年成为“统一方法”（Unified Method）版本0.8。随后，Ivar Jacobson加入，并采用他的用例(User case)思想,到1996年，成为“统一建模语言”版本0.9。

1997年1月，UML版本1.0被提交给OMG组织，作为软件建模语言标准的候选。其后的半年多时间里，一些重要的软件开发商和系统集成商都成为“UML伙伴”，如IBM，Microsoft，HP等.1997年11月7日被正式采纳作为业界标准。

UML创始人



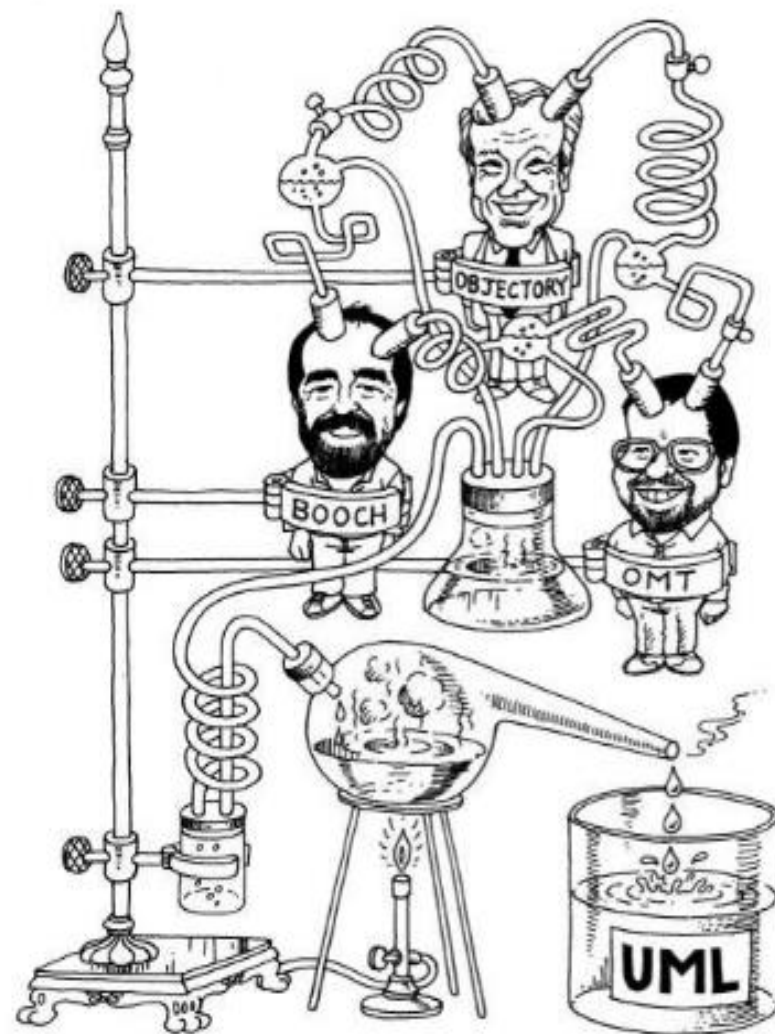
Jim Rumbaugh



Grady Booch



Ivar Jacobson



UML的历史

◇ 为什么UML能得到广泛的应用

- * 图形化的建模语言
- * 开发者用来为面向对象系统建立模型
- * 具有灵活性与可扩展性

◇ 由Object Management Group (OMG)推荐成为国际标准

目前最新的UML规范说明（<http://uml.org>）



UML的特点

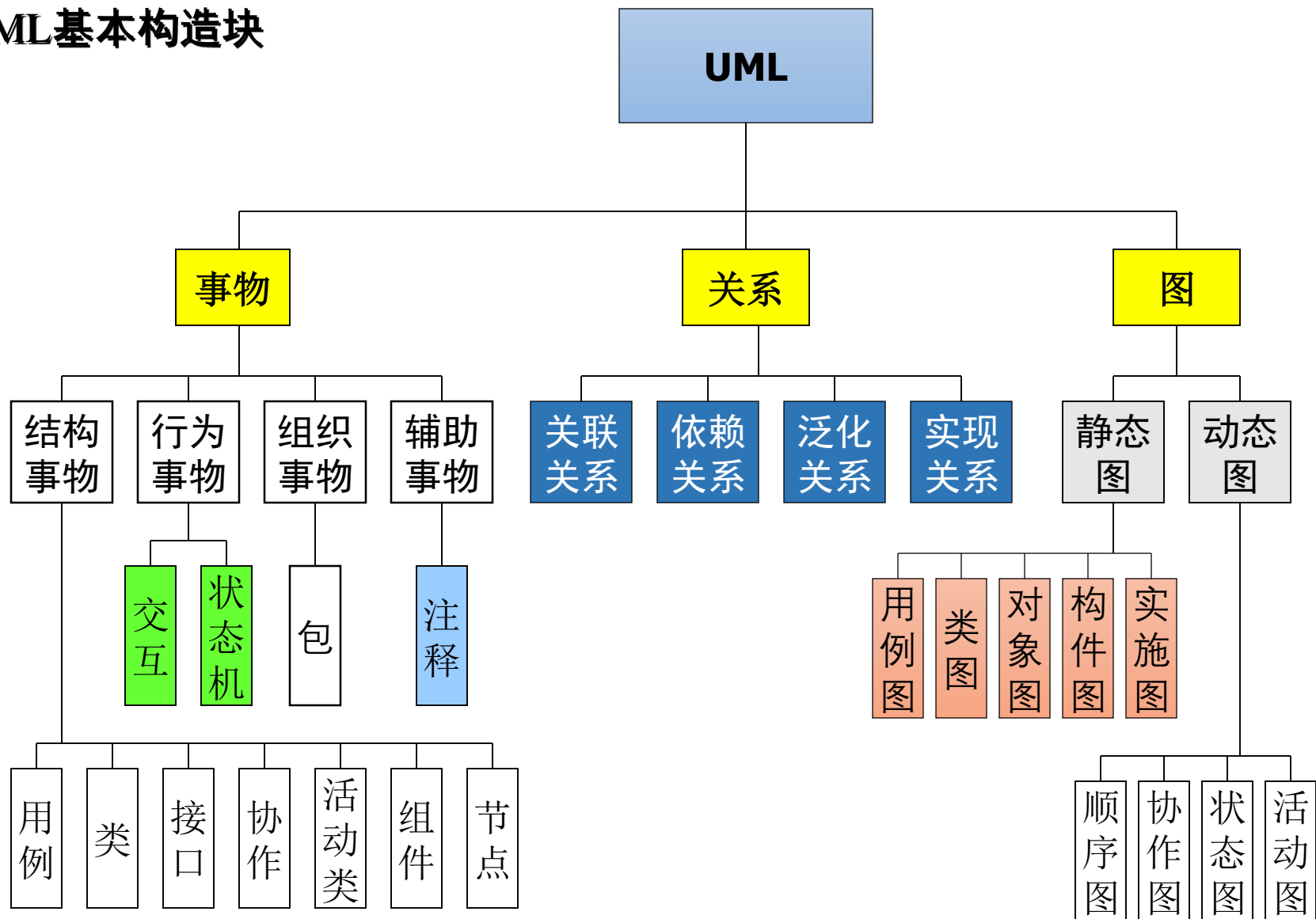
◇ UML的主要特点可归纳为:

- * 统一的标准: **UML**已被**OMG**接受为标准的建模语言
- * 面向对象
- * 可视化、表示能力强大
- * 独立于过程
- * 概念明确, 建模表示法简洁, 图形结构清晰, 容易掌握使用

◇ UML和程序设计语言的关系

- * 用**Java**, **C++** 等 **programming language**是用编码实现一个系统
- * 用**UML**是对一个系统建立模型
- * 一些软件工具可以根据 **UML**所建立的系统模型来产生**Java**, **C++** 或其它程序设计语言代码框架。

UML基本构造块



UML的重点图

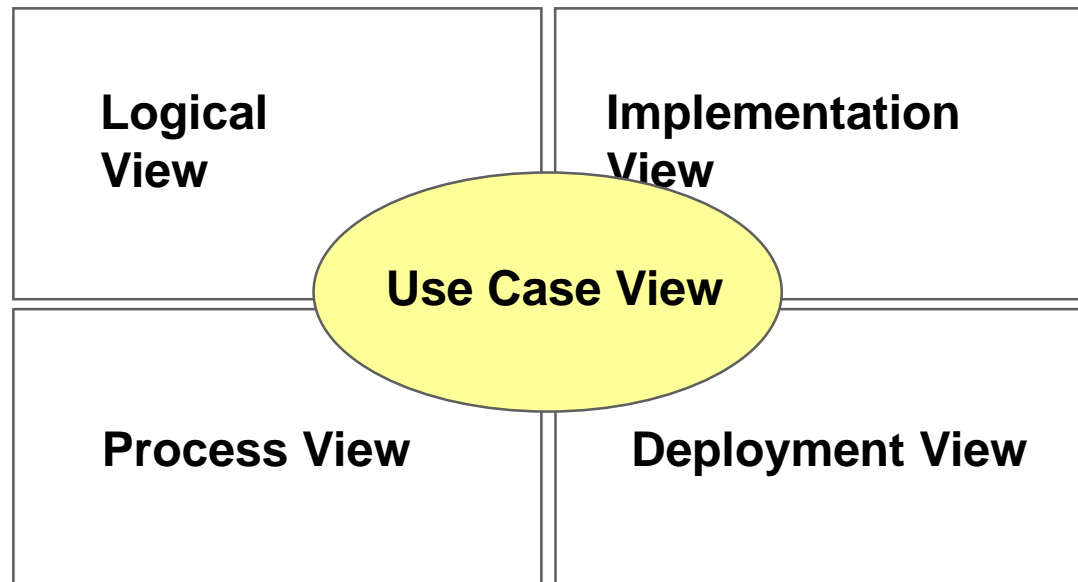
- 1、用例图（**use case diagram**）
- 2、类图（**Class diagram**）
- 3、活动图（**activity diagram**）
- 4、顺序图（**Sequence Diagram**）
- 5、交互纵览图(**Interaction Overview Diagram**)
- 6、通信图(**Communication Diagram**)
- 7、状态图(**State Machine Diagram**)
- 8、构件图(**Component Diagram**)
- 9、部署图（**Deployment Diagram**）

UML中的视图

◇ UML中的视图包括:

用例视图(Use Case View)、逻辑视图(Logical View)、实现视图(Implementation View)、进程视图(Process View)、部署视图(Deployment View)等。

这5个视图被称为“4+1”视图。



Logical View 描述系统设计特征，包括结构模型视图和行为模型视图，前者描述系统的静态结构(类图、对象图)，后者描述系统的动态行为(交互图、状态图、活动图)。

Use case View描述系统的外部特性、系统功能等。

Implementation View 表示系统的实现特征，常用构件图表示。

逻辑视图

实现视图

Use case
视图

进程视图

配置视图

Process View 表示系统内部的控制机制。常用类图描述过程结构，用交互图描述过程行为。

Deployment View 配置视图描述系统的物理配置特征。用配置图表示。

UML在系统开发各阶段的应用

- ◇ 在分析阶段，用户的需求用**UML**模型来描述。
- ◇ 在设计阶段，引入定义软件系统中技术细节的类（如处理用户接口、数据库、通信和并行性等问题的类）。
- ◇ 在实现阶段，用面向对象程序设计语言将来自设计阶段的类转换成实际的代码。
- ◇ **UML**模型还是测试阶段的依据
 - * 单元测试使用类图和类规格说明
 - * 集成测试使用构件图和协作图
 - * 系统测试使用用例图来验证系统的行为