

第01讲 绪论

徐峰磊

QQ:474017512

Copyright©2022, Software Research Team in USTS

All rights reserved

主要内容

1

课程定位 (Location)

Where are you ?

2

课程概述 (Introduction)

What are you
doing?

3

建模总结 (Summary)

4

UML简介 (Unified Modeling Language)

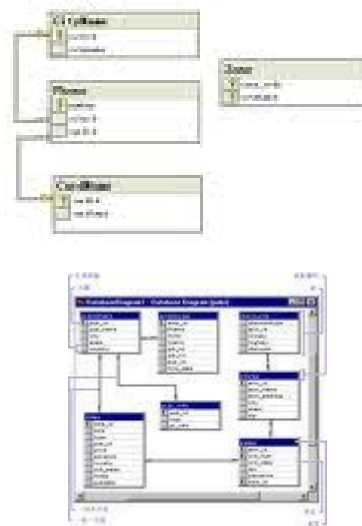
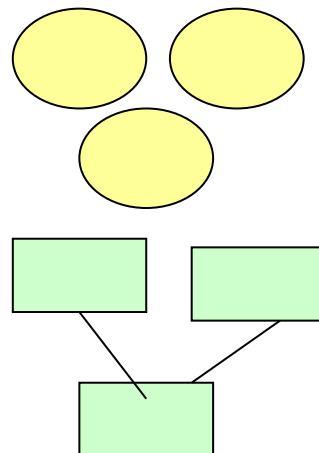
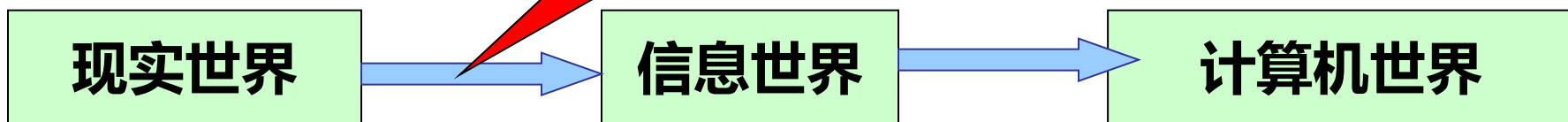


课程定位

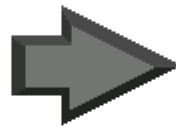
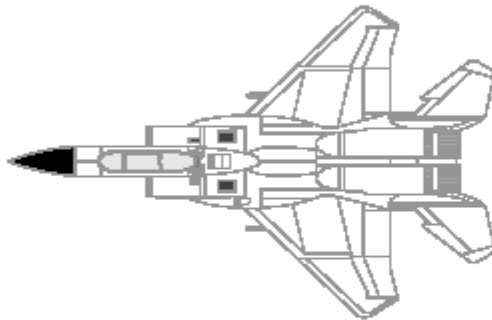
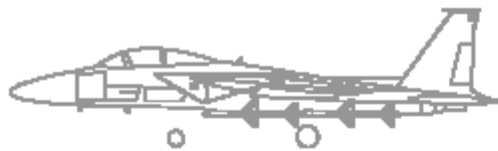
➤ 三“界”变换

软件建模
与分析

数据库设计
数据化



模型



模型是对现实世界的简化

- 为什么工程师要建造模型(models)? 为什么航天工程师要建造航天器的模型? 为什么桥梁工程师要建造桥的模型? 提供这些模型的目的是什么?
- 这些工程师建造模型来查明他们的设计是否可以正常工作。航天工程师建造好了航天器的模型, 然后把他们放入风洞中了解这些航天器是否可以飞行。桥梁工程师建造桥的模型来了解桥能否接立起来。建筑工程师建造建筑的模型了解客户是否喜欢这种建筑模样。通过建立模型来验证事物是否可工作。

- 为了更好的了解一个过程或事物，人们通常根据所研究对象的某些特征 (形状、结构或行为等) 建立相关的模型
- 模型是从一个特定视点对系统进行的抽象
 - 它可以是实物模型，例如建筑模型、教学模型、玩具等
 - 也可以是抽象数字或图示模型，例如数学公式或图形等

开发软件为什么需要模型

- 在开发软件的过程中，开发者在动手编写程序之前需要研究和分析软件的诸多复杂和纷乱的问题

用户需求的准确描述问题

功能与功能之间的关系问题

软件的质量和性能问题

软件的结构组成问题

建立几十个甚至几百个程序或组件之间的关联问题等等

- 在这个复杂的开发过程中，我们最关注的还是开发者之间的交流问题

- 软件开发中能否消除技术人员与非技术人员(用户)之间、使用不同技术的开发人员之间、不同功能使用者之间的等等交流障碍是软件开发成功的关键
- 直观的软件模型将有助于软件工程师与他们进行有效的交流

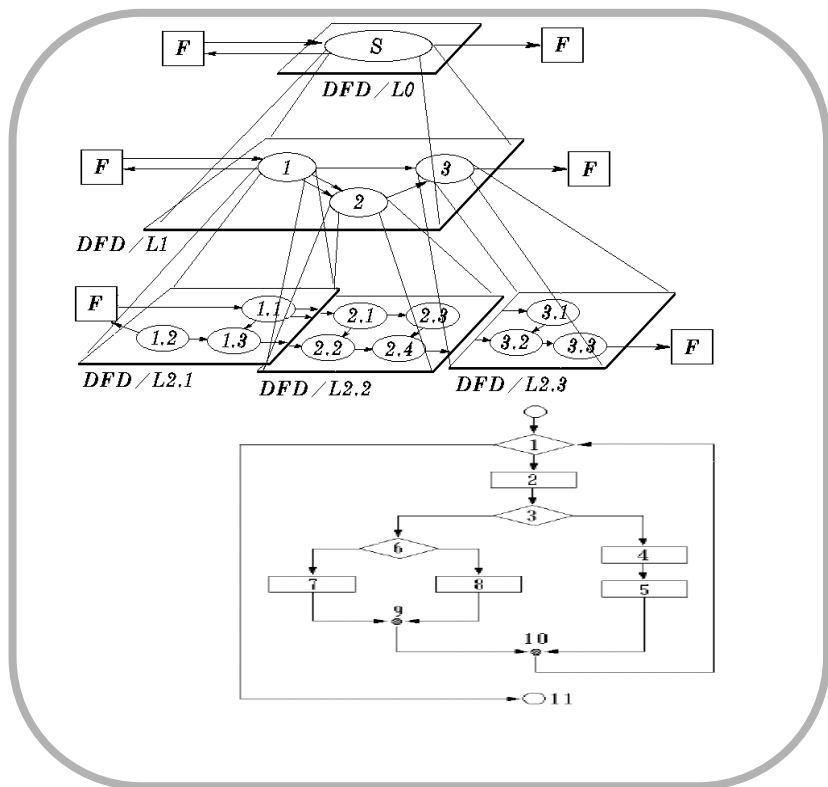
开发软件为什么需要模型

- 软件开发中需要建立
 - -需求模型
 - 分析模型
 - 设计模型
 - 实施模型
 - 测试模型
 - 部署模型
- 在系统开发生命周期中，需要从多角度来建立模型才能全面、准确地分析和设计软件系统

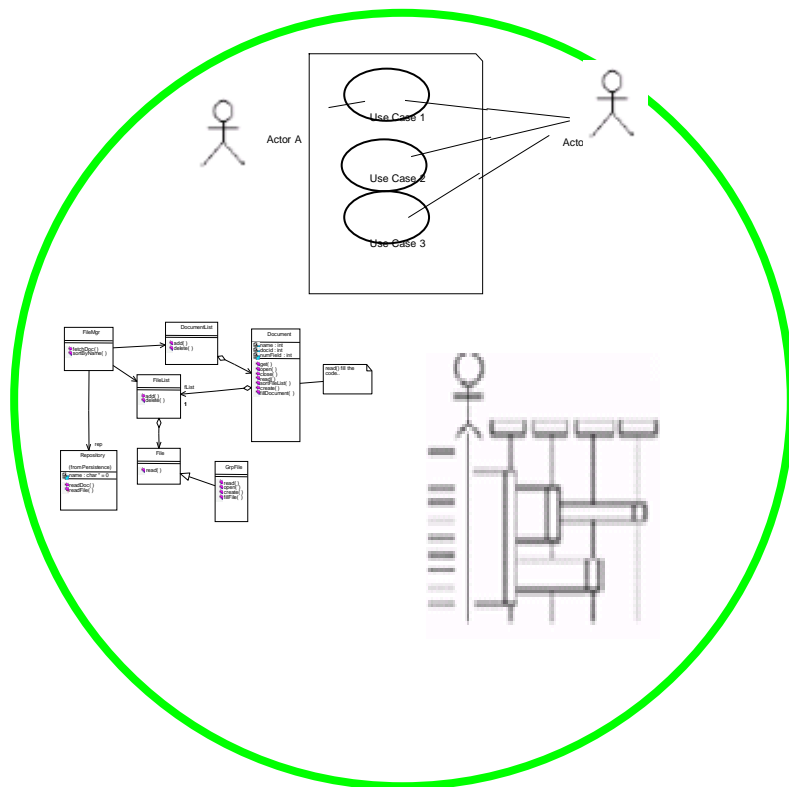
- 软件模型有多种表达方式或语言
- 开发者们经过多年的实践发现，以图形符号为基础的统一建模语言(Unified Modeling Language, UML) 描述软件模型既简单又清晰

背景介绍

► 建模方法



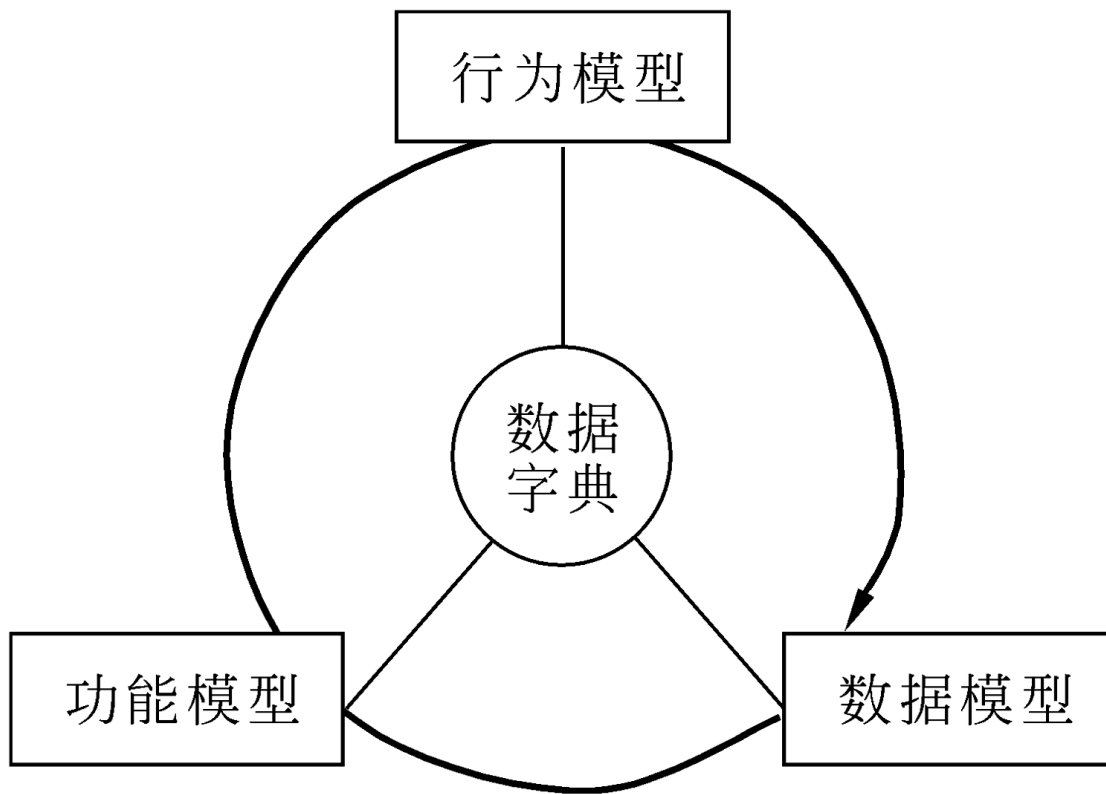
结构化方法
(Structural)



面向对象方法(Object-Oriented)

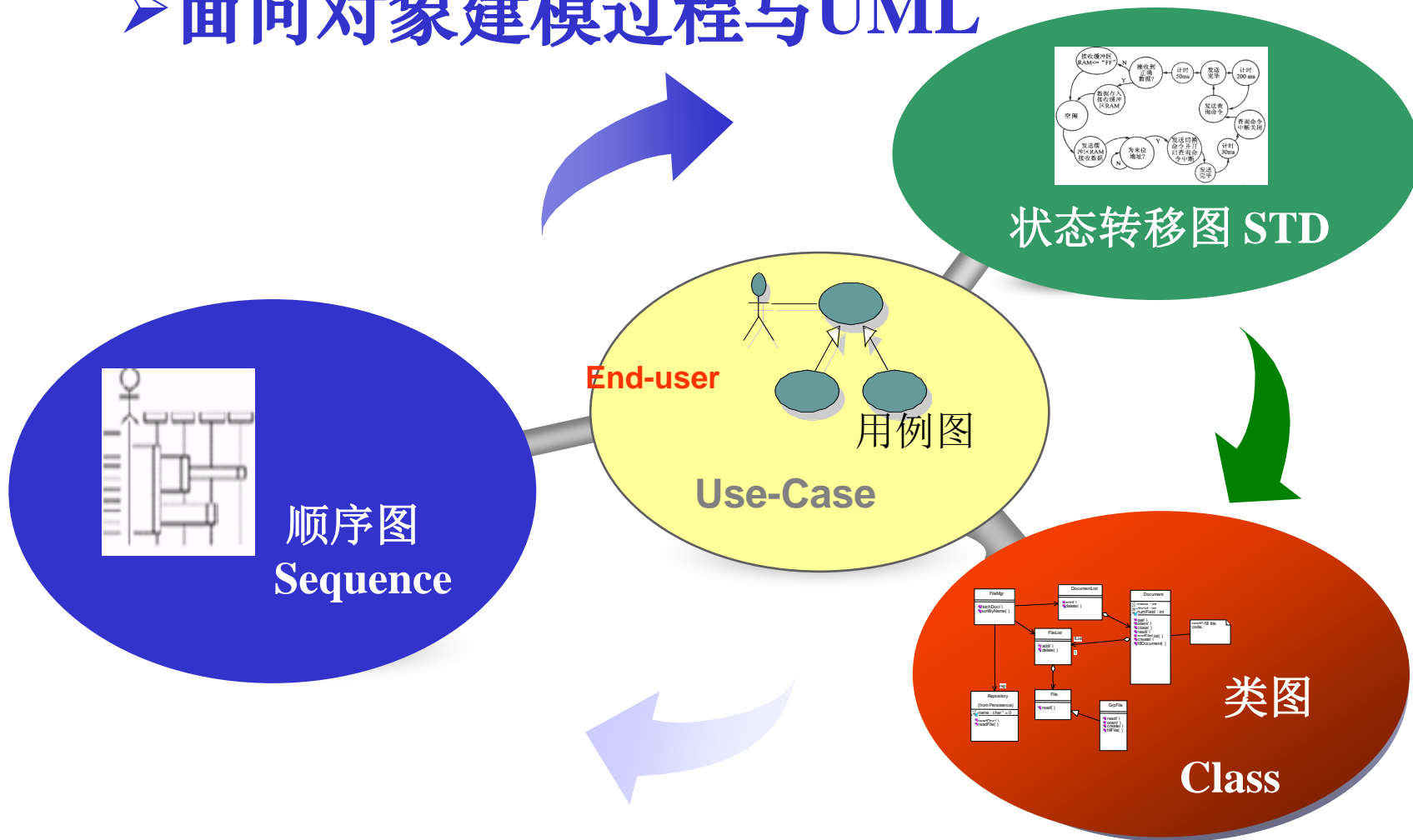
课程概述

➤ 建模过程

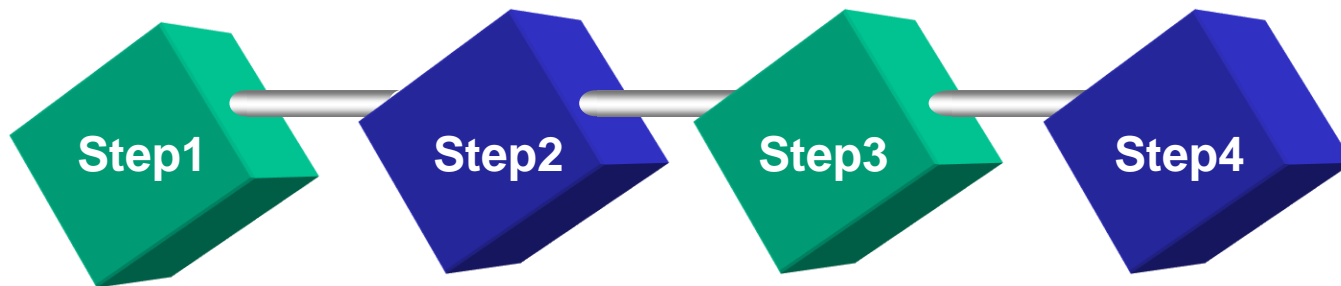


课程概述

➤ 面向对象建模过程与UML



UML建模流程



1.1 识别Actor

- ✓谁使用该系统?
- ✓谁改变系统的数据?
- ✓谁从系统获取信息?
- ✓谁需要系统的支持以完成日常工作任务?
- ✓谁负责维护、管理并保持系统正常运行
- ✓系统需要应付那些硬件设备
- ✓系统需要和那些外部系统交互
- ✓谁对系统运行产生的结果感兴趣
- ✓时间、气温等内部外部条件

1.2 寻找用例

2.1 创建类 (名词识别法)

- ◆ 边界类
- ◆ 实体类
- ◆ 控制类

2.2 创建类图

分析功能顺序

通常一个Actor与一个用例之间的交互对应一个边界类

控制类协调边界类和实体类。通常一个用例对应一个控制类。

分析行为状态

实体类通常是用例中的对象，对应客观世界中的事物。

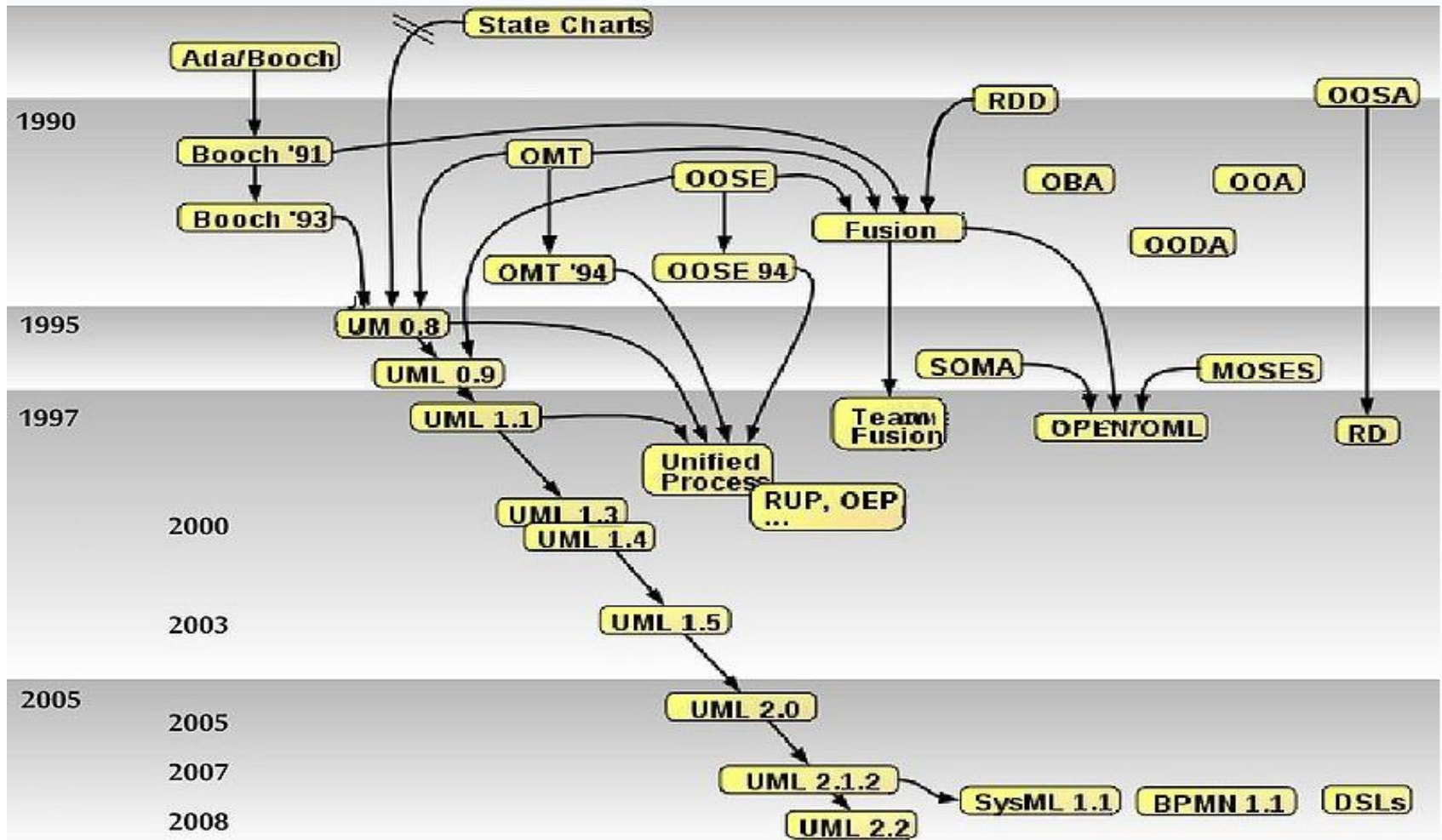
小结

- UML建模四步曲
- 由外向内，逐步精化



- 《UML系统建模》是一门与软件开发密切相关的建模方法。1968年产生软件工程，后来面向对象的分析与设计方法的发展在20世纪80年代末至20世纪90年代中出现了一个高潮，UML是这个高潮的产物。
- 它不仅统一了Booch、Rumbaugh和Jacobson的表示方法，而且对其有了进一步的发展，并最终统一为大众所接受的标准建模语言。

1.1 UML的发展史



1.2 日常生活中的应用

- UML的目标是以面向对象相关图的方式来描述任何类型的系统。最常用的是建立软件系统的模型，以及处理复杂数据的信息系统、具有实时要求的工业系统或工业过程，但它同样可以用于描述非软件领域的系统，如机械系统、企业机构或业务过程等

1.3 本课程学习中需要注意的问题

- (1) UML常用的9种图的画法及作用，适用于小型案例，也是后续中大型软件的基础。
- (2) 中型案例可按照统一过程的进行。
- (3) 大型案例可能涉及到人命关天的安全系统，大型复杂系统的需求是很难说清或抽取的，这种系统建模过程中的复杂网络需求获取方法和形式化方法技术需要不断更新与训练。

怎样成为优秀的软件模型设计者

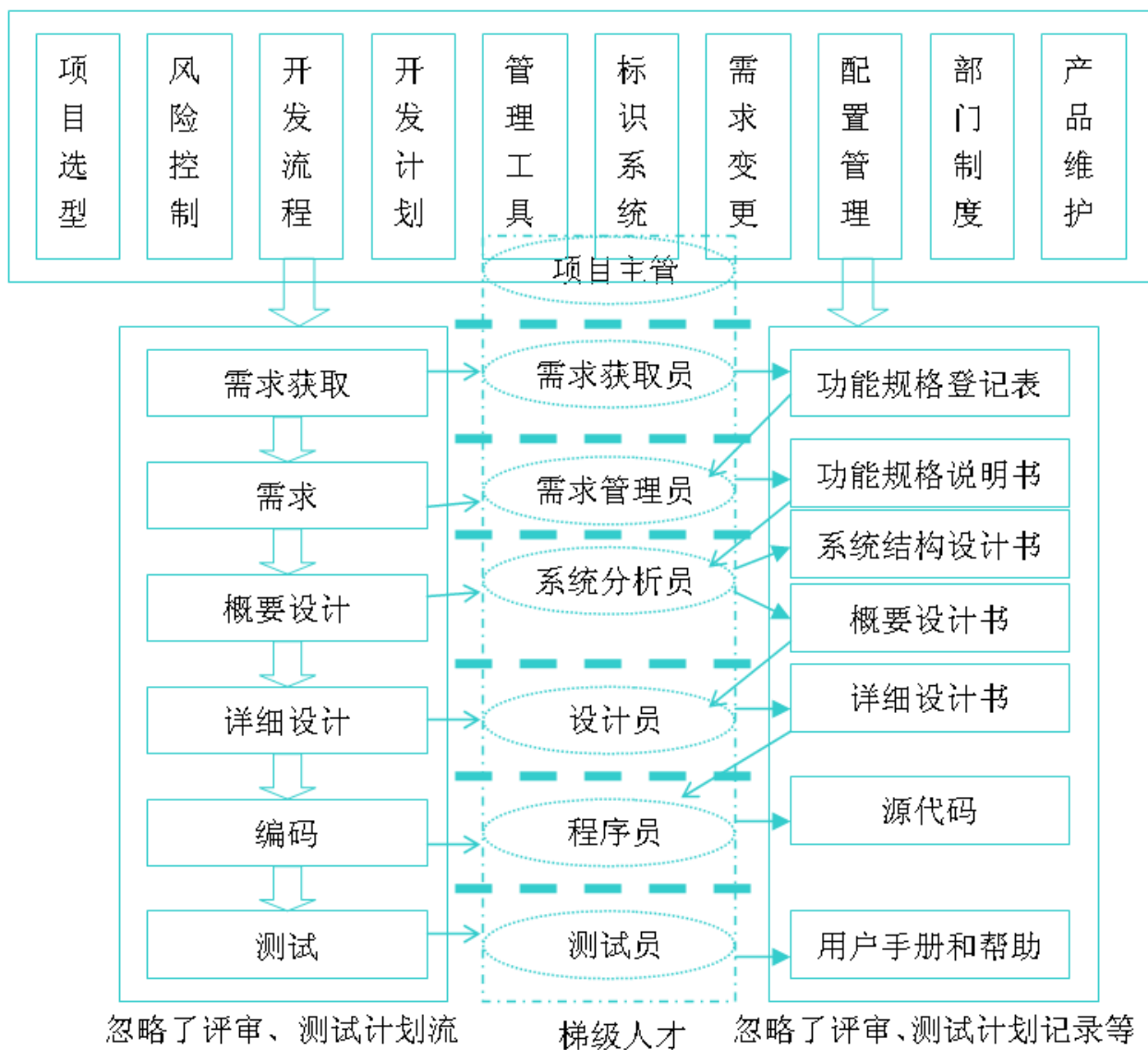
- 1. 人远比技术重要
- 2. 理解你要实现的东西
- 3. 谦虚是必须的品格，永远认识到自己的不足
- 4. 需求就是需求，必须照办
- 5. 比起需求的改变，更多的是你对需求的理解发生改变

- 6. 经常阅读别人的设计
- 7. 降低软件模块间的耦合度
- 8. 提高软件的内聚性
- 9. 考虑软件的移植性
- 10. 接受变化，拥抱变化

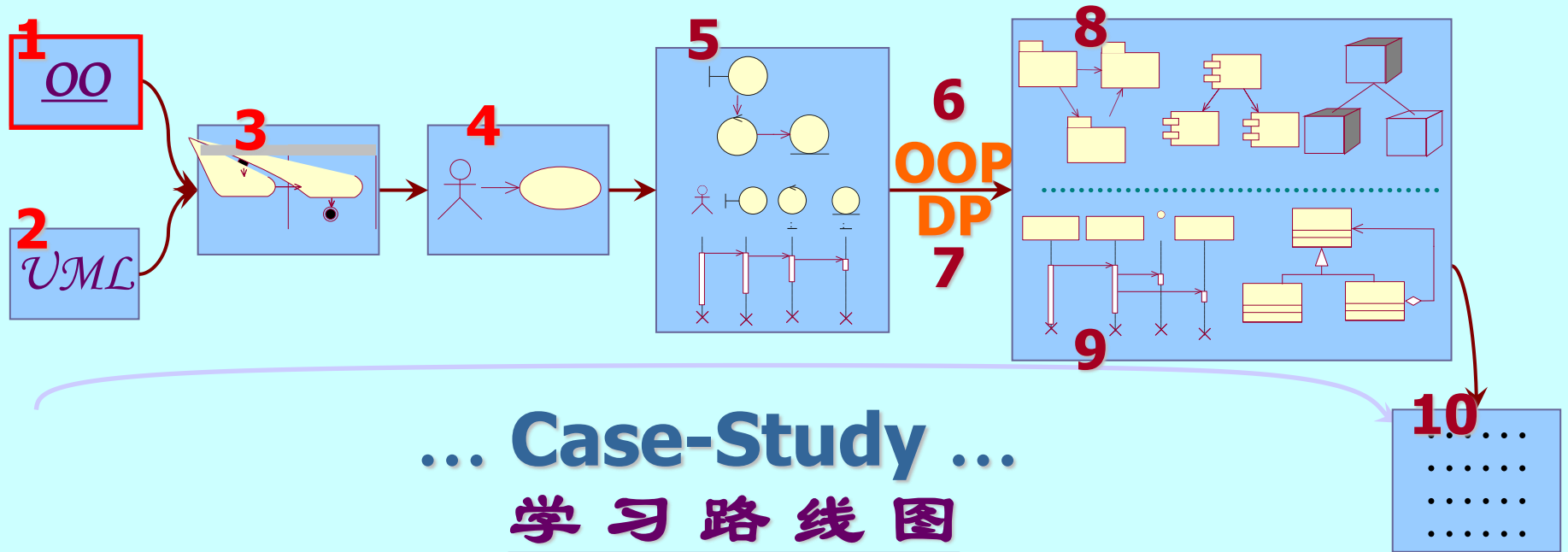
- 11. 不要低估对软件规模的需求
- 12. 性能仅仅是很多设计因素之一
- 13. 管理接口
- 14. 简单可靠的设计，不断的重构代码
- 15. 证明你的设计在实践中可行

- 16. 应用已知的模式
- 17. 研究每个模型的长处和弱点
- 18. 在现有任务中应用多个模型
- 19. 理解完整的过程
- 20. 常做测试，早做测试
- 21. 不断的将工作归档
- 22. 技术常变，基本原理不会

项目管理



学习路线图



开发过程解析

- 业务建模：用软件建模方法描述业务流程；其目标是认识业务本质，该业务本质是后续用例建模的基础
- 用例建模：采用UML用例建模技术描述软件需求，该需求模型将为后续用例分析提供输入
- 用例分析：采用UML用例分析技术分析软件需求，建立软件系统的分析模型
- 架构设计：在系统的全局范围内，以分析模型为基础，设计系统的架构
- 构件设计：根据架构设计的成果，将分析模型细化，设计系统构件的实现细节
- 代码实现：将系统构件映射到目标语言上

Any Question?