



# UML建模

陈静玉

西安电子科技大学 软件学院

# 课程内容



## 第一部分： 信息系统建模 基础知识

- 模型的基本概念
- 面向信息系统的建模
- 面向对象的建模方法

## 第二部分： 如何运用 UML建模

- 统一建模语言UML简介
- 需求建模
- 结构模型
- 行为模型
- UML建模实例

## 第三部分： 展望

- UML的发展与其他建模技术

# 评分方式

- 考核方式：大作业 70%，平时成绩 30%
- 大作业内容
  - 根据应用系统的需求，设计和建立各种模型和文档
  - 需求模型40%，分析模型40%，设计模型20%
- 大作业提交

[umlsubmit11@163.com](mailto:umlsubmit11@163.com)

  - 邮件要求：
    - 题目：UML\_学号\_姓名\_xxx次作业
    - 以附件形式发送

# 课件下载及作业提交



- 课件下载

用户名: `umlshare@126.com`

password: `uml123456`

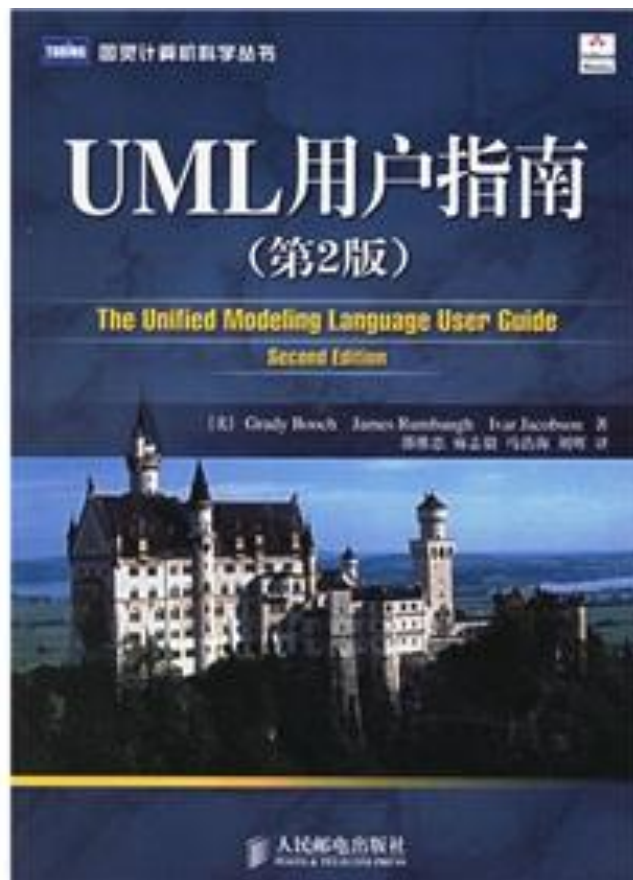
- 大作业提交

`umlsubmit11@163.com`

# 教材



- 《UML用户指南》（第2版）Grady Booch, James Rumbaugh, Ivar Jacobson 著. 邵维忠, 麻志毅, 马浩海, 刘辉 译. 人民邮电出版社



# 参考书目



- (美) Jim Arlow, Ila Neustadt 著. 方贵宾, 胡辉良 译. 《UML 2.0和统一过程》(原书第2版). 北京: 机械工业出版社, 2006
- Grady Booch, James Rumbaugh, Ivar Jacobson编 《UML参考手册》. 北京: 机械工业出版社, 2001
- Blaha, Rumbaugh 编 《UML面向对象建模与设计》. 北京: 人民邮电出版社, 2006
- (美) Craig Larman著. 《UML和模式应用》(第3版). Prentice Hall PTR . 2006
- 青润 著. 《软件工程之全程建模实现》. 电子工业出版社, 2004
- 徐宝文, 周毓明, 卢红敏 编 《UML与软件建模》. 北京: 清华大学出版社, 2006
- 张维明 主编 《信息系统建模》. 北京: 电子工业出版社, 2002
- J L. Whitten等编 《系统分析与设计方法》. 北京: 机械工业出版社, 2004
- Wendy Boggs, Michael Boggs编 《UML与Rational Rose 2002从入门到精通》. 北京: 电子工业出版社, 2002



第一部分：信息系统建模的  
基础知识

# 第一章 模型的基本概念

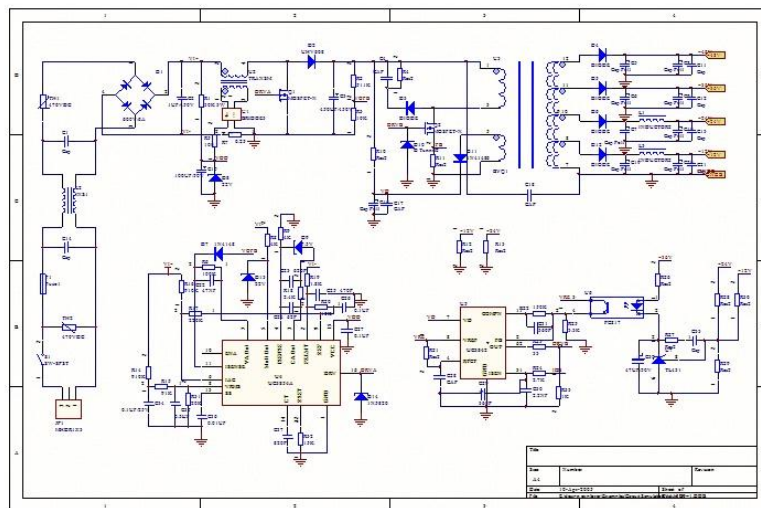
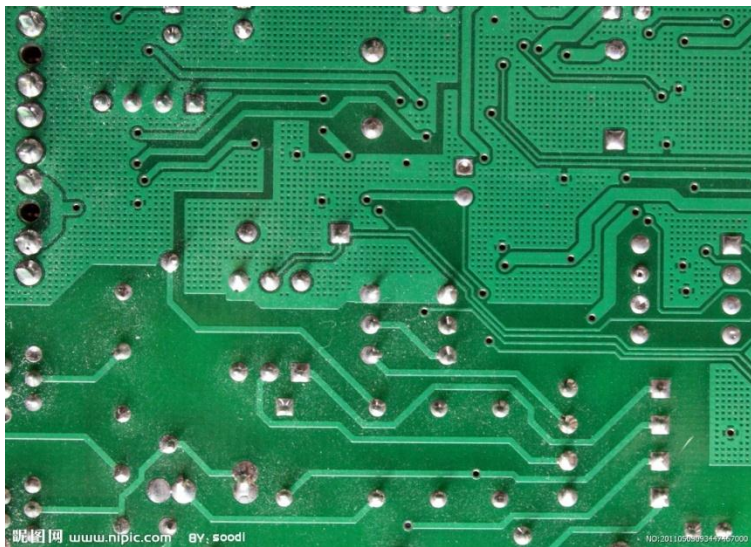
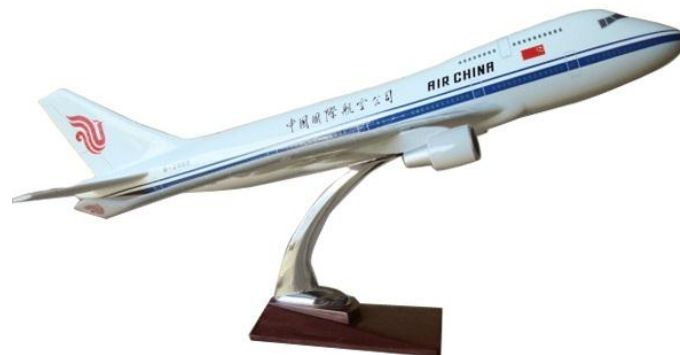
# 提纲



- 模型的含义
- 模型的用途
- 建模的重要性
- 模型的层次
- 模型的内容



# 模型的含义



# 模型的含义



- 什么是模型？
  - 模型是对现实世界中事物的简化和抽象
- 模型的范畴非常广泛
  - 自然科学与工程技术的概念、公式、定理等
  - 社会科学的学说、原理、政策等
- 模型已得到广泛应用
  - 工业生产领域
  - 科学研究中

# 模型的含义

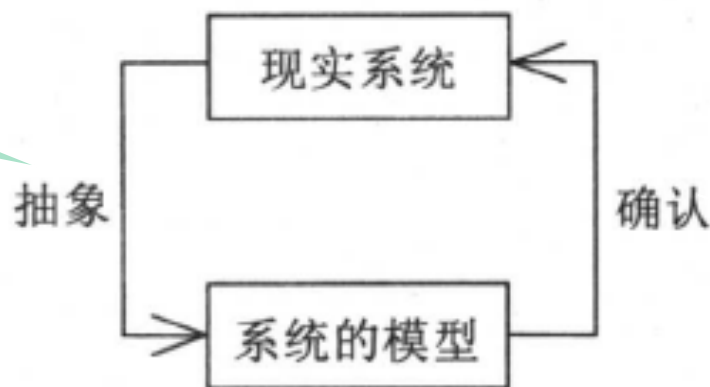


- 几种对模型的定义
  - 模型是实际系统或过程的一部分属性的**抽象**或模拟，而不是全部属性的复制
  - 模型是对真实对象与真实关系中有用的和令人感兴趣的特性的**抽象化**
  - 模型是对一切客观事物及其运动形态的特征和变化规律的一种定量**抽象**
- 抽象
  - 强调本质，扬弃次要因素

# 模型的用途

- 模型是理解、分析、开发或改造现实系统的一种手段，有助于设计者
  - 把握全局，不陷入细节
  - 洞察规律，分离重要属性

建立模型



分析模型

# 模型的用途



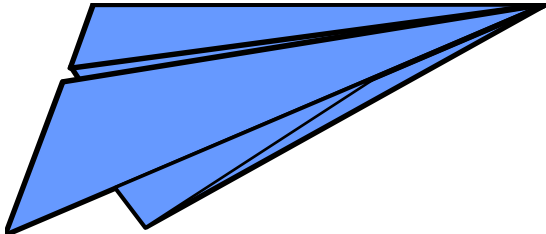
- 模型的几种用途
  - 在构建物理实体之前先测试
  - 促进相关人员的理解和交流
  - 降低复杂度
  - 缩短开发周期
  - .....

# 建模的重要性

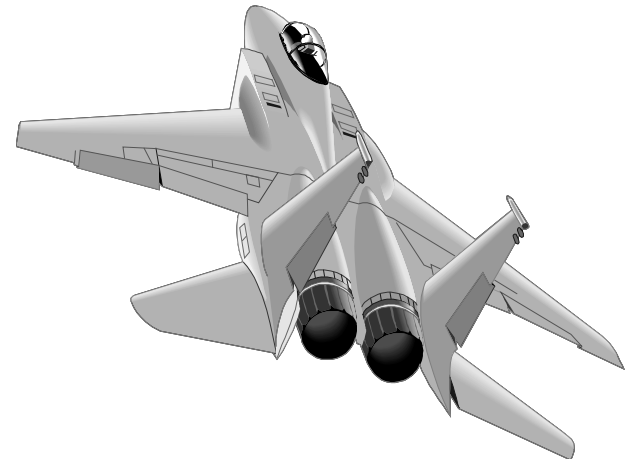


Less Important

More Important



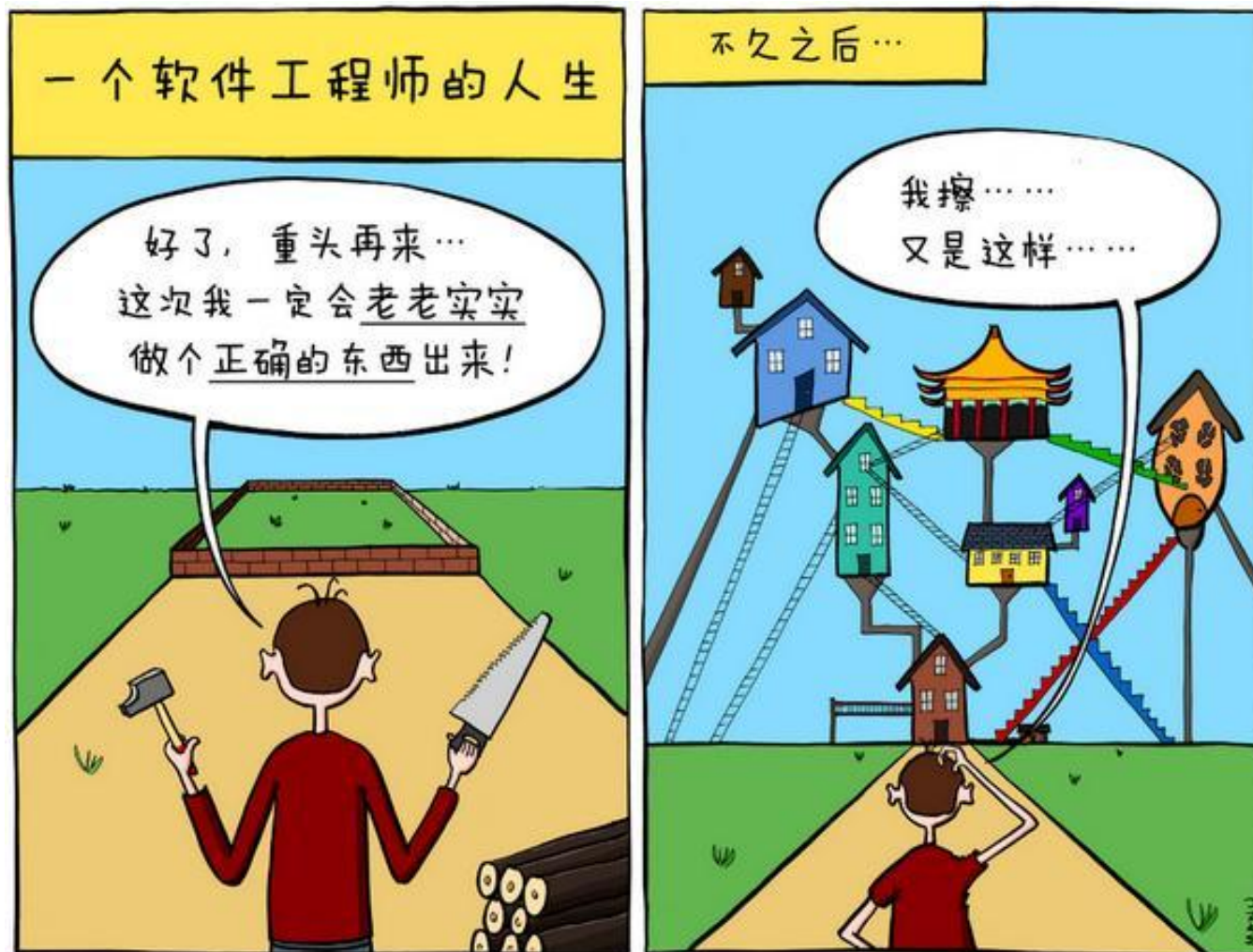
Paper Airplane



Fighter Jet



# 建模的重要性



汉化：@一只浣

# 建模的重要性

- “建模是为了更好地理解正在开发的系统”
  - 帮助我们按照实际情况或按照所需要的样式对系统进行可视化
  - 允许我们详细说明系统的结构和行为
  - 给出一个指导我们构造系统的模板
  - 对我们做出的决策进行文档化

——《UML用户指南》



# 模型的层次

- 概念模型
  - 表述对象系统的轮廓和主要特征
  - 可能不完善，无法实现
- 逻辑模型
  - 考虑结构合理性及实现可行性
  - 没有具体细节
- 物理模型
  - 确定的可实现模型
  - 具有具体实现细节

抽象层次依次降低



# 模型的内容



- 模型的组成
  - 系统：描述的对象
  - 目标：要达成的目标或功能
  - 组件：构成系统的子部分
  - 约束条件：系统所处环境及约束
  - 变量
    - 内部变量
    - 外部变量
    - 状态变量
  - 相关：变量间的关系

# 模型的内容



- 模型的表示

一个6元组  $M = \{O, G, T, V, R, S\}$

- $O$  — 对象集

- $G$  — 目标集

- $T$  — 环境及约束条件集

- $V$  — 变量集

- $R$  — 变量间的关系集

- $S$  — 状态集（由初态到终态）

# 小结



- 重点
  - 理解建模的本质是抽象
  - 了解建模的重要性



第一部分：信息系统建模的  
基础知识

## 第二章 面向信息系统的建模

# 提纲



- 信息系统发展现状
- 信息系统建模
- 建模的四个原则
- 建模过程
- 面向对象的建模方法

# 对软件开发的基本认识

- 一个软件系统的开发，从两个方面刻画：
- 软件开发资源方面：
  - 工作人员，硬件资源，软件资源
  - 各种资源协调管理
- 软件开发过程方面：
  - 软件需求，总体设计，详细设计，代码实现，测试，产品提交，后期维护，升级
  - 各阶段所得到的产品（文档）在逻辑上相互一致，开发具有连续性

# 对软件开发资源的基本认识

- 硬件和软件的特点比较：
- 表现形式不同：硬件是实体，软件是思想产物。
- 生产方式不同：硬件流水线生产，软件要人与人沟通。
- 产品要求不同：硬件可有容许的误差，软件不允许有误差。
- 维护方式不同：硬件有备用件，有故障的软件，其备份产品可能依然有故障。



# 软件的固有复杂性

- 软件复杂性的产生原因：
  - 和计算机的体系结构有关。
    - 冯. 诺依曼机，硬件简单，软件复杂。
  - 相互联系的概念结构，很难用一组数学公式或物理定律来描述。
  - 各元素之间的相互作用关系具有不确定性。
  - 软件似乎可以随意扩充和修改。
  - 软件系统的生命周期大于硬件系统的生命周期。

# 控制软件复杂性的基本方法

- 分解——大系统分解成若干个小系统。
- 抽象——抽取系统的基本特性，忽略非基本特性。
  - 过程抽象——某个操作由一系列更低级的操作来完成。
  - 数据抽象——定义数据类型及其上的操作。
- 模块化——高内聚，低耦合。
- 信息隐蔽——封装。把模块内的实现细节与外界隔离，用户只需要知道模块的功能。

# 面向对象技术

- 面向对象（object-oriented, OO）
  - 体现了思想“分解，抽象，模块化，信息隐蔽”
- 传统的结构化方法：
  - 着眼于一个系统需要什么样的方式和处理过程。对问题进行功能分解和再分解。功能映射。
- 面向对象方法：
  - 对问题域进行自然分割，以更接近人类思维的方式建立问题域模型。对象映射。
- 面向对象=对象+类+继承+通信

# 面向对象技术

- 结构化分析和设计之间存在鸿沟
- 面向对象分析和设计之间不存在鸿沟

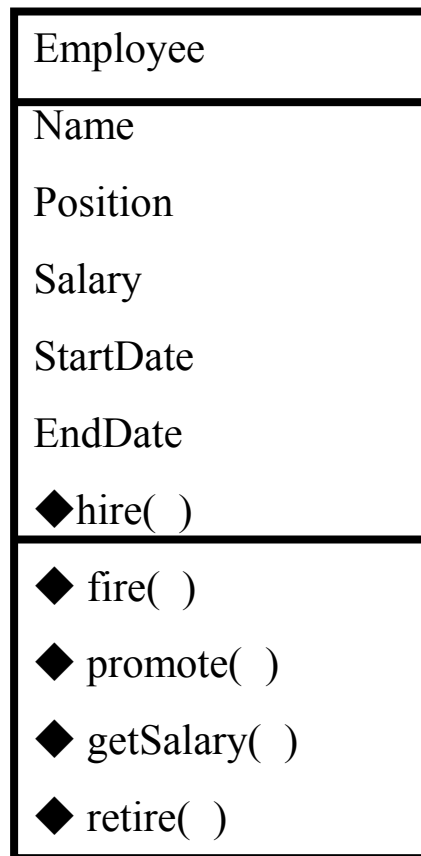


# 面向对象技术

- 面向对象技术的基本观点：
  - 客观世界由对象组成，任何客观实体都是对象，复杂对象可以由简单对象组成。
  - 具有相同数据和操作的对象可归纳成类。对象是类的一个实例。
  - 类可以派生出子类。子类除了继承父类的全部特性外，还可以有自己的特性。
  - 对象之间的联系通过消息传递来维系。

# 面向对象领域中的基本概念

对象、实例、类、属性、方法、封装、继承、多态、消息。



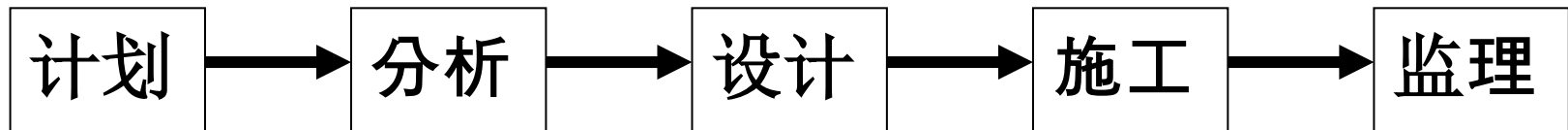
类Employee

# 面向对象领域中的基本概念

- 对象之间的关系：
  - 对象的分类关系
  - 对象的组成关系
  - 对象属性之间的静态联系
  - 对象行为之间的动态联系
- 面向对象方法分别用：
  - 一般/特殊结构，
  - 整体/部分结构，
  - 实例连接，
  - 消息连接，
- 来描述对象之间的上述4种关系。

# 对软件开发过程的基本认识(举例)

- 建筑工程流程



在北京王府井大街盖30层高的综合大楼，集购物、商务、休闲于一体，投资1亿，1年工期

建筑外观要求，地基要求，抗震要求，土地面积要求，使用面积要求，资金要求

建筑设计，结构设计，给排水设计，供暖设计，电气设计，地基设计，预算

按照图纸和文档，分别施工：建筑，结构，给排水，供暖，电气，地基

验收工程

设计者，校对者，审核者三级把关

施工者，设计单位派出的监理者

用户，设计单位，施工单位，政府



# 对软件开发的基本认识

- 软件流程



功能需求分析，资金需求分析，人力分析，工期分析

概要设计（系统功能结构图），详细设计（各模块功能实现的流程图）

使用某种计算机语言编程

程序代码是程序员个人杰作，思想上的产物，随意性强，太精练

模块测试，功能测试

多数情况下程序员就是测试员，语句执行路径的复杂性

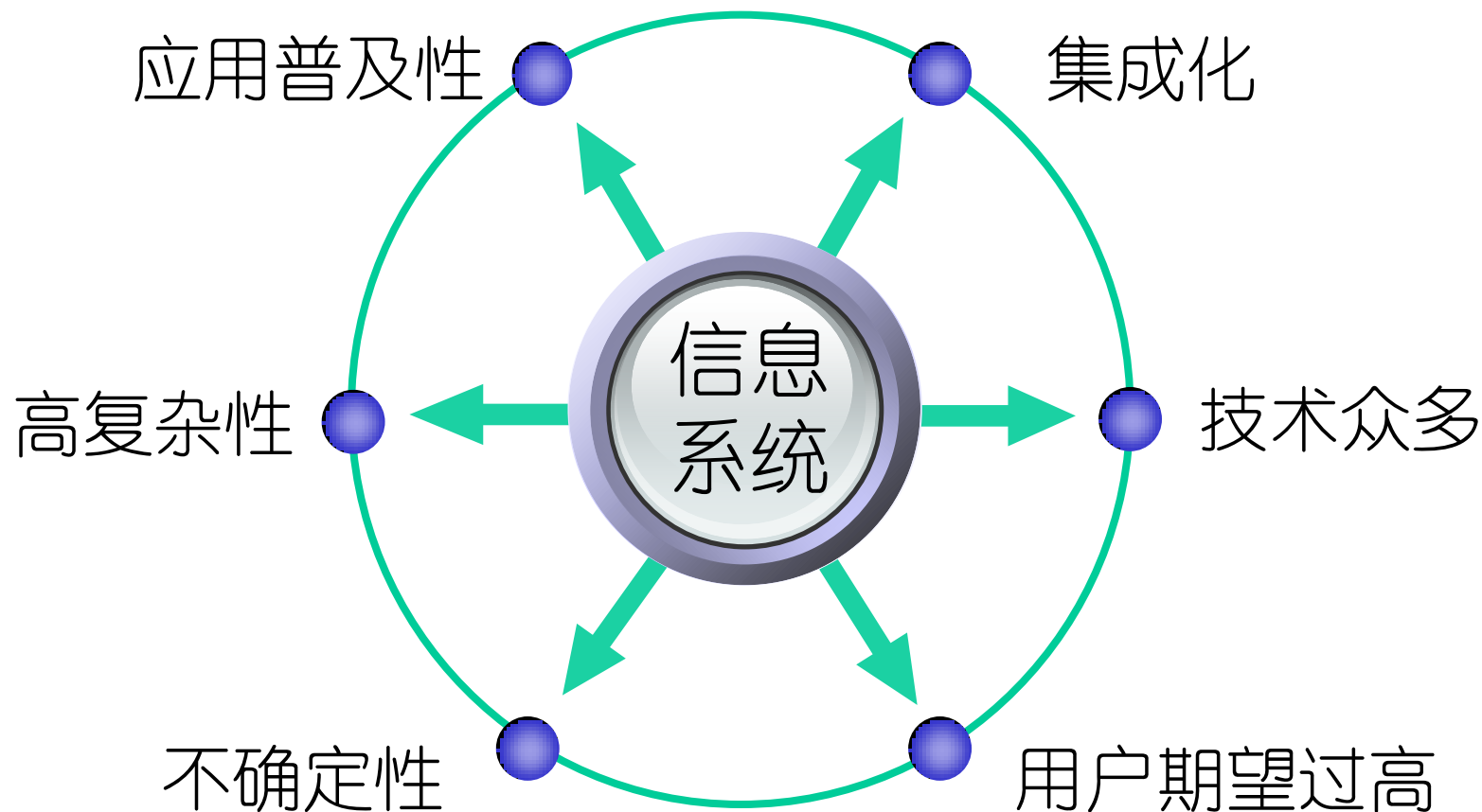
软件产品验收

用户要求以前描述、理解的不对，用户要求不断变化

# 软件危机及软件开发过程

- 提出各式各样的软件开发方法学：
  - **瀑布式软件开发方法**
  - **快速原型方法**
  - **螺旋式软件开发方法**
  - **变换式软件开发方法**
  - **增量式软件开发方法**
  - **净室软件开发方法**
  - **喷泉式软件开发方法**

# 信息系统发展现状



# 信息系统建模



- “分而治之”的思想
  - 理清复杂的信息关联
  - 寻找杂乱数据的规律
  - 将系统需求映射到软件结构
- 体现在开发过程中
  - 需求模型：全面描述用户需求
  - 设计模型：全局把握系统及内部联系
  - 数据模型：存储和处理海量数据
  - .....

# 信息系统建模



- 开放性
  - 采用已有标准；实现互操作
- 一体化
  - 各个阶段模型的无缝结合
- 易用性
  - 可视化；屏蔽细节
- 可重用性
- 多领域的协调

# 信息系统建模



- 信息系统模型的两种表现形式
  - 形式化描述
  - 图形化描述
- 形式化模型
  - 有严格的数学基础
  - 不易掌握；难以推广
- 图形化模型
  - 直观；易于描述层次与组成；有工具支持
  - 精确性与严谨性欠缺

# 建模的四个原则



- 建模的四个原则
  - 选择正确的模型
  - 分层
  - 联系实际
  - 分治

—— 《UML用户指南》

# 建模的四个原则

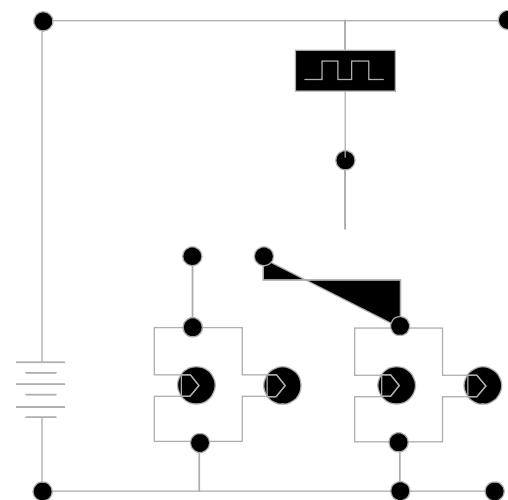
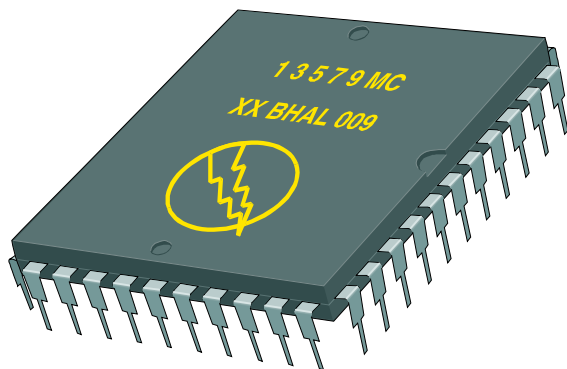
- 选择正确的模型
  - “The model you create influences how the problem is attacked.”
  - 所选模型将影响你对世界的看法
  - 不同的模型生成不同的系统

观点	生成模型
数据库开发者	实体-关系模型
结构化开发者	算法为中心的模型
面向对象开发者	以类及类间的协同工作为中心的模型



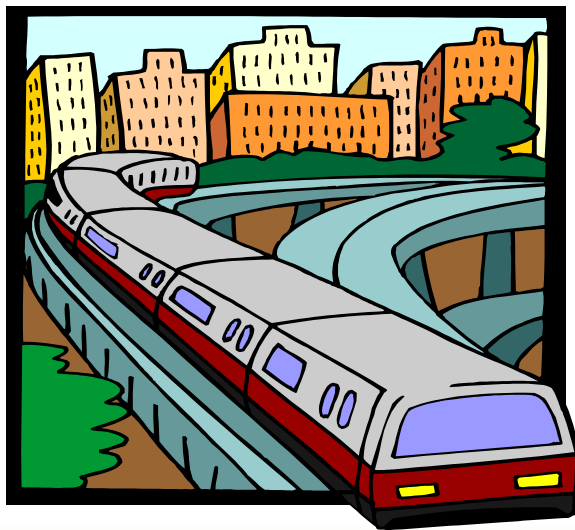
# 建模的四个原则

- 分层：有不同抽象层次的模型
  - “Every model may be expressed at different levels of precision.”
  - 好的模型选择不同抽象层次的依据：
    - 观察的角色
    - 观察的原因



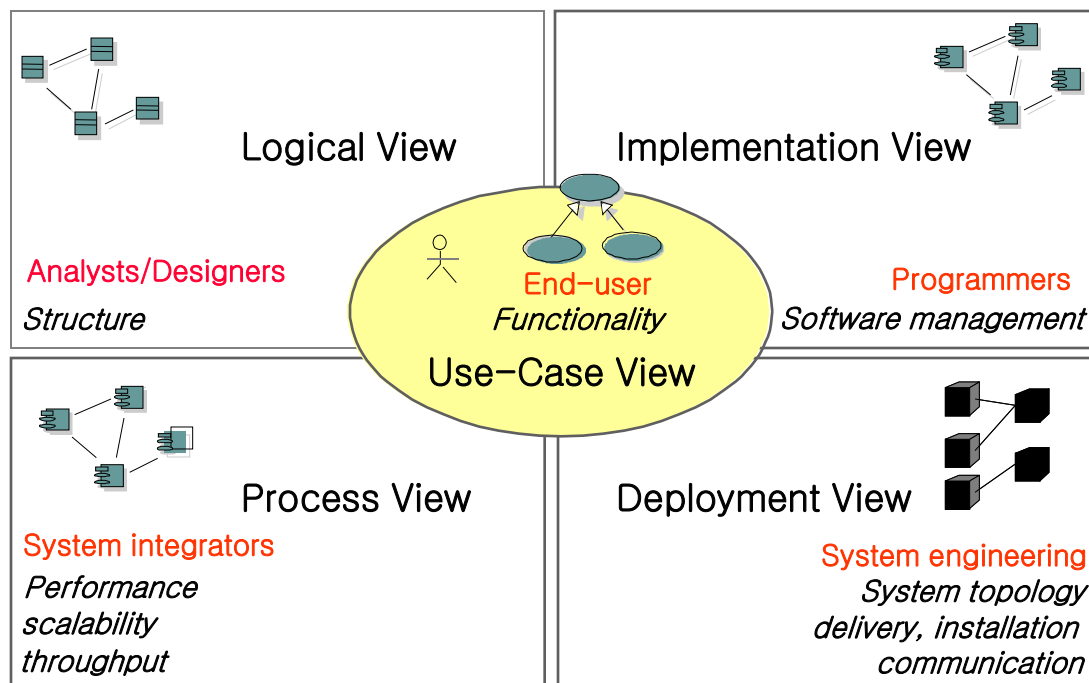
# 建模的四个原则

- 模型要与现实世界相联系
  - “The best models are connected to reality.”
  - 模型是现实世界的简化
    - 与现实世界存在映射
    - 联系薄弱时能够精确地知道这些模型是如何与现实世界拖节的



# 建模的四个原则

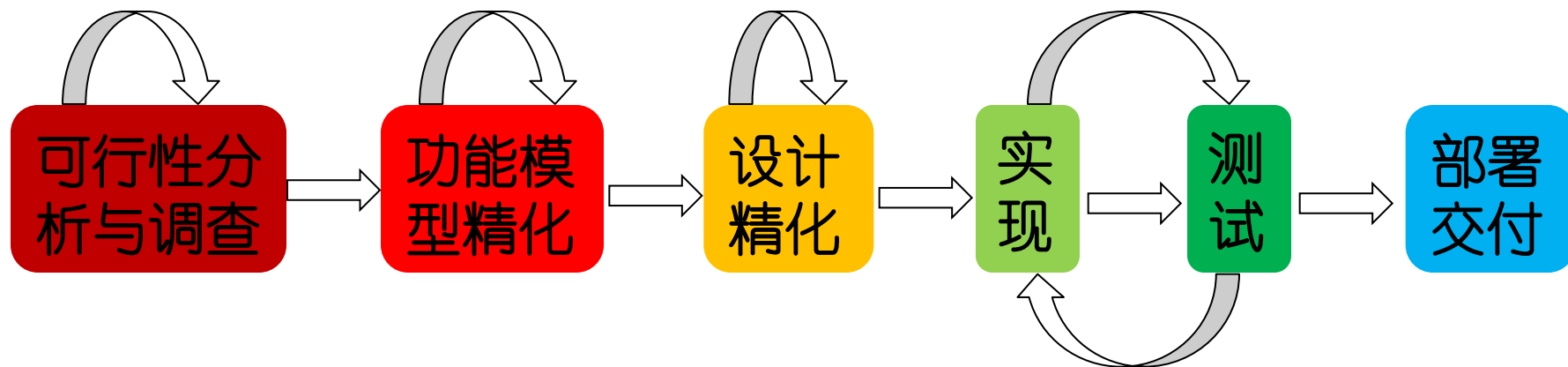
- 分治：若干模型反映不同侧面或多个组件
  - “No single model is sufficient.”
  - 能独立构建与研究并相关关联的多个模型



# 建模过程



- 两个思路
  - “自顶向下”
  - “自底向上”
- 信息系统建模过程



# 建模过程



- 可行性分析和调查



- 论证系统可行性

- 是否必要？多大代价？多长时间？……

- 功能模型精化

- 功能模型：从用户观点的“黑箱”视图

- 反复确定和细化系统功能的过程

- 需求建模

- 辨认业务对象（无设计/执行细节）



# 建模过程



- 功能模型精化（续）



- 及早界定用户需求

- 阶段性成果：需求说明书；业务对象模型；人机界面原型；……

- 设计精化

- 业务对象向设计对象的逐步精化

- 划分子系统

- 确定分布措施、数据存储措施和开发环境

- 阶段性成果：系统结构文档；对象结构文档；数据结构文档；……



# 建模过程



- 实现

- 设计模型向源代码的转换
- 数据模型生成物理数据库模式
- 阶段性成果：源代码；数据库模式；……



- 测试

- 校验系统是否满足需求和产生期望结果
- 系统测试与交付测试

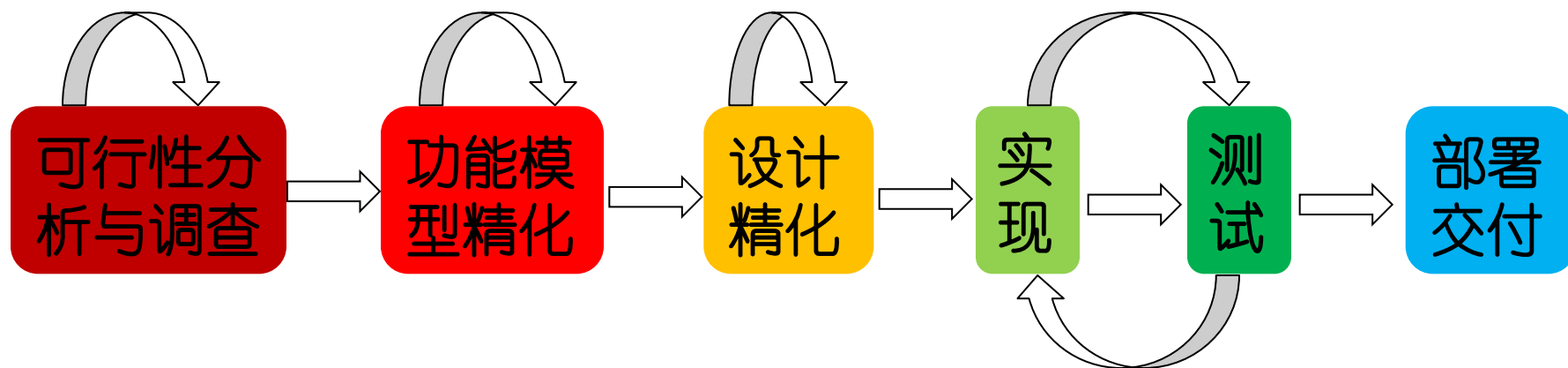


- 部署与交付

- 阶段性成果：最终用户文档；产品说明



# 建模方法



- “功能模型精化”阶段

— 需求分析方法  $\Rightarrow$  需求模型

- “设计精化”阶段

— 分析和设计方法  $\Rightarrow$  逻辑模型



# 信息系统建模

- 两种最常用的建模方法：
- 基于**功能的结构化**的和**面向对象**的。
  - 基于**功能的结构化**的方法中，主要的模块是程序或函数。开发人员将注意力集中在控制流和将庞大的算法拆分成各个小块。随着需求的变化和系统的增长，运用基于功能（算法）的建模方法建立起来的系统很难维护。
  - 在**面向对象**的建模方法中，主要的模块是对象或者类。在操作界面上，可以有按钮、菜单、对话框等对象。在数据库中，可以有实体类对象（对应一个信息表）。在中间层，可以将事务或交易的规则，封装在一个对象中。

## 面向过程

- 结构化方法
  - 通过描述与细化内部活动分析系统功能
    - 以过程或函数为主要构造块
    - 精力集中于控制流程和算法的分解
  - 缺点：难以维护
    - 需求发生变化时
    - 系统增长时

- 结构化分析方法
  - 数据流分析（DFD）法
    - 从数据流动的角度
    - 自顶向下
  - IDEF0法
    - 可同时表达系统活动与数据流及其关系
    - 适于系统全面功能建模
  - 缺点：需求分析之初就要深入内部细节

# 建模方法



- 建模方法与模型表示

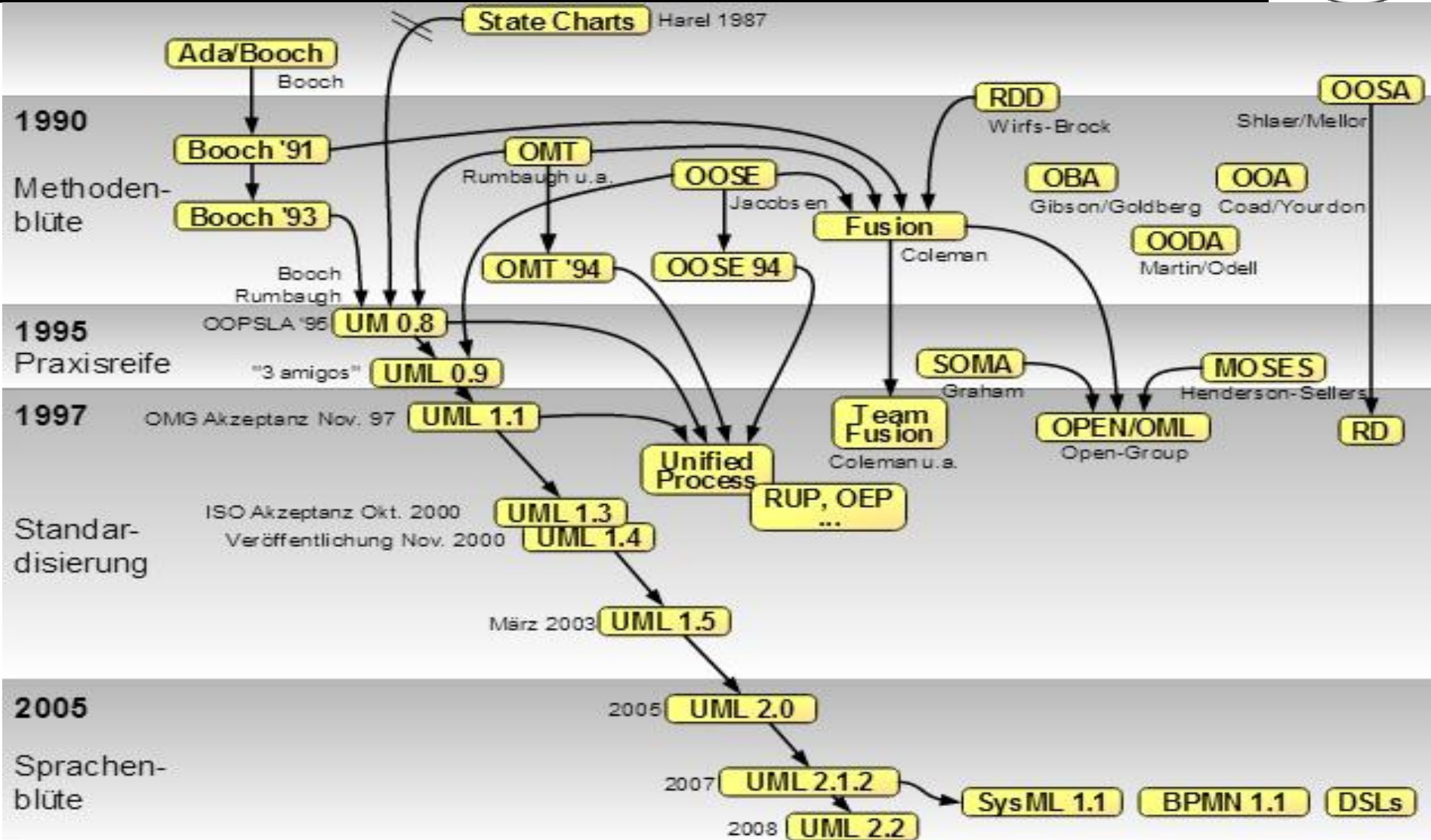
建模方法		模型表示
结构化方法	DFD	数据流图，数据字典，层次数据流图
	IDEF0	活动数据图

## 面向对象

- 用例驱动的分析方法
  - 用例：捕获系统中的用户功能需求
  - 分析系统与用户的交互状况
    - 系统应具有哪些功能或特性？
    - 给系统输入什么样的数据？
    - 系统有什么反应或输出？
  - 特点
    - 不涉及系统任何实现细节
    - 贯穿整个建模过程

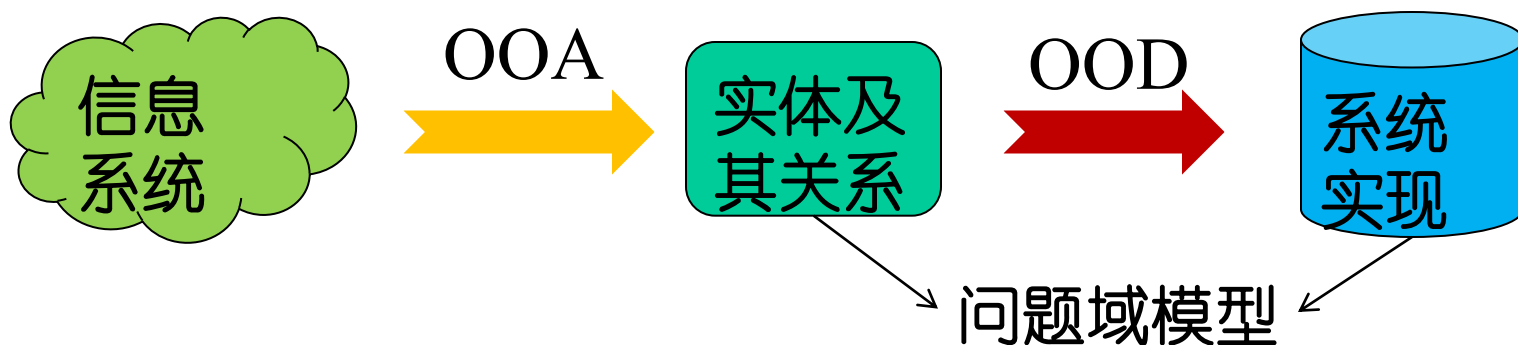
- 面向对象（OO）的建模方法
  - “OO”思想：封装；继承；多态；……
  - 描述系统的三种模型
    - 类模型：描述系统内部对象及其关系
    - 状态模型：描述对象生命历史
    - 交互模型：描述对象间交互行为
  - OO建模历史
    - Booch方法；OOSE方法；OMT方法……
    - 统一建模语言UML(Unified Modeling Language)

# 建模方法



# 建模方法

- OOA (Object Oriented Analysis)
  - 面向对象的分析
  - 把系统分解为实体及其关系
- OOD (Object Oriented Design)
  - 面向对象的设计
  - 解决实体与关系如何实现的问题

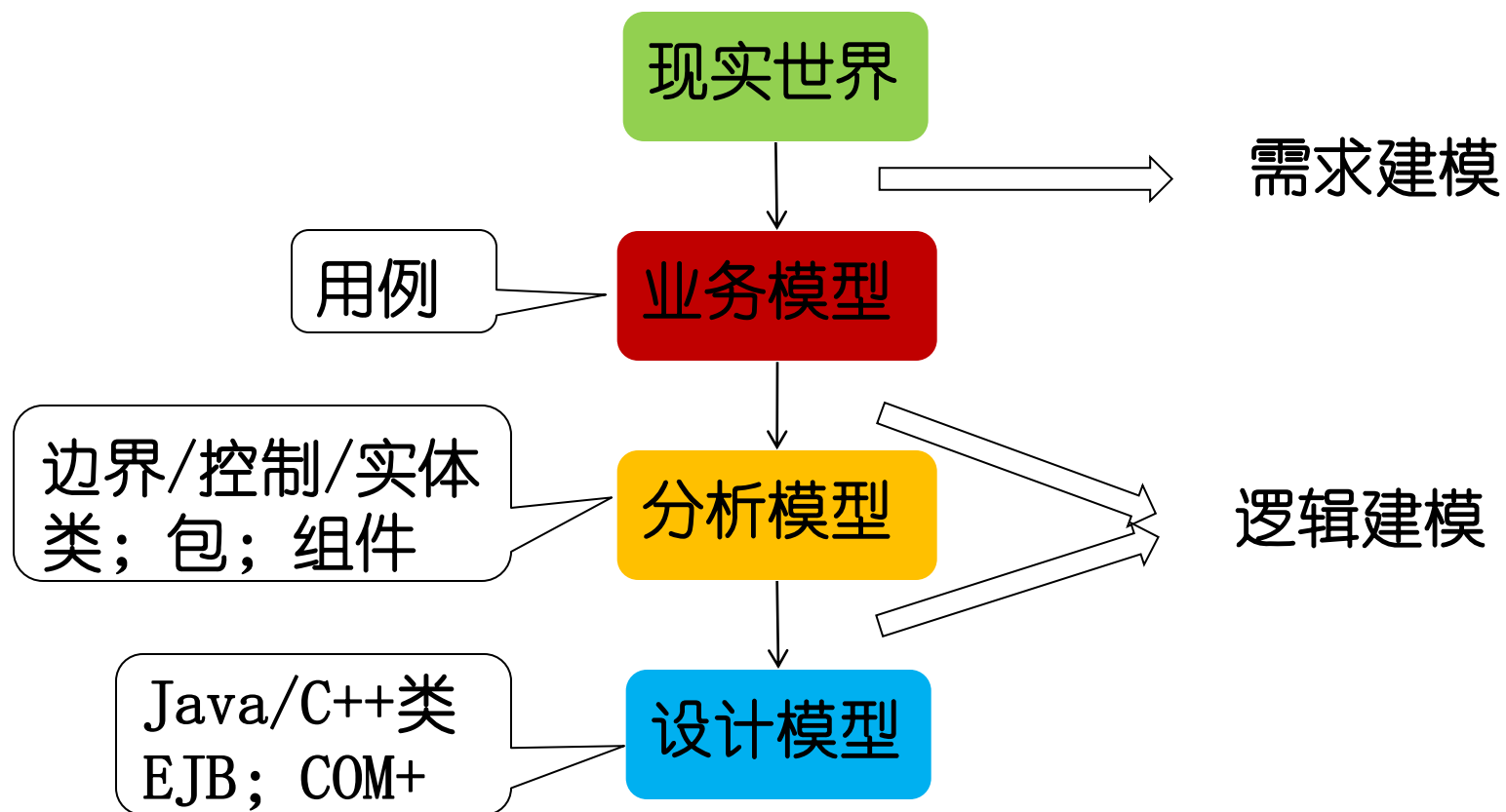




# 建模方法



- 基于UML的建模方法



# 小结



- 重点
  - 了解信息系统建模的特点
  - 建模的四个原则
  - 理解面向对象的建模方法