

UML介绍及建模基础

钟业弘

1601214497

内容提要

► UML

- 什么是UML
- 多种模型之间的转化
- 统一过程

► 建模基础

- 建模
- 抽象层次
- 视图
- 对象分析方法

什么是UML

- ▶ UML:统一建模语言，定义了建立模型需要的基本元素以及元素间的关系。
 - ▶ 目的：形成统一的标准，使得沟通更为容易，功能拆分变得简单。
 - ▶ 特点：可视化，用UML的表示法，可以把文字难以表达清楚的东西通过图形表达出来。

试着用文字表示汽车的组成

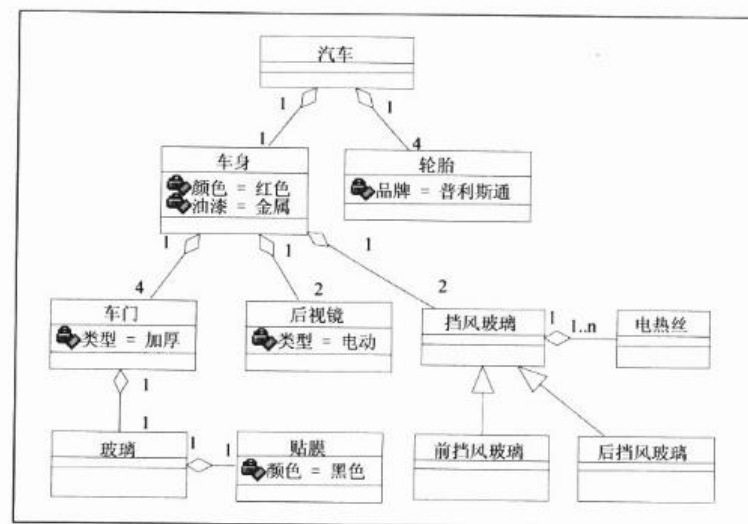


图 1.5 汽车的 UML 表述

多种模型的转化

► 现实世界到业务模型

- 参与者 (actor) : 现实世界的人
- 用例 (use case) : 业务目标

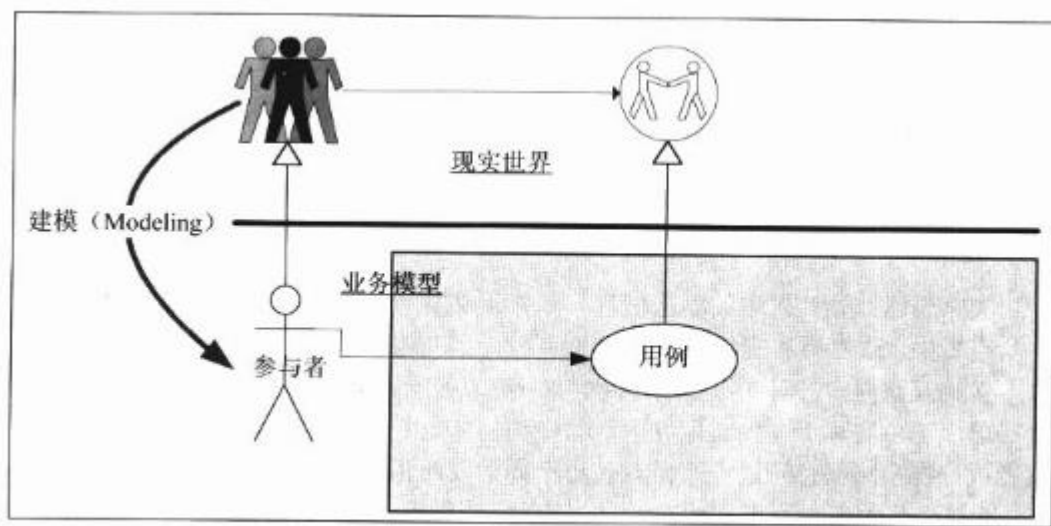


图 1.6 从现实世界到业务模型

多种模型的转化

► 业务模型到概念模型



边界类 (boundary)：任何事物都分里外，外面的事物与里面事物的交互需要边界。边界决定了外面能对里面做什么事



实体类 (entity)：完成业务目标时所涉及的事物。



控制类 (control)：边界和实体都是静态的，不会产生动作，控制类则表示原始需求中的动态信息

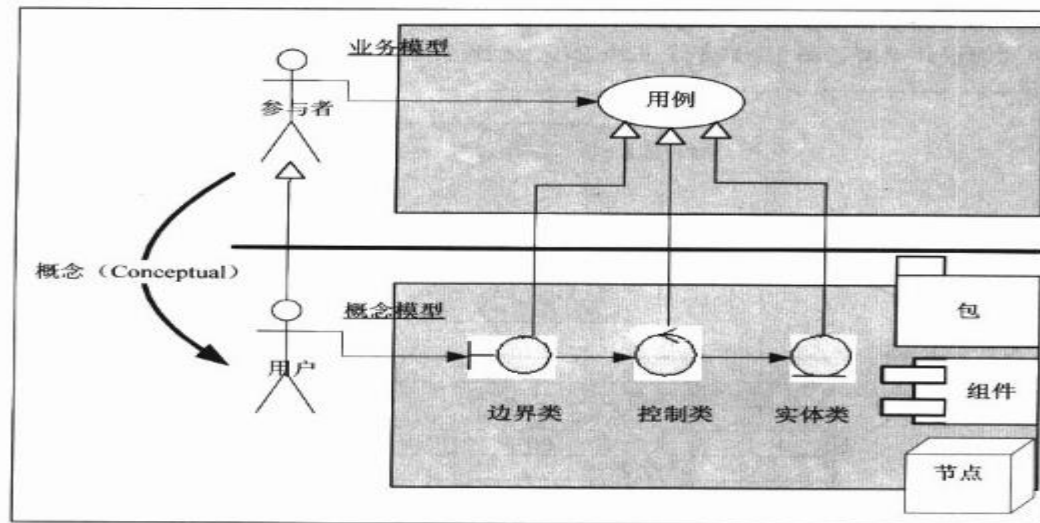


图 1.7 业务模型到概念模型

多种模型的转化

- ▶ 从概念模型到设计模型
- ▶ 设计模型的工作就是建造零部件，组装的过程
 - ▶ 边界类->操作界面或者系统接口
 - ▶ 实体类->数据库表、XML文档等具有持久化特征的类
 - ▶ 控制类->工作流、算法体等

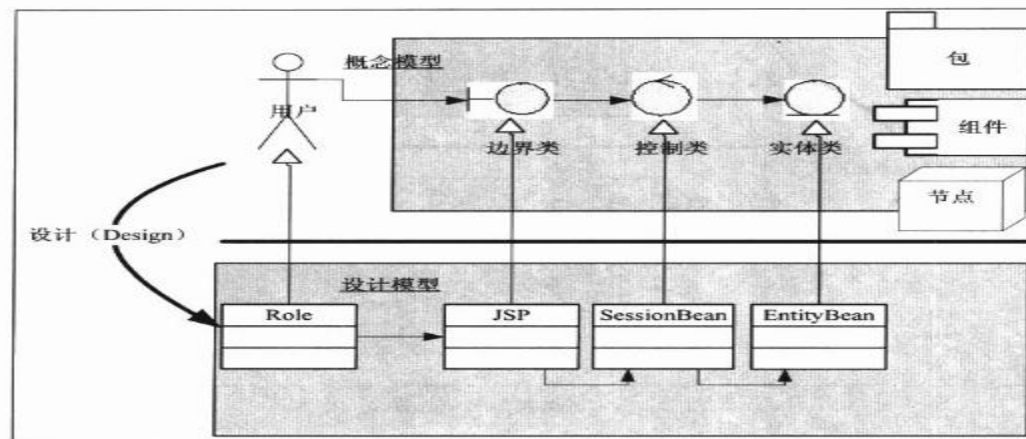
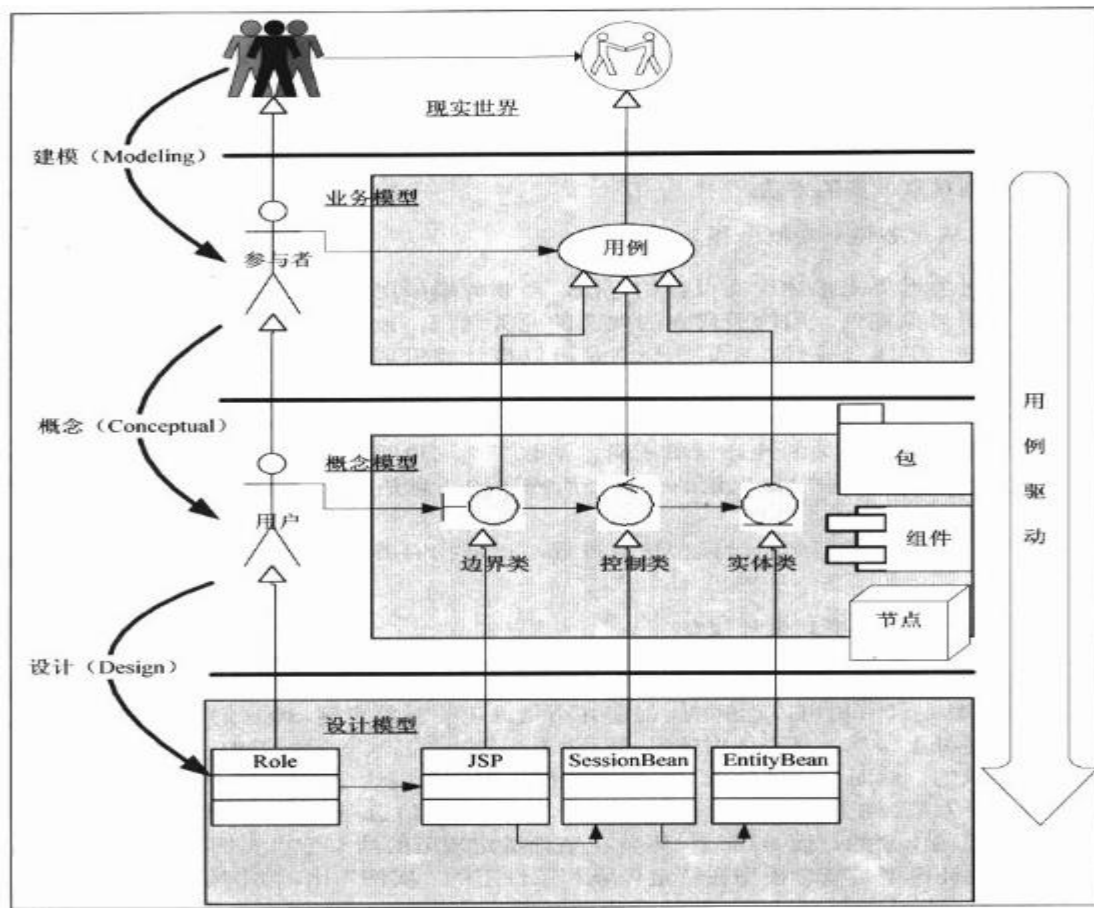


图 1.8 从概念模型到设计模型

总结模型转化

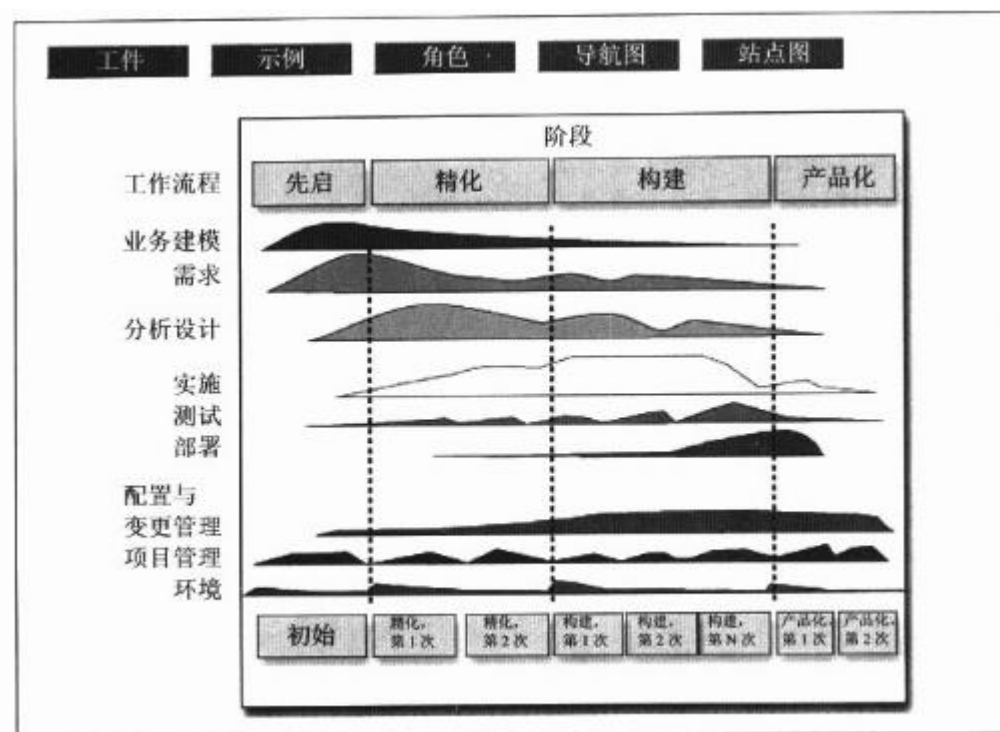


统一过程

- ▶ 统一过程RUP是一种软件开发的方法论。
 - ▶ 适用于大型项目的开发
 - ▶ 归纳和继承了软件开发活动中的最佳实践
 - ▶ 定义了软件开发过程中的重要阶段和工作：4个阶段和9个工作流
 - ▶ 定义了参与软件开发过程的各个角色和职责
 - ▶ 定义了软件生成过程中产生的工件
 - ▶ 采用演进式软件生命周期（迭代）将工作、角色和工件串在一起，形成统一过程

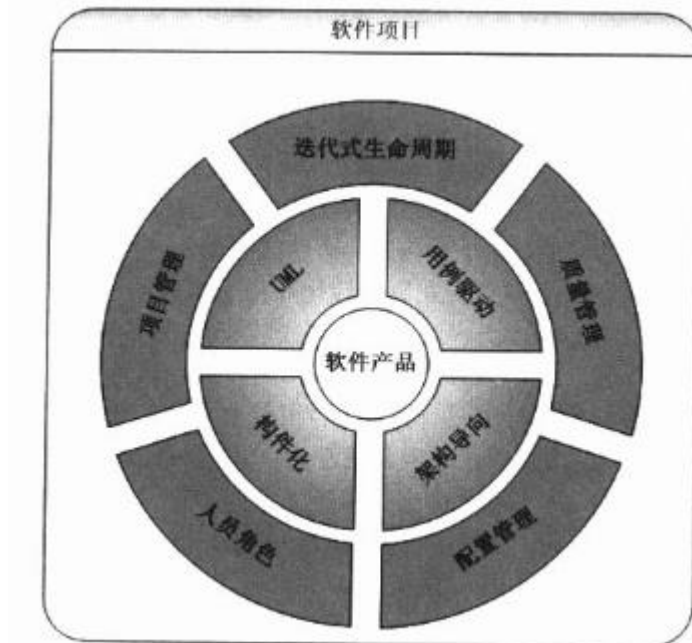
统一过程

► 4大阶段和9个工作流



统一过程

- ▶ RUP和UML就像是音乐理论和五线谱的关系，UML是一种语言，是一种工具，而RUP是一种方法论，两者即相对独立又相辅相成。
- ▶ 统一过程的最佳实践：软件开发精华的总结



建模

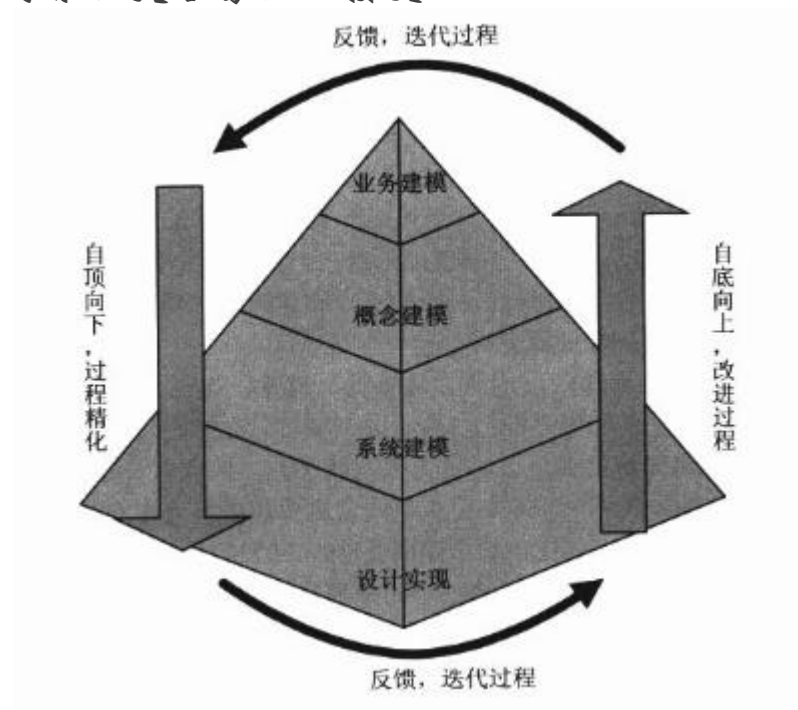
- ▶ 建模：是指通过对客观事物建立一种抽象的方法以表征事物并获得对事物本身的理解，同时将这种理解概念化，将这些逻辑概念组织起来，构成一种对所观察的对象的内部结构及工作原理的便于理解的表达。
- ▶ 怎么建？：选好抽象角度
 - ▶ 面向过程：通盘考虑，使问题变得复杂化
 - ▶ 面向对象：分解成小块，使问题变得简单化
- ▶ 什么是模？：静态的事物（物）+特定的条件（规则）+特定的动作（参与者的驱动）=特定的场景（事件）

抽象层次

- ▶ 抽象层次越高，具体信息越少，概况能力越强；反之，具体信息越丰富，概况能力越弱。
- ▶ 并非信息越多越容易理解，相反，信息量越少越容易理解和处理。这就是面向对象比面向过程具有优势的原因
 - ▶ 关于石头有多硬
 - ▶ 能砸碎玻璃，摸起来的感觉 —— 高层抽象
 - ▶ 摩氏5级 —— 低层抽象
 - ▶ 关于太阳系与银河系中心的距离
 - ▶ 27000光年 —— 高层抽象
 - ▶ 255439722759681600000米 —— 低层抽象
- ▶ 抽象的方法：自顶向下，自底向上。

抽象方法

- ▶ 自顶向下的抽象方法适用于让人们从头开始认识事物
 - ▶ 介绍汽车时，先讲发动机的功能比先讲发动机的原理更容易让人接受
- ▶ 自底向上的抽象方法适用于在实践中改进提高认知。
- ▶ 软件开发主体一般采用自顶向下的方法



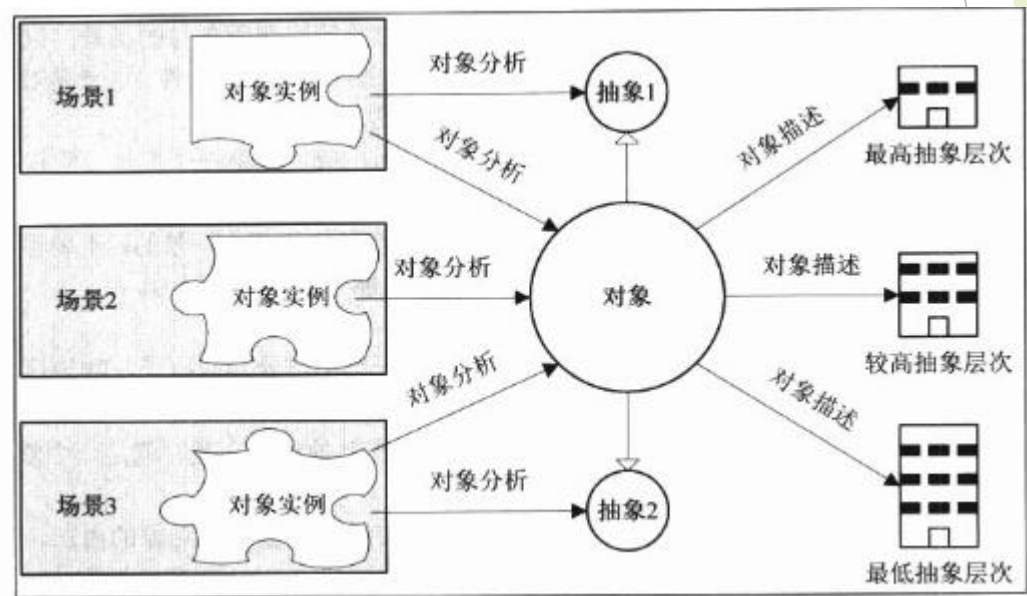
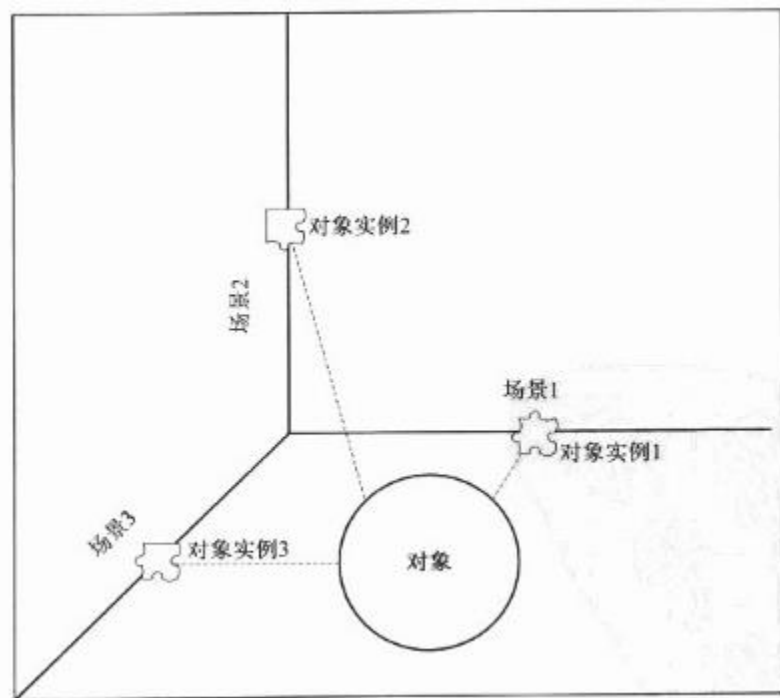
视图

- ▶ 视图：对象的一个方面
- ▶ 视角：观察事物的角度，不同视角观察同一事物会得到不同的视图
- ▶ 建模过程中，要将信息展示给不同的人看，需要不同的视角
 - ▶ 软件干系人：客户、系统分析员、架构师、设计师、开发人员、测试人员、项目经理
 - ▶ 不同的干系人所关心的侧重点不同
- ▶ UML定义了用例图、对象图、类图、包图、活动图等不同的视图，多方面描述软件的结构和组成

对象分析方法

- ▶ 一切皆是对象
- ▶ 对象都是独立的
 - ▶ 对象并不属于某个场景，每个场景看到的仅是对象映射到该场景的一个方面
- ▶ 对象具有原子性
 - ▶ 对象是不可分割的整体，好比一个鸡蛋，永远不要打破蛋壳，我们对鸡蛋的所有理解都只应该来自蛋壳，一旦打破，就会一片混乱。
 - ▶ 要将对对象的认知附加在对象边界上，称之为面向接口编程。
- ▶ 对象都是可抽象的
 - ▶ 对象有很多个方面，参与场景时会表现出一个方面，我们可以把这个方面抽象出来代表对象与场景交互。
- ▶ 对象都是层次性
 - ▶ 视问题的复杂度，将对象划分成多个抽象层次

对象分析方法



► 谢谢