面向对象分析与设计
Object-Oriented Analysis and Design

北京理工大学软件学院
马　锐
Email: mary@bit.edu.cn

第3章 UML2.0概述

3.1 RUP概述

3.2 UML概述

3.1 RUP概述(1)

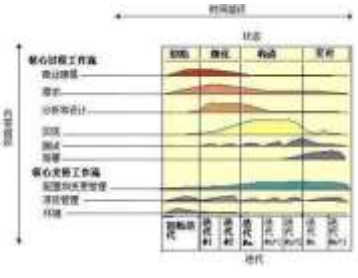
- RUP(Rational Unified Process)：统一过程方法（统一软件开发过程模型）
- 软件开发过程
  - 软件描述
  - 软件开发
  - 软件有效性验证
  - 软件不断改进
- 软件开发模型
  - 对软件开发全过程、活动和任务进行抽象描述

3.1 RUP概述(2)

- RUP的特点
  - 用例驱动（采用用例描述用户需求）
  - 以构架为中心
  - 采用迭代和增量模型
  - 采用UML语言描述软件开发过程
  - 有一系列功能强大的软件工具支撑
- RUP过程框架模型

3.1 RUP概述(3)

4个阶段
9个工作流
多个角色
多个活动
多个工件



3.1 RUP概述(4)

- RUP软件开发过程（4个阶段）
  - 初始阶段
    - 为系统建立商业用例
    - 确定要开发系统的边界
    - 找出与系统交互的所有外部实体
    - 列出实体与系统的交互过程
  - 细化阶段
    - 分析问题领域，准备架构设计
      - 确定系统的范围
      - 主要的功能需求与性能需求
    - 编制项目计划，淘汰高风险因素

3.1 RUP概述(5)

- 构建阶段
  - 完成选择所需要的构件
  - 开发应用程序的主要功能
- 产品化阶段
  - 确保软件对最终用户可用

➢ RUP核心过程工作流

- 业务建模
  - 描述了系统开发的一个构想，使用业务用例模型描述该构想
    - 业务模型定义的过程、角色、责任

3.1 RUP概述(6)

- 需求
  - 描述系统应该做什么，并达成共识
  - 理解系统所解决问题的定义和范围
- 分析设计
  - 将需求转化为未来系统的设计，为系统开发一个健壮的结构
- 实施
  - 通过分层次的组织实现子系统来定义代码结构
  - 用构件的形式实现类

3.1 RUP概述(7)

- 将所开发的构件作为单元进行测试
- 将每个实现人员的工作成果集成到一个可执行的系统中
- 测试
  - 验证对象间的交互是否符合设计要求
  - 验证软件中所有构件是否正确集成
  - 检验所有需求是否正确实现
- 部署
  - 成功生成版本并将软件分发给用户

3.1 RUP概述(8)

➢ RUP核心支持工作流

- 配置与变更管理
  - 描绘了如何在多个成员组成的项目中控制和管理变更
- 项目管理
  - 平衡各种可能产生冲突的目标
- 环境
  - 向软件开发组织提供软件开发环境，包括过程和工具

3.1 RUP概述(9)

➢ RUP的角色

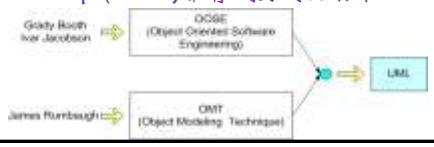
- 分析员角色集
  - 业务流程分析员、业务设计员、业务模型复审员、需求复审员、系统分析员、用户界面设计员
- 开发人员角色集
  - 构架设计师、构架复审员、代码复审员、数据库设计员、系统设计员、设计复审员、实施员、集成员

3.1 RUP概述(10)

- 测试人员角色集
  - 测试设计员、测试员
- 经理角色集
  - 变更控制经理、配置经理、部署经理、流程工程师、项目经理、项目复审员
- 其他角色集

3.2 UML概述(1)

- UML (Unified Modeling Language)
  - 应OOAD的需要而产生
  - 1996年由Grady Booch、James Rumbaugh和Ivar Jacobson提出
  - 1997.11，由Object Management Group (OMG)推荐成为国际标准



13

3.2 UML概述(2)

- UML是一种构建软件系统和文档的通用可视化建模语言，独立于开发过程
- UML能与所有的开发方法一同使用，可用于软件开发的整个生命周期
- UML能表达系统的静态结构和动态信息，并能管理复杂的系统模型，便于软件团队之间的合作开发
- UML不是编程语言，但支持UML语言的工具可以提供从UML到各种编程语言的代码生成，也可以提供从现有程序逆向构建UML模型

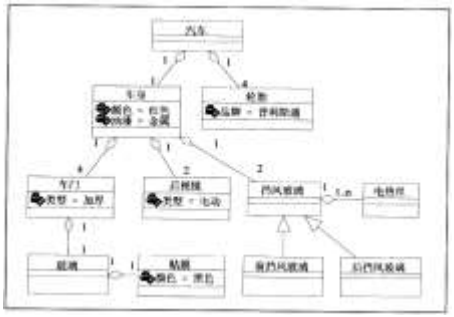
14

3.2 UML概述(3)

- 可视化示例
  - 造一辆车身是红色金属漆的小轿车，装备4个普利司通的轮胎，它是一辆4门车，车身是加厚的，并且前后门玻璃上贴黑色的膜。前后挡风玻璃都装有电热丝，后视镜是电动可调的。
  - 理解
  - 问题
    - 车门装在哪里？车门怎么打开？

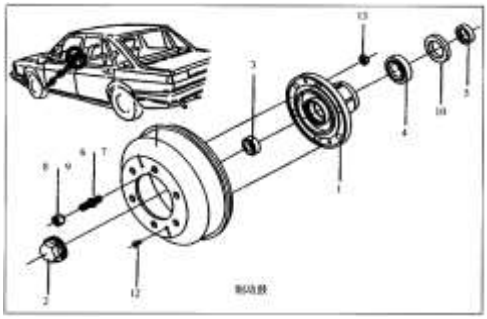
15

3.2 UML概述(4)



16

3.2 UML概述(5)



17

3.2 UML概述(6)

- UML演变
  - Grady Booch的Booch方法
    - 20世纪80年代提出，90年代成熟
    - 区分系统的逻辑和物理结构并描述这两种结构的静态和动态语义
    - 描述系统动态行为和静态属性，从宏观和微观两个角度描述系统演化过程
      - 通过微观识别对象、类、对象语义、对象及类间关系，实现类与对象

18

3.2 UML概述(7)

- 宏观开发过程是微观开发过程的控制框架
  - 在宏观上完成建立需求（概念）、建立行为模型（分析）、产生设计架构（设计）、导出实现（发展），管理交付的产品（维护）等整个软件过程
- 提供了丰富的面向对象概念
  - 类、对象、继承、消息、操作
- 提供了一些模型
  - 类图、对象图、顺序图、状态图

19

3.2 UML概述(8)

- Rumbaug提出的OMT方法
  - Object Modeling Technique
  - Loomis、Shan和Rumbaugh于1987年提出，用于关系数据库设计
  - 1991年J. Rumbaugh在实体关系模型基础上扩展了类继承、行为等概念
  - 系统开发过程分为分析、设计（系统设计、对象设计）和实现
  - 提供了对象、动态、功能三个模型

20

3.2 UML概述(9)

- Ivar Jacobson提出的OOSE方法
  - Object Oriented Software Engineering
  - 1992年提出
  - 强调通过用例(Use Case)分析系统实际操作，并以此作为切入点，逐步展开系统分析过程
  - 采用模型驱动方法，开发过程围绕需求、分析、设计、实现、测试等5个模型的建立进行

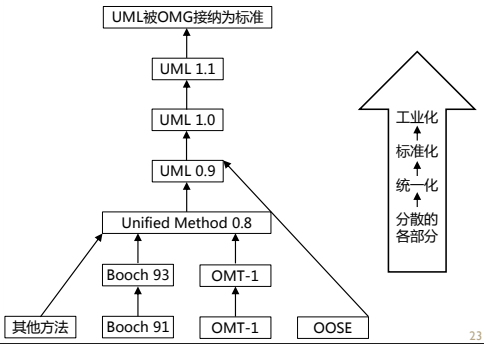
21

3.2 UML概述(10)

- 需求模型：特色部分，从用户使用角度完整地描述系统功能需求
- 分析模型：需求模型中用例的结构化抽象
- 设计模型：分析模型进一步精化，并考虑软件实现环境，定义实现详细策略和机制
- 实现模型：实现模块的代码
- 测试模型：单元测试直到系统测试

22

3.2 UML概述(11)



23

3.2 UML概述(12)

- UML组成
  - 静态结构
    - 详细描述系统中主要对象的属性和方法，以及对象之间的关系
  - 动态结构
    - 详细描述系统中重要对象的时间特性和对象间为完成某个目标而相互进行通信的机制
- UML定义
  - UML语义

24

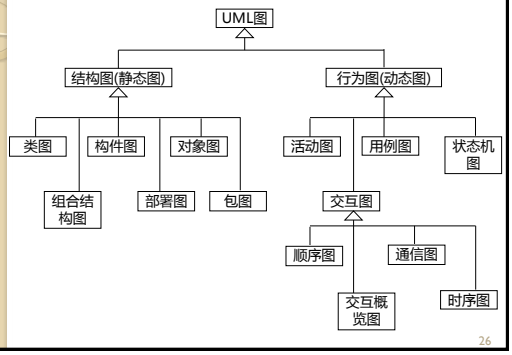
3.2 UML概述(13)

- 描述基于UML的精确元模型定义
- UML表示法
  - 定义了UML符号的具体表示法，为开发者或开发工具使用这些图形符号和文本语法、为系统建模提供了统一的标准

➢ UML模型图
- 提供了5类，共13种图用于建模：用例图、静态图（类图、对象图、包图）、行为图（状态机图、活动图）、交互图（顺序图、通信图、时序图、交互概览图）、实现图（构件图、部署图、组合结构图）

25

3.2 UML概述(14)



26

3.2 UML概述(15)

➢ 结构图(structure diagram)：静态建模
- 类图(class diagram)：描述系统中各个对象的类型以及其间存在的各种关系
- 对象图(object diagram)：描述在某一时刻一组对象以及它们之间的关系
- 构件图(component diagram)：描述构件的组织机构和相互关系，用于表达如何在实现时把系统元素组织成构件，支持以构件为单位进行软件制品的实现与发布

27

3.2 UML概述(16)

- 部署图(deployment diagram)：描述节点、节点间的关系以及构件和节点间的部署关系
- 包图(package diagram)：描述模型元素分组（包）以及分组之间依赖的图
- 组合结构图(composite structure diagram)：描述类和构件的内部结构，其中包括与系统其他部分的交互点

➢ 行为图(behavior diagram)：动态建模
- 用例图(use case diagram)：描述一组用例和参与者以及它们之间的关系的图

28

3.2 UML概述(17)

- 活动图(activity diagram)：描述活动、活动的执行顺序以及活动的输入与输出
- 状态机图(state diagram)：描述一个对象（或其他实体）在其生存期内所经历的各种状态以及状态变迁
- 交互图(interaction diagram)：是顺序图、交互概览图、通信图和时序图的统称
  - 顺序图(sequence diagram)：描述一组角色和由扮演这些角色的实例发送和接收的消息

29

3.2 UML概述(18)

- 交互概览图(interaction overview diagram)：以一种活动图的变种来描述交互的图，关注于对控制流的概览
- 通信图(communication diagram)：描述一组角色、角色间的连接件以及由扮演这些角色的实例所发出的消息
- 定时图(timing diagram)：描述在线性时间上对象的状态或条件变化

30