

# UML建模基础

## —UML概述

东软IT人才实训中心

# 课程结构

内容	课时 (H)
第一章： <b>OO</b> 基础、 <b>UML</b> 概述	<b>1</b>
第二章： 用例图	<b>1</b>
第三章： 类图	<b>2</b>
第四章： 交互图	<b>1.5</b>
第五章： 状态图和活动图	<b>0.5</b>
第六章： 组件图和部署图	<b>0.5</b>
第七章： 正向工程和逆向工程	<b>0.5</b>

# 第一章：OO基础、UML概述

目标：

本章旨在向学员简要介绍UML建模的重要性、UML的概念模型，通过本课的学习，学员应该掌握如下知识：

- 1) 了解UML的概念模型
- 2) 简要介绍UML的“4+1view”

学时：1学时

教学方法：讲授ppt  
+上机练习

# 1.1 OO基础

- OO的几个基本概念
  - 对象
  - 类
  - 属性
  - 方法
  - .....

# 1.1 OO基础(续)

- 对象
  - 对象是指封装了事物的状态（State）和行为（Behavior）的一个实体。可以是物理的（如一个卡车）、概念的（如一个化学过程）或软件的（如一个链表）

# 1.1 OO基础(续)

- 状态
  - 是对象可以处于的状况
  - 对象的状态随时间变化
  - 用属性 (Attribute) 和关系 (relation) 表示
- 行为
  - 行为决定对象如何动作和做反应
  - 对象的可见行为用一系列它响应的消息来模型化
  - 用操作 (Operation)、方法 (Method) 和状态机 (State machine) 表示
- 标识 (Identity)
  - 每个对象有唯一的标识

# 1.1 OO基础(续)

- 类是对一系列具有相同属性、操作、关系和语义的对象的描述
  - 对象是类的实例
  - 类定义了它的所有对象的结构和行为的模板

# 1.1 OO基础(续)

- 面向对象
  - OO的三大特征
    - 封装 (Encapsulation)
    - 继承 (Inheritance)
    - 多态 (Polymorphism)
  - OO包括的三大部分
    - 面向对象的分析 (OOA)
    - 面向对象的设计 (OOD)
    - 面向对象的编程 (OOP)



# 1.2 UML概述

## – 什么是UML?

- UML: 统一建模语言  
Unified Modeling Language
- UML是由Rational公司三位世界级面向对象技术专家Grady Booch, Ivar Jacobson和Jim Rumbaugh 提出的。
- UML是一种标准的图形化建模语言，它是面向对象分析与设计的一种标准表示。

# 1.2 UML概述(续)

- 什么是模型?
  - 模型就是真实世界的简化
  - 为我们提供一个系统的原型
- 为什么要建模?
  - 为了更好的理解我们将要或正在开发的系统
  - 因为我们通常无法理解一个复杂系统的全部
  - 模型能为我们做什么?
    - 帮助我们对系统进行可视化
    - 允许我们详细说明系统的结构或行为
    - 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

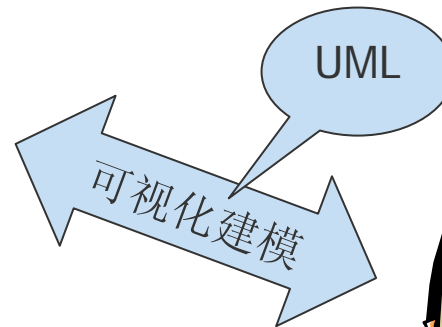
## 1.2 UML概述(续)

- 什么是可视化建模?

“建模获取系统的关键部分”



业务流程

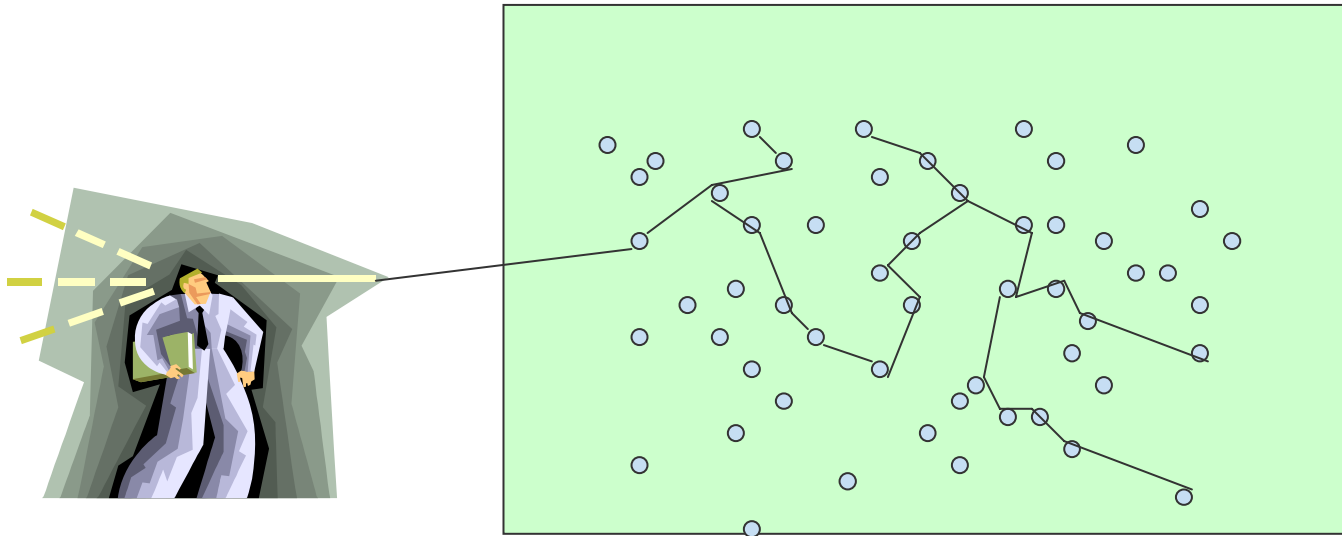


计算机系统

可视化建模就是用标准的  
图形表示法来建模

# 1.3 可视化建模的作用

- 可视化建模获取业务流程
  - 用例（use case）分析是一种从用户的角度获取业务流程的技术
  - 使用相同的语言，不至于产生歧义
  - 用例分析能让分析师在构建系统之前理解要构建什么



# 1.3 可视化建模的作用(续)

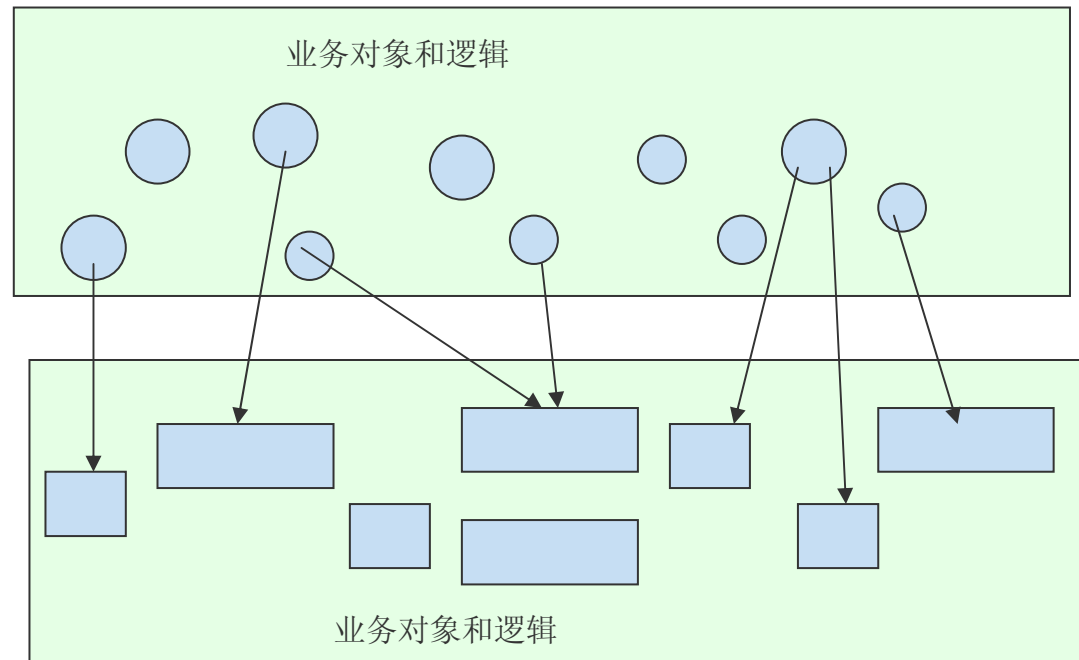
- 可视化建模是一个交流工具



业务领域

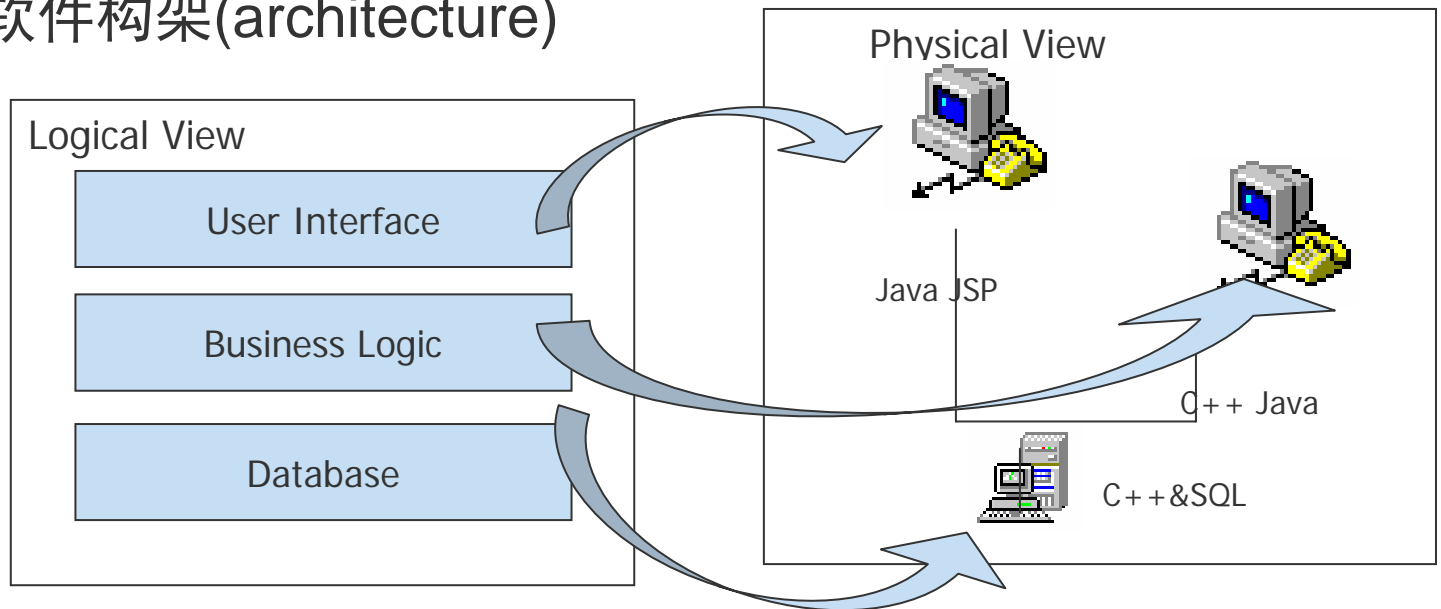


计算机领域



# 1.3 可视化建模的作用(续)

- 管理复杂性
  - 把3000多个类放在一张图中不好
  - 可视化建模的“包”(package)
    - 把元素模型化成有意义的组合
    - 为不同的人提供不同级别的抽象
  - 软件构架(architecture)



# 1.3 可视化建模的作用(续)

- 促进复用(reuse)
  - 复用是软件的“圣杯”
  - 不止是复用代码，而是复用建立原始工件时需要的所有分析、设计、实现、测试、文档化
  - 可以有一个类复用、多个类（或一个组件）的复用、应用模式等复用方式
  - 可视化建模让你从复用的角度看，如果想复用工件，什么是可用的

# 1.4 UML的概念模型

- UML的概念模型
  - UML建模的三个主要元素
    - 构造块：事物、关系、图
    - 规则：命名、范围、可见性、完整性、执行
    - 公共机制：规范说明、通用划分、扩展机制

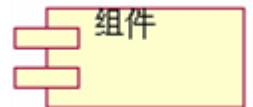
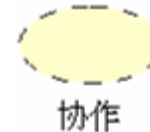
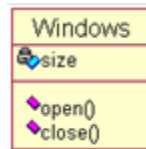


# 1.4 UML的概念模型(续)

- UML元素 – 构造块
  - 事物
    - 对模型中最具有代表性的成分的抽象
  - 关系
    - 把事物结合在一起
  - 图
    - 聚集了相关的事物

# 1.4 UML的概念模型(续)

- UML元素 – 构造块 – 事物
  - 结构事物：通常是UML模型的静态部分，描述概念或物理元素
    - 类
      - 用例：通常代表一个需求
    - 接口
    - 协作：表示一个用例的实现
    - 主动类：至少拥有一个进程或线程的类
    - 组件
    - 节点：运行时存在的物理元素，如一个设备



# 1.4 UML的概念模型(续)

- UML元素 – 构造块 – 事物 (续)

- 行为事物：是UML模型的动态部分，是模型中的动词

- 交互 (interaction)：可描述一个对象群体的行为或单个操作的行为

显示

- 状态机 (state machine)：可描述单个类或一组类之间协作的行为

等待

- 分组事物：是UML中的组织部分

- 包 (package)

- 注释事物：是UML中的注释部分

- 注解 (note)

包

注释

# 1.4 UML的概念模型(续)

- UML元素 – 构造块 – 关系

- 关系

- 依赖 (dependency) : 一个事物发生变化会影响到另一个事物。



- 关联 (association) : 描述对象之间的连接。聚合。



- 泛化 (generalization) : 特殊/一般关系, 也就是继承。



- 实现 (realization)

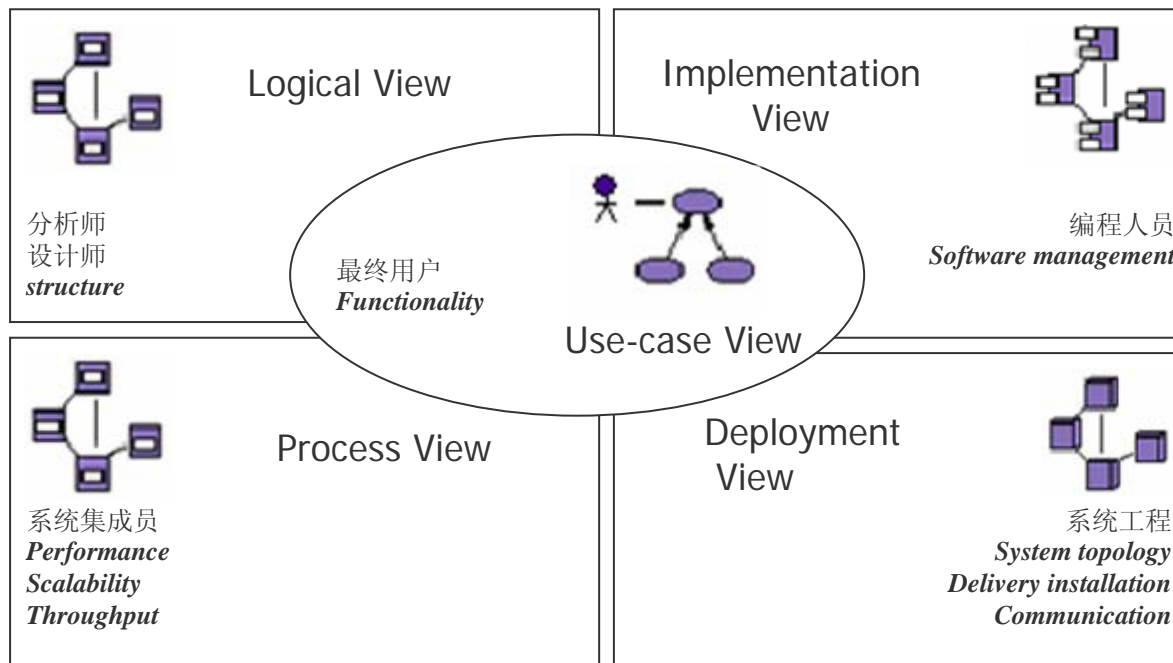


# 1.4 UML的概念模型(续)

- UML元素 – 构造块 – 图 (Diagram)
  - 用例图 (Use Case Diagram)：模型化系统与外界的交互
  - 类图 (Class Diagram)：模型化系统的结构
  - 时序图 (Sequence Diagram)：模型化系统的行为
  - 协作图 (Collaboration Diagram)：模型化系统的行为
  - 组件图 (Component Diagram)：模型化组件的组织 and 依赖
  - 部署图 (Deployment Diagram)：模型化系统的硬件分布
  - 活动图 (Activity Diagram)：模型化系统内的事件流
  - 状态图 (Statechart Diagram)：模型化状态相关的方面

# 1.5 UML的“4+1view”

- 模型由不同的view和diagram构建而成，描述了不同的视点和系统的构建块
- View用来表示被建模系统的各个方面



# 1.5 Rose中的“4+1view”（续）

View	作用	模型	图
Use Case View	是其他view的“心脏”，说明系统做什么	Use Case Model	Use Case Diagram Activity Diagram（可选） Statechart Diagram（可选）
Logical View	主要描述实现系统内部功能的设计	Analysis Model（可选） Design Model	Class Diagram Sequence Diagram Collaboration Diagram Activity Diagram（可选） Statechart Diagram（可选）
Process View	阐述系统的性能、伸缩性和吞吐量	Process Model	Class Diagram
Implementation View	以分包、分层和配置管理的形式描述了静态软件模型的组织	Implementation Model	Component Diagram
Deployment View	如何把执行体和其他运行时组件映射到下层平台或计算节点	Deployment Model	Deployment Diagram

# 1.6 小结

- UML是一个通用的标准建模语言，可对任何具有静态结构和动态行为的系统进行建模。
- UML适用于系统开发的不同阶段：
  - 需求分析阶段用用例；
  - 分析阶段用类图；
  - 实现阶段用动态模型；
  - 构造阶段用OO编程语言。
  - 测试阶段：单元测试用类图；集成测试用部件图和协作图；系统测试用例图。
- 在应用标准建模语言UML时，需要有相应的开发工具来支持。



# 1.7 术语

缩语、术语	英文全称	解 释
UML	Unified Modeling Language	统一建模语言
OO	Object-Oriented	面向对象
OOA	Object-Oriented Analysis	面向对象分析
OOD	Object-Oriented Design	面向对象设计
OOP	Object-Oriented Program	面向对象编程

# Neusoft

Beyond Technology