

软件工程 模型与方法

Models & Methods of SE

面向对象需求分析方法 - UML简介

本章内容

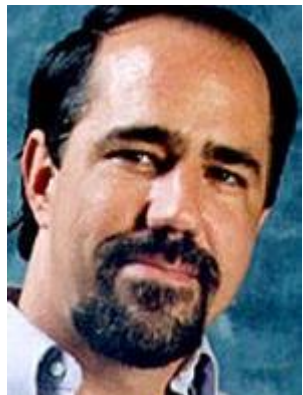
- UML：统一建模语言简介
- 面向对象的需求分析建模
 - 领域建模
 - 领域模型的定义和表示
 - 业务背景：概念类及关系，类图
 - 业务流程：活动图
 - 用例建模
 - 用例图
 - 用例说明
 - 系统顺序图
 - 操作契约

UML 简介

- UML的发展历程
- UML概述
- UML中的组成
- UML中的图

UML的发展历程

- 统一建模语言UML是由Grady Booch、Ivar Jacobson和James Rumbaugh发起，在Booch方法、OOSE方法和OMT方法基础上，广泛征求意见，集众家之长，几经修改而成的一个面向对象的建模语言。
- 该建模语言得到了“UML 伙伴联盟”的应用，并得到工业界的广泛支持，由OMG 组织采纳作为业界标准，是软件界第一个统一的建模语言。



UML发展的四个阶段

- 各自为政：上世纪80年代到1993年期间，面向对象方法出现了百家争鸣的局面，但是不同方法的模型相互转换几乎不可能；
- 统一阶段：1994年10月开始，Rational公司的Booch、Rumbaugh和Jacobson在Booch、OMT和OOSE方法的基础上进行研究，于1996年发布了统一建模语言UML（Unified Modeling Language）；
- 标准化阶段：OMG为了使UML标准更加完善，发布了征求建议书（RFP），随后，Rational软件有限公司建立了UML Partners联盟，各软件开发商和系统集成商共同努力，1997年制定出UML1.1标准，被OMG采纳；
- 工业界应用：1998年OMG接管了UML标准的维护工作，UML已成为软件工业界事实上的标准。

UML的主要特点

- 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，它
 - 是一种可视化的建模语言，不是一种可视化的程序设计语言；
 - 是一种建模语言规格说明，不是工具或知识库的规格说明；

UML 的基本结构

- 基本构造块 Basic building block
 - 事物 Thing
 - 关系 Relationship
 - 图 Diagram
- 语义规则 Rule
 - name、scope、visibility、integrity、execution
- 通用机制 Common mechanism
 - specification、adornment、common division、extensibility mechanism

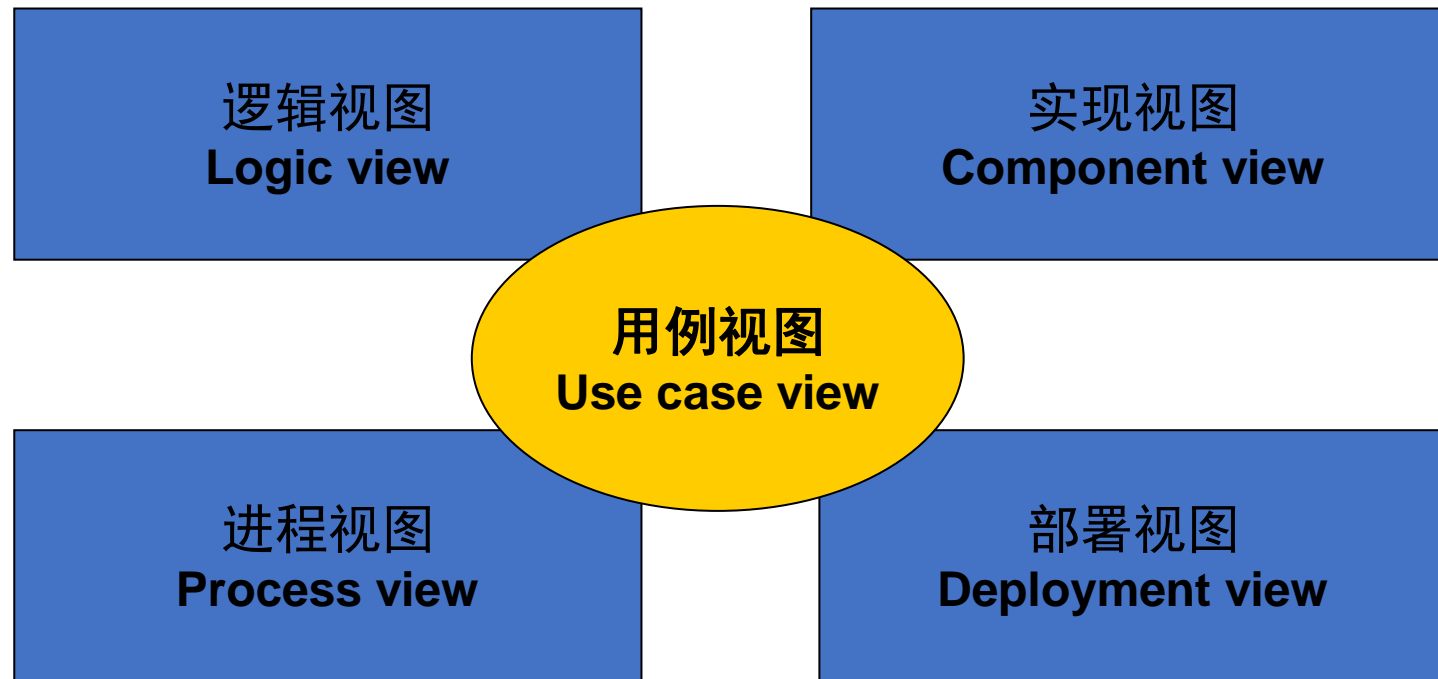
UML 的基本结构

- Structural thing
 - Class, interface, component, node
- Behavior thing
 - use case, Interaction, state machine
- Group thing
 - package
- Annotation thing
 - note

- Dependency
- Association
- Generalization
- Realization

UML的 4+1 视图

- UML 用模型来描述系统的结构（静态特征）以及行为（动态特征）。从不同的视角为系统的架构建模，形成系统的不同视图（view），称为4+1视图，从1个需求的角度出发描述与系统设计的4个维度之间的关系。



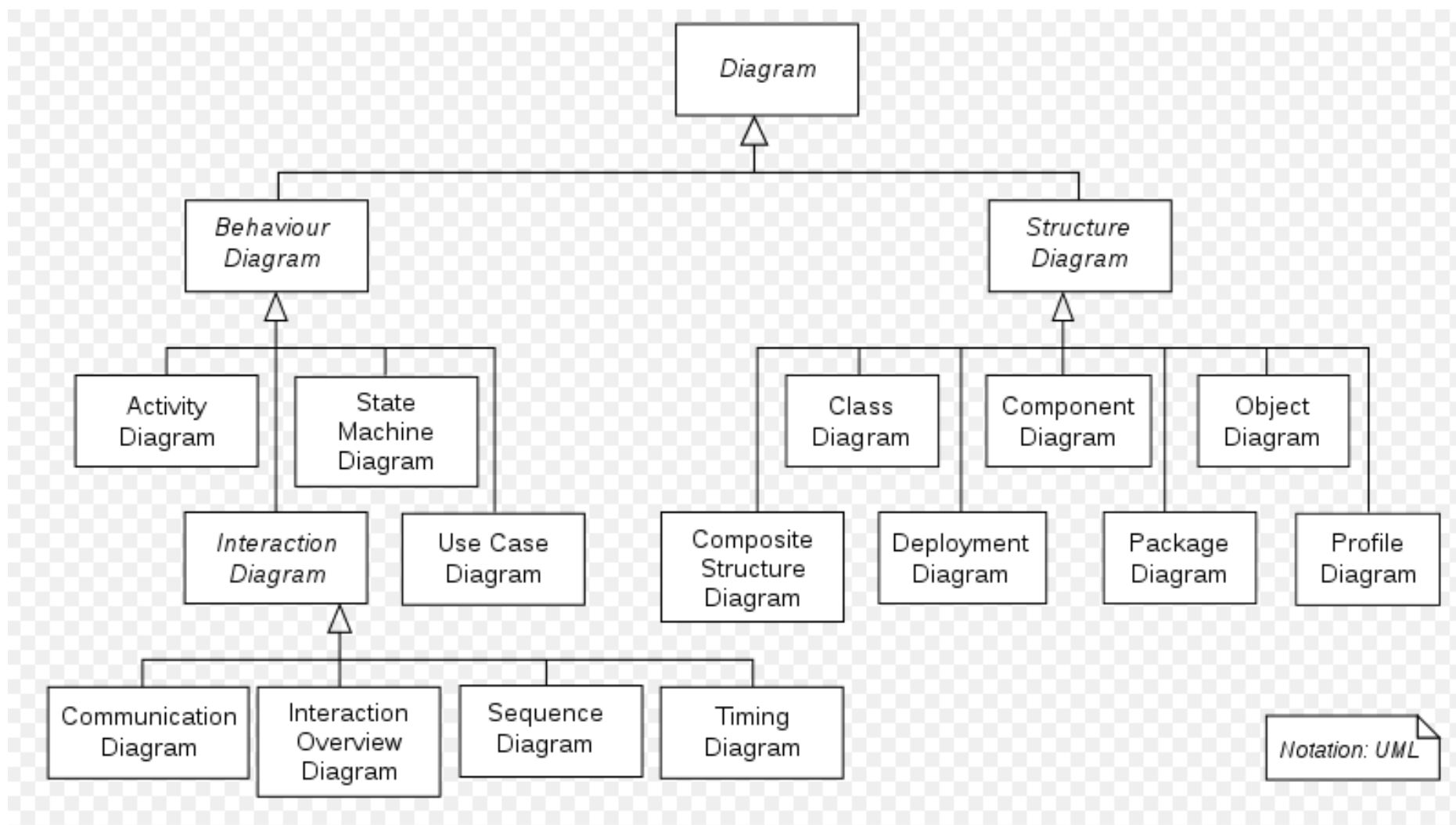
UML 4+1视图的作用

- **用例视图**：表示功能需求，强调从用户的角度看到的或需要的系统功能。这种视图也叫做用户模型视图（user model view）或场景视图（scenario view）；
- **逻辑视图**：展现软件系统设计的静态结构的组成及特征，也称为结构模型视图（structural model view）或静态视图（static view）；
- **进程视图**：描述软件系统设计的并发和同步等特性，关注系统非功能性需求，也称为行为模型视图（behavioral model view）、过程视图（process view）、协作视图（collaborative view）和动态视图（dynamic view）；
- **构件视图**：关注软件代码的静态组织与管理，也称为实现模型视图（implementation model view）和开发视图（development view）；
- **部署视图**：描述硬件的拓扑结构以及软件和硬件的映射问题，关注系统非功能性需求（性能、可靠性等），也称为环境模型视图或物理视图（physical view）；

UML的9个基本图

- 用例图 (Use case diagram) : (从用户的角度) 描述系统的功能;
- 类图 (Class diagram) : 描述系统的静态结构 (类及其相互关系) ;
- 对象图 (Object diagram) : 描述系统在某个时刻的静态结构 (对象及其相互关系) ;
- 顺序图 (Sequence diagram) : 按时间顺序描述系统元素间的交互;
- 协作图 (Collaboration diagram) : 按照时间和空间的顺序描述系统元素间的交互和它们之间的关系;
- 状态图 (State diagram) : 描述了系统元素 (对象) 的状态条件和响应;
- 活动图 (Activity diagram) : 描述了系统元素之间的活动;
- 构件图 (Component diagram) : 描述了实现系统的元素 (类或包) 组织;
- 部署图 (Deployment diagram) : 描述了环境元素的配置并把实现系统的元素映射到配置上。

UML 图的关系



UML视图与图的关系

- 用例视图：使用用例图；
- 逻辑视图：使用类图、对象图，顺序图/协作图；
- 进程视图：使用状态图和活动图；
- 构件视图：使用构件图；
- 部署视图：使用部署图。