

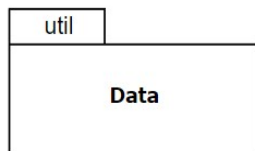
其他UML产物

Other UML Artifacts

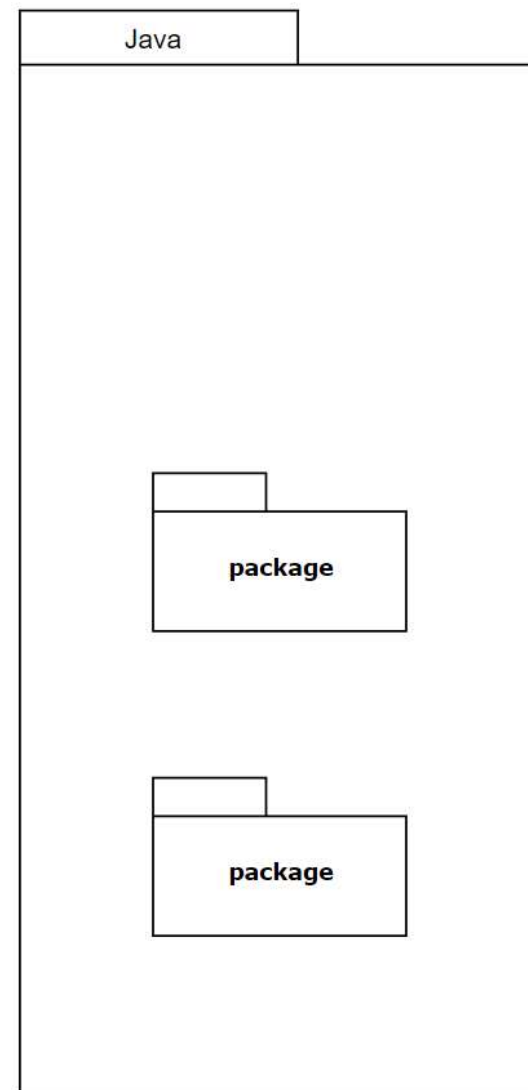
- 包图
- 组件图和部署图

包图

Package

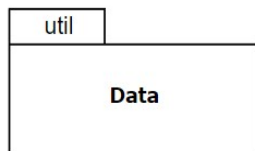


- 一个用来将模型单元分组的通用机制
- 将模型元素分组隐藏，从而简化UML图

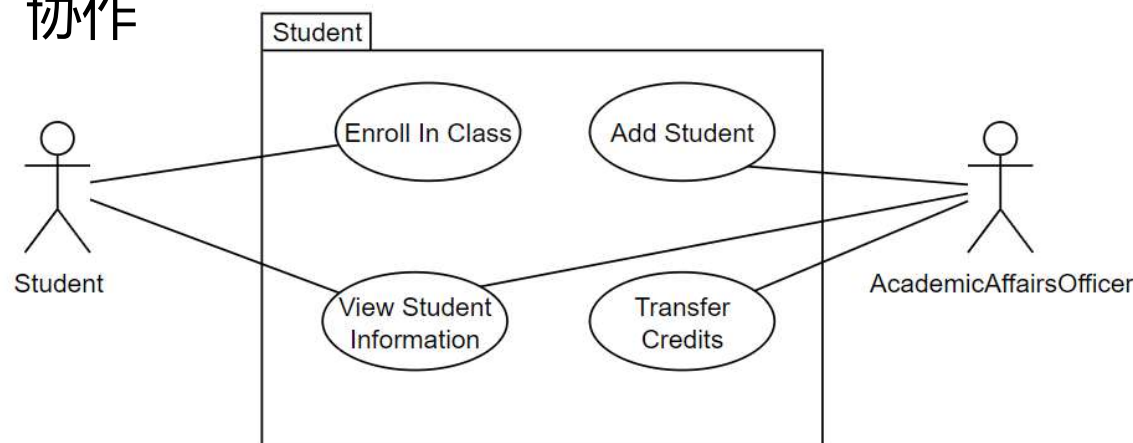
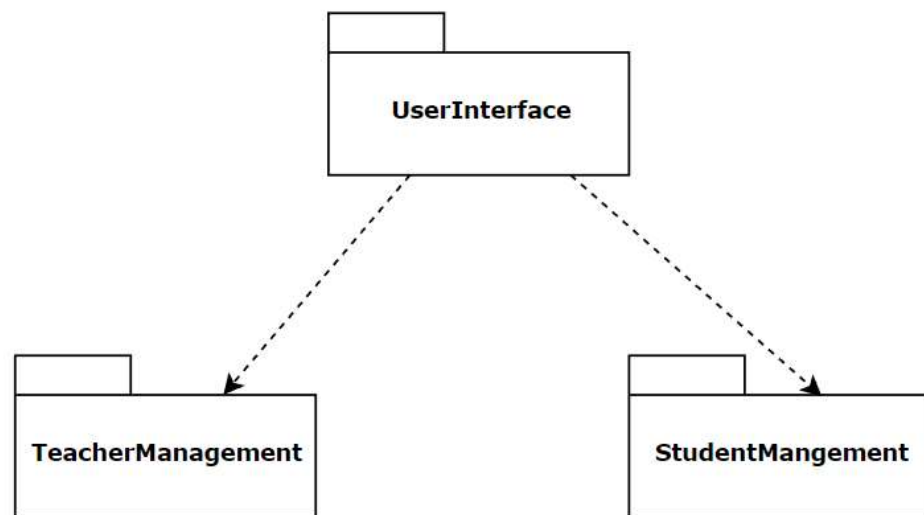


嵌套的包

包图 Package



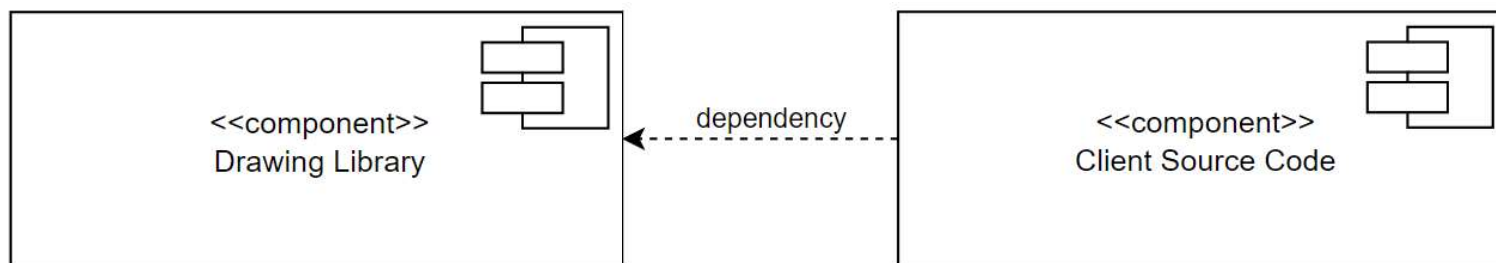
- 一个用来将模型单元分组的通用机制
- 将模型元素分组隐藏，从而简化UML图
- 多用于类图和用例图（类包图和用例包图）
- 将协作较多的用例或类分到一个包里，协作多可以从顺序图或通信图中看出



组件图

Component Diagram

- 实现类型的模型图，更注重系统物理架构的概况
- 展示系统的组件，如文件、包、共享库、数据库间的相关关系
- 各组件在其他UML图内作更详细地描述

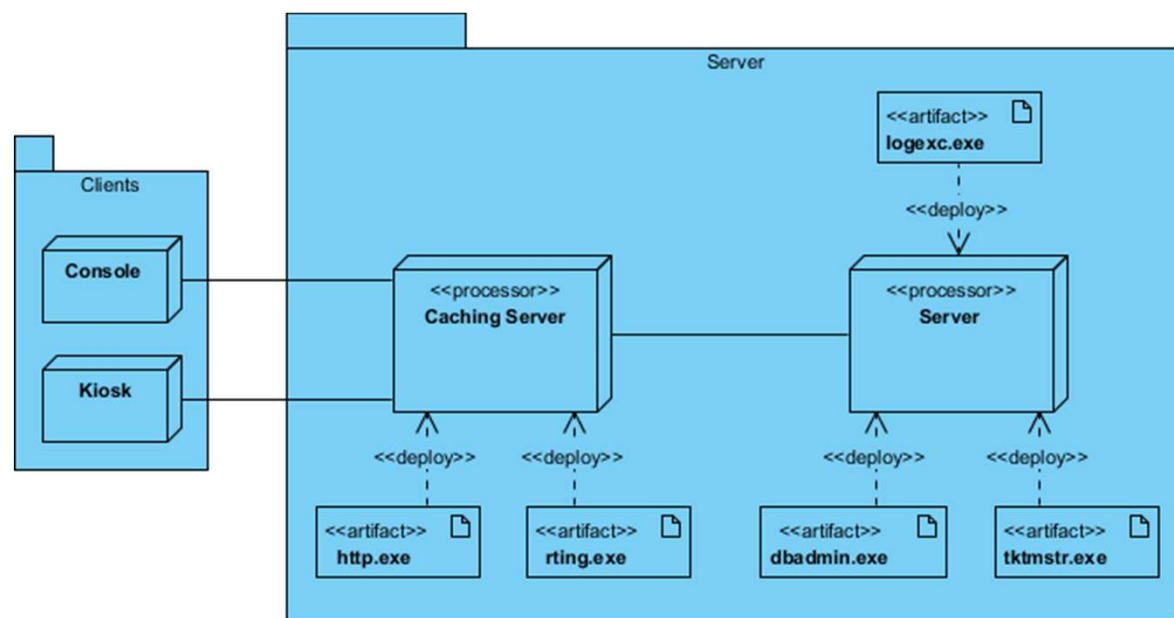


组件的UML符号

部署图

Deployment Diagram

- 也是实现类型的模型图，更注重系统物理架构的概况
- 展示系统的组件，如文件、包、共享库、数据库间的相关关系
- 各组件在其他UML图内作更详细地描述



组件的UML符号

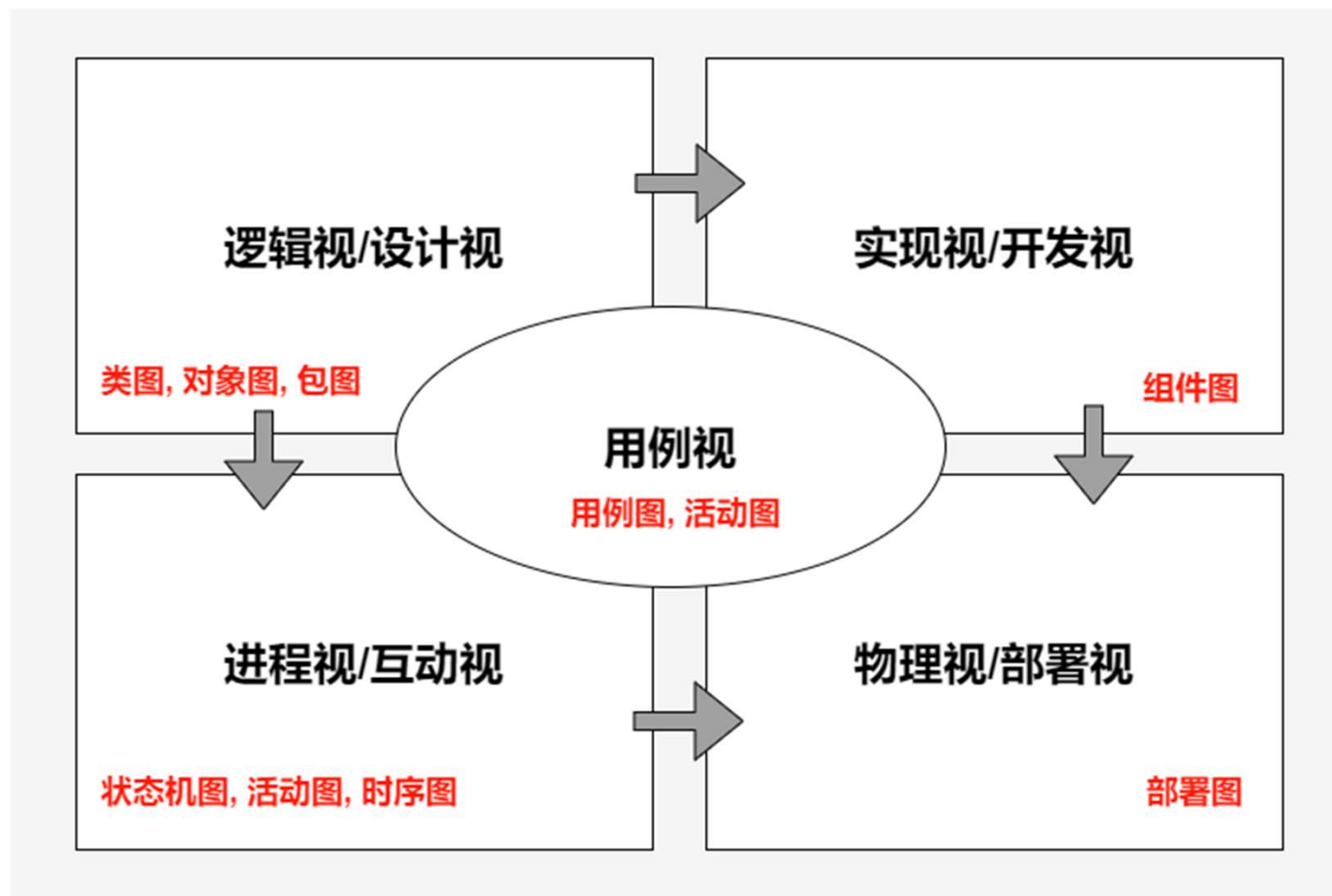
4 + 1 视图

4+1 Concepts

4+1视图

4+1视图是一种软件架构设计方法

- 提供多个视角来描述软件系统的不同方面，以便更全面地理解系统的结构和行为
- 每个视图都关注系统的不同方面



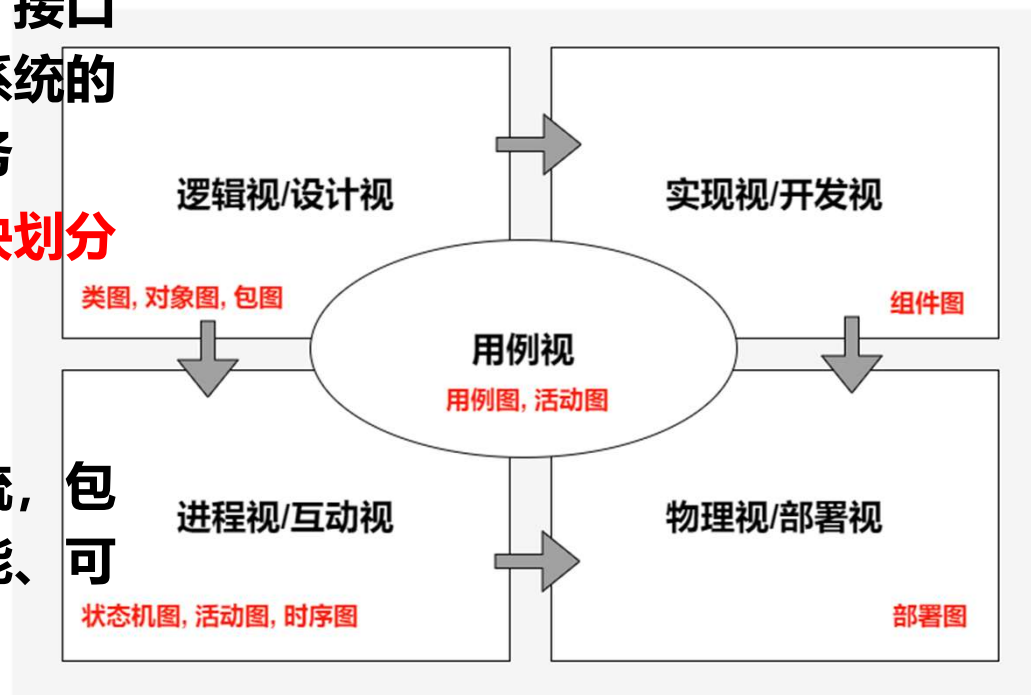
4+1视图

• 设计视 (Design View)

- 系统的设计视包括类、接口、和协作，这些类、接口和协作组成了问题域词汇表和解决方案，支持系统的功能需求，也即系统应该提供给最终用户的服务
- 描述系统的功能抽象。主要描述系统的功能模块划分和模块之间的关系

• 交互视 (Interaction View)

- 系统的互动视描述了系统不同部分之间的控制流，包括可能的并发和同步机制。它体现了系统的性能、扩展性、和总处理能力
- 主要描述系统运行时的非功能性属性



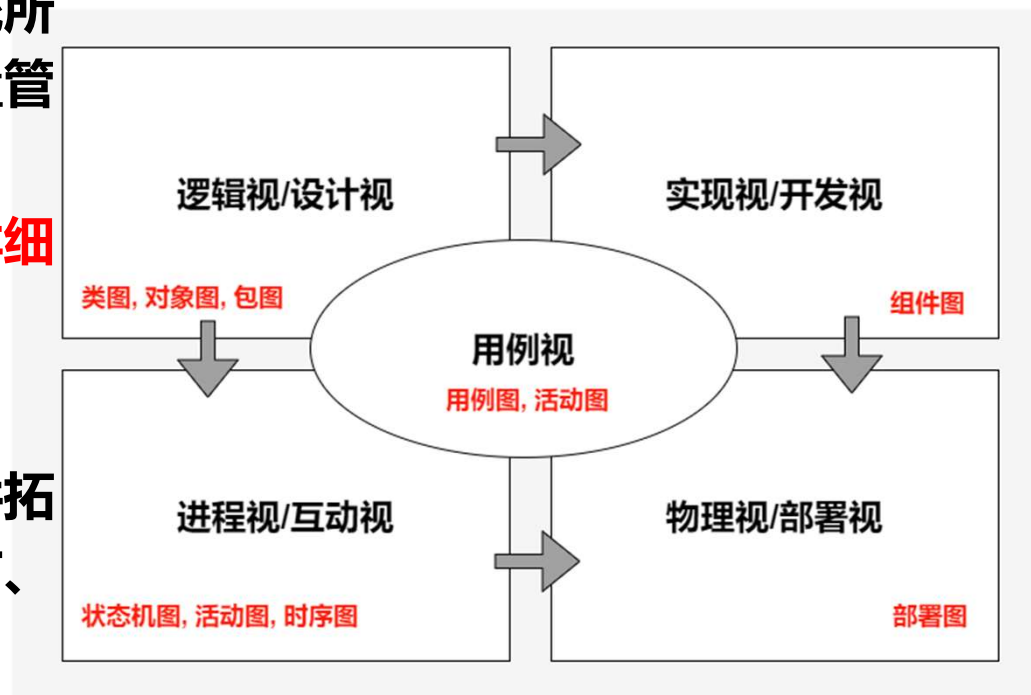
4+1视图

- **实现视 (Implementation View)**

- 系统的实现视包括用于组装、发布物理软件系统所需的各种产物，主要描述了软件系统版本的配置管理
- 系统的详细设计和构造抽象。主要给出了系统详细设计和构造的结构

- **部署视 (Deployment View)**

- 部署视包括了构成用于运行软件系统的系统硬件拓扑的节点，主要描述了物理系统组成部分的分布、交付、和安装
- 系统物理部署环境的映射关系



4+1视图

- **用例视 (Use Case View)**

- 系统的用例视通过用例描述了最终用户、分析人员和测试人员可以看到的系统行为
- **描述重要的系统用例**

