概述—UML统一建模语言回顾

UML 定义

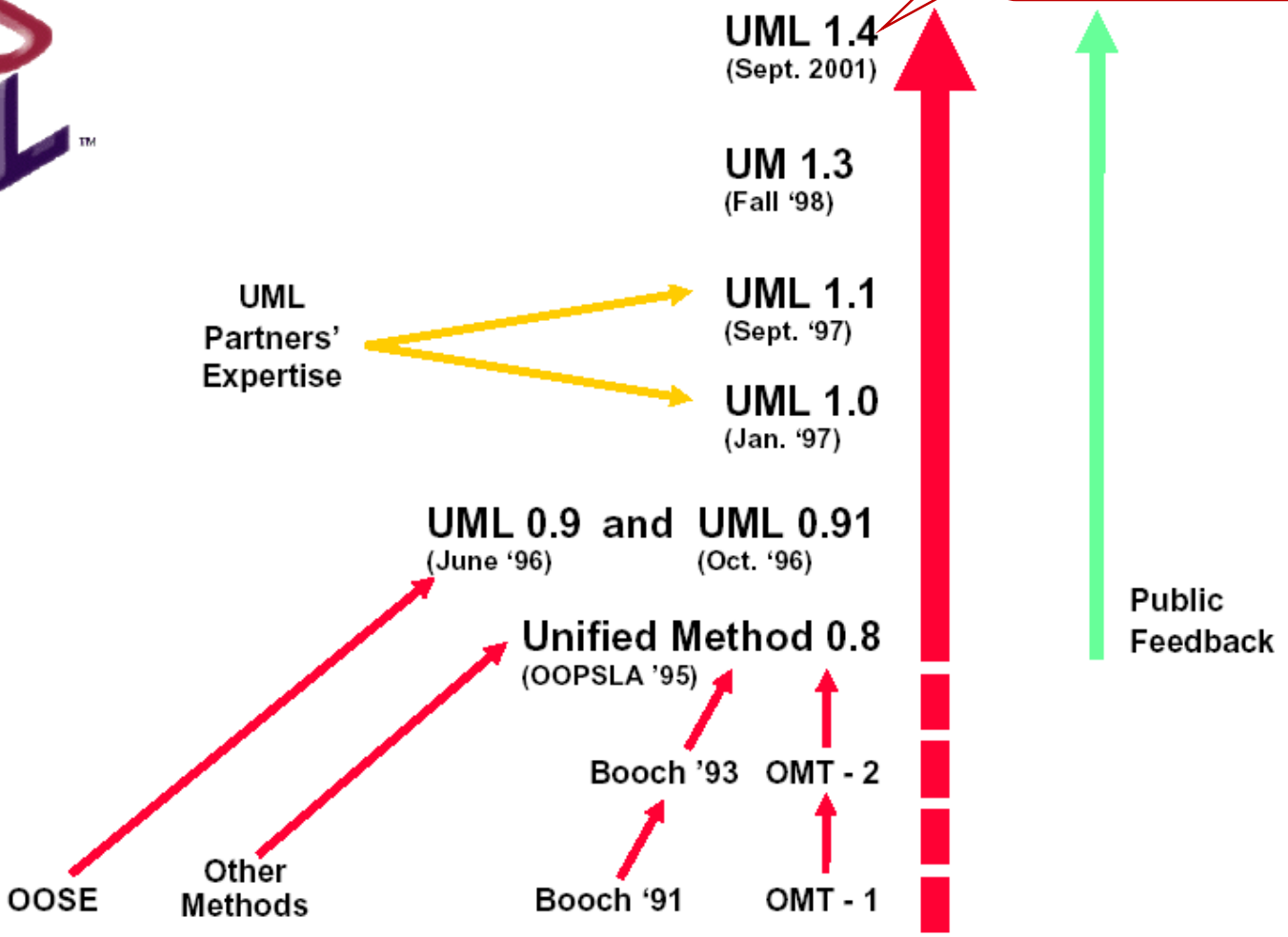
- **UML — Unified Modeling Language**
- **UML 是一种对软件系统的制作过程/产出物进行下述工作的描述语言：可视化（visualizing）、详述 （specifying）、构造 （constructing）、文档化（documenting）**

- UML 是可视化语言
  - UML 是图形化语言，便于交流
- UML 是一种可以详细描述的语言
  - 所建的模型是精确的，无歧义和完整的
- UML 是用于构造系统或理解系统的语言
  - UML 既支持正向工程，又支持反向工程
- UML 是文档化语言
  - 将所构造的系统记录下来
  - 便于后续人员跟进

概述—UML统一建模语言回顾

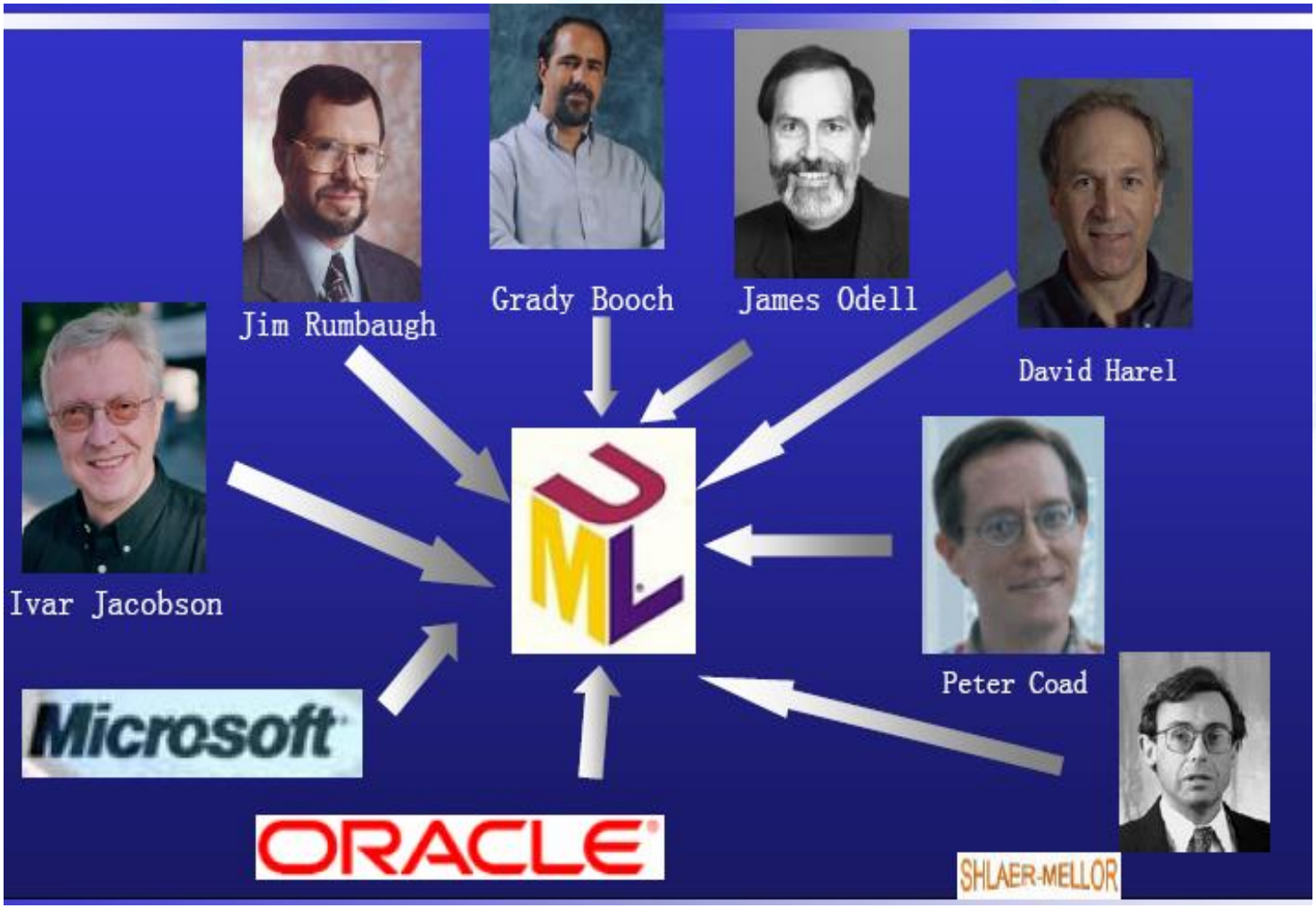
UML 的发展历史

目前UML的最新版本：V2.5.1
2017.12发布



概述—UML统一建模语言回顾

为 UML 创建做出贡献的人们



概述—UML统一建模语言回顾

UML 工具

概述—UML统一建模语言回顾

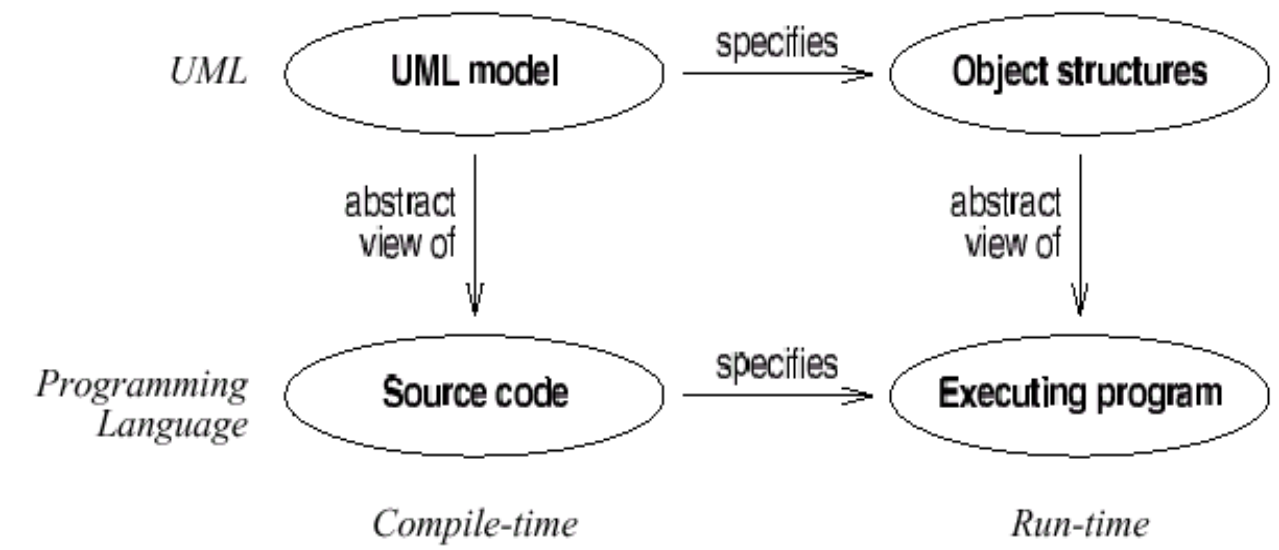
UML 的特点

- 统一标准：已成为面向对象的标准化的统一的建模语言
  - 面向对象
  - 可视化、表示能力强大
  - 独立于过程
  - 概念明确，建模表示法简洁，图形结构清晰，容易掌握使用

概述—UML统一建模语言回顾

UML 和代码的关系

- 用 Java，C++ 等 programming language 是用编码实现一个系统
- 用 UML 是对一个系统建立模型
- 一些工具（如Rational Rose）可以根据 UML 所建立的系统模型来产生 Java，C++ 或其它程序语言代码框架



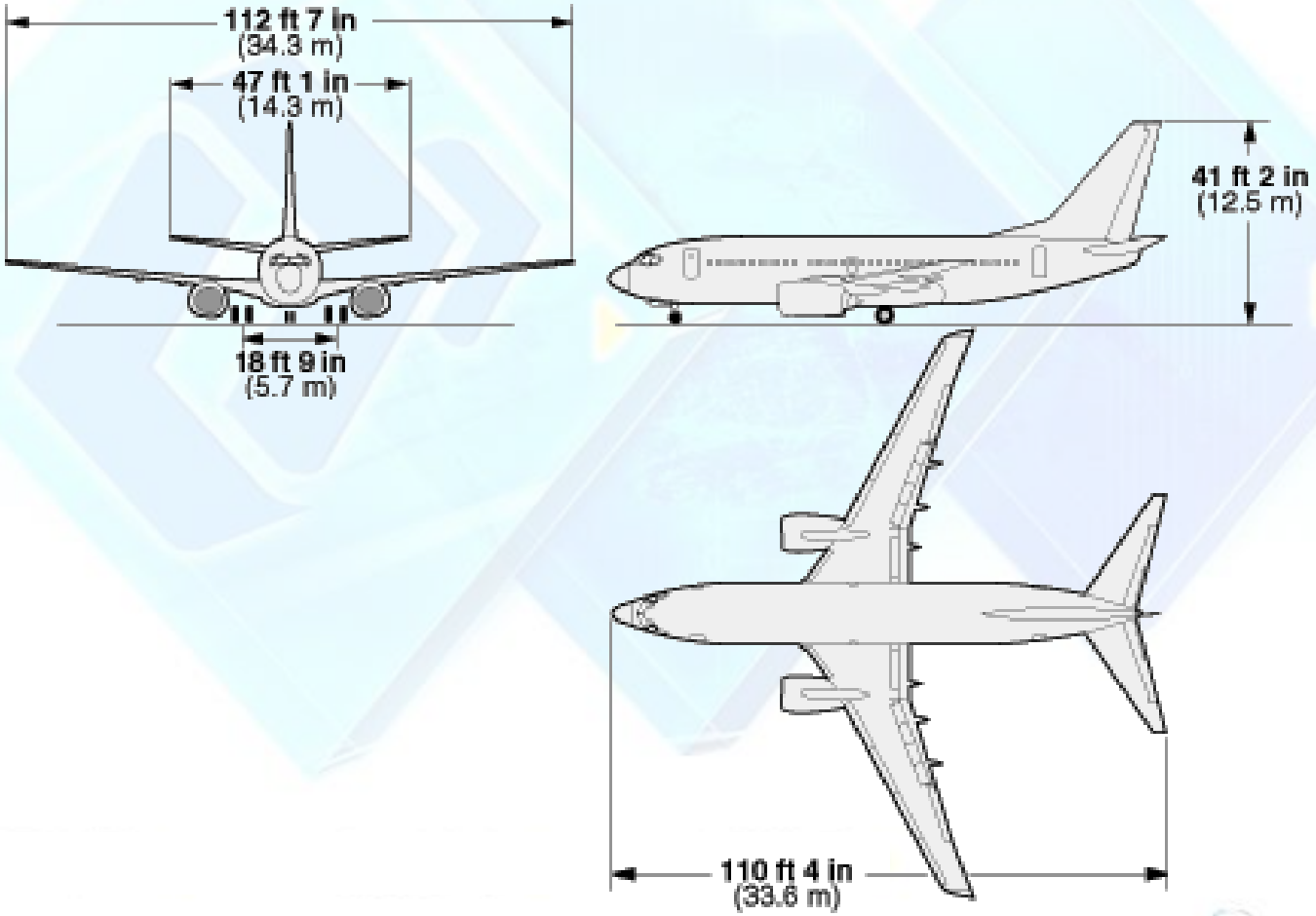
概述—UML统一建模语言回顾

UML 的构成

- 视图（Views）
- 图（Diagrams）
- 模型元素
- *通用机制

视图（Views）

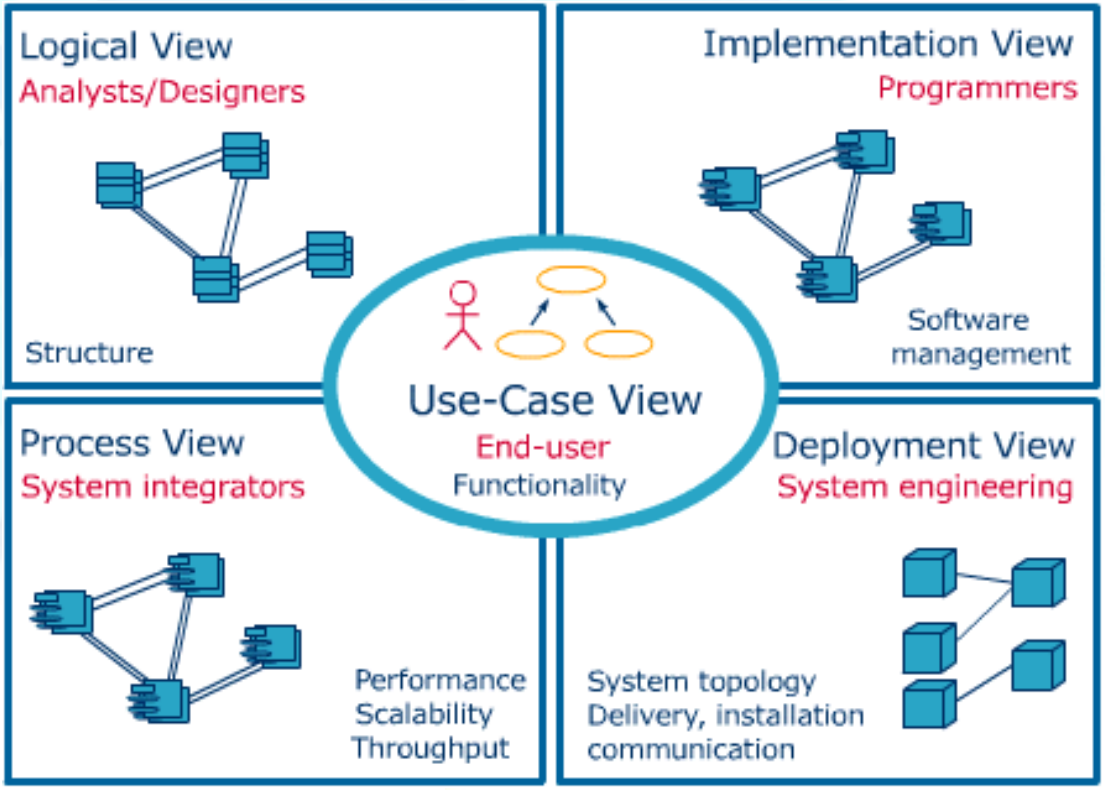
- 飞机的三视图



概述—UML统一建模语言回顾

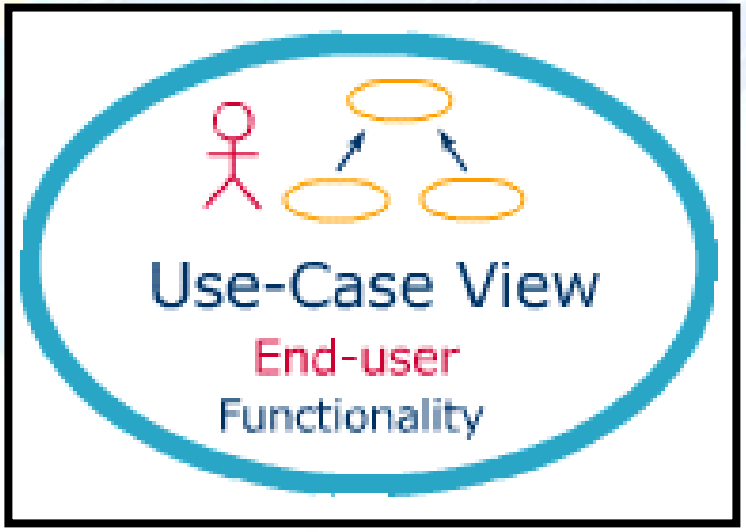
视图（Views）

- 视图是表达系统某一方面特征的 UML 建模元素的子集，它是由一个或者多个图组成的对系统某个角度的抽象
- 视图包括：
  - **Use Case View**
  - **Logical View**
  - **Process View**
  - **Implementation View**
  - **Deployment View**



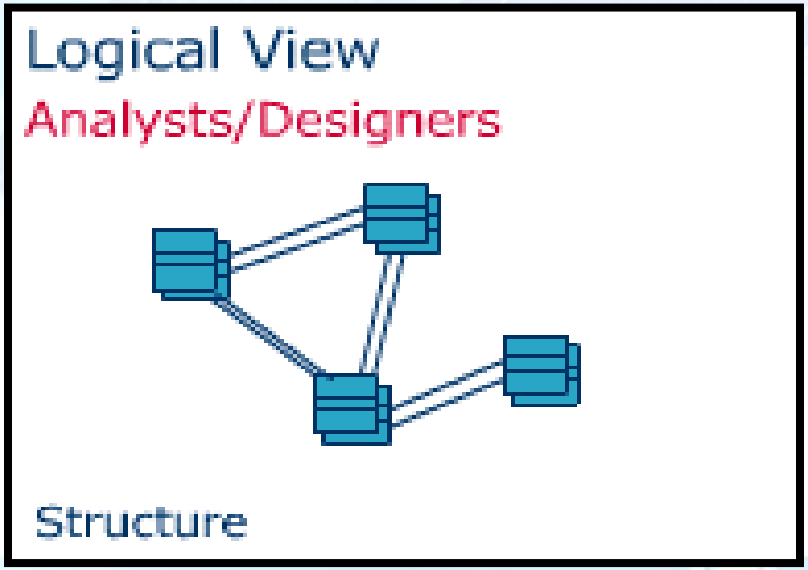
Use-Case View（用例视图）

- 用途：描述系统应该具备的功能，即被称为参与者（执行者）的外部用户所能观察到的功能
- 用例视图是几个视图的核心，它的内容直接驱动其他视图的开发
- 包含UML图：用例图
- 使用者
  - 分析人员



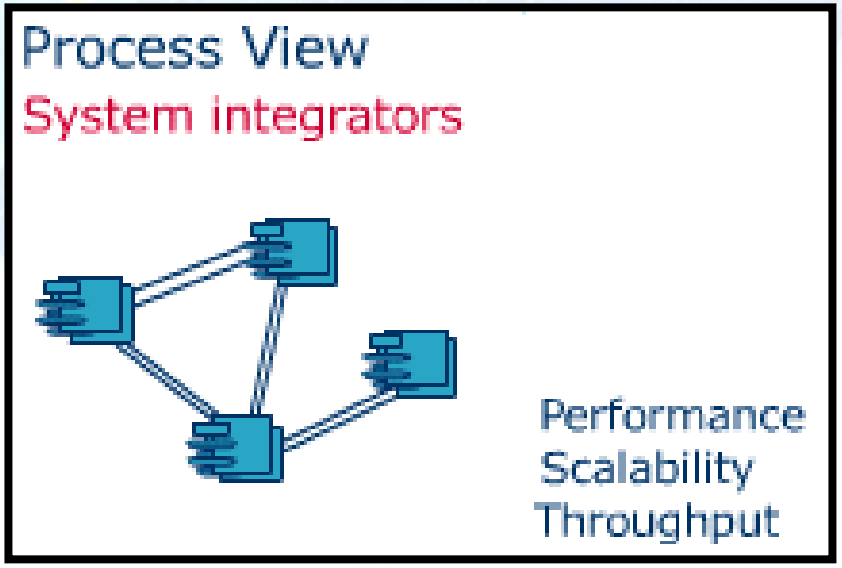
Logical View（逻辑视图）

- 用途：描述用例视图中提出的系统功能的实现
- 逻辑视图既描述系统的静态结构，也描述系统内部的动态协作关系
- 包含UML图：类图、对象图、状态图、时序图、协作图、活动图
- 静态结构在类图和对象图中描述
- 动态模型在状态图、时序图、协作图、活动图中描述
- 使用者
  - 分析人员
  - 设计人员
  - 开发人员



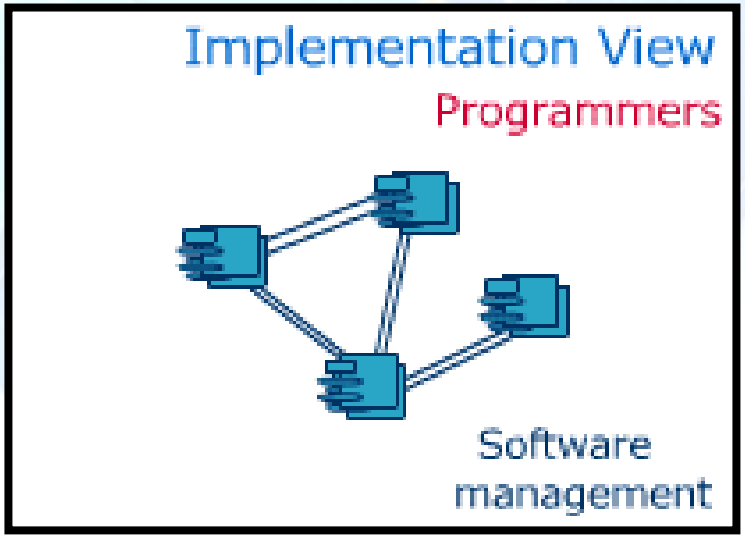
Process View（进程视图）

- 用途：考虑资源的有效利用、代码的并行执行以及系统环境中异步事件的处理
- 解决在并发系统中存在的通信和同步问题
- 包含UML图：状态图、协作图、组件图、活动图
- 使用者
  - 开发人员
  - 系统集成人员



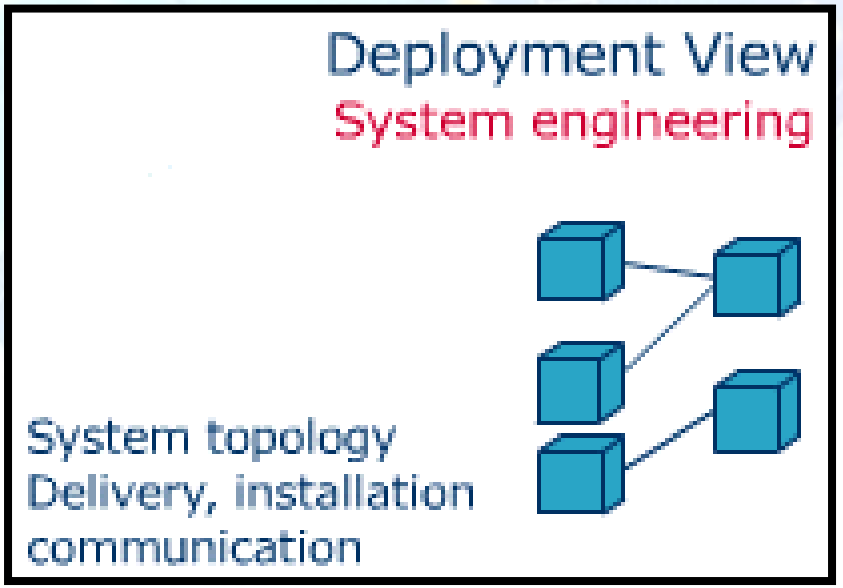
Implementation View（实现视图）

- 用途：描述系统的实现模块以及它们之间的依赖关系
- 包含UML图：组件图
- 使用者
  - 开发人员



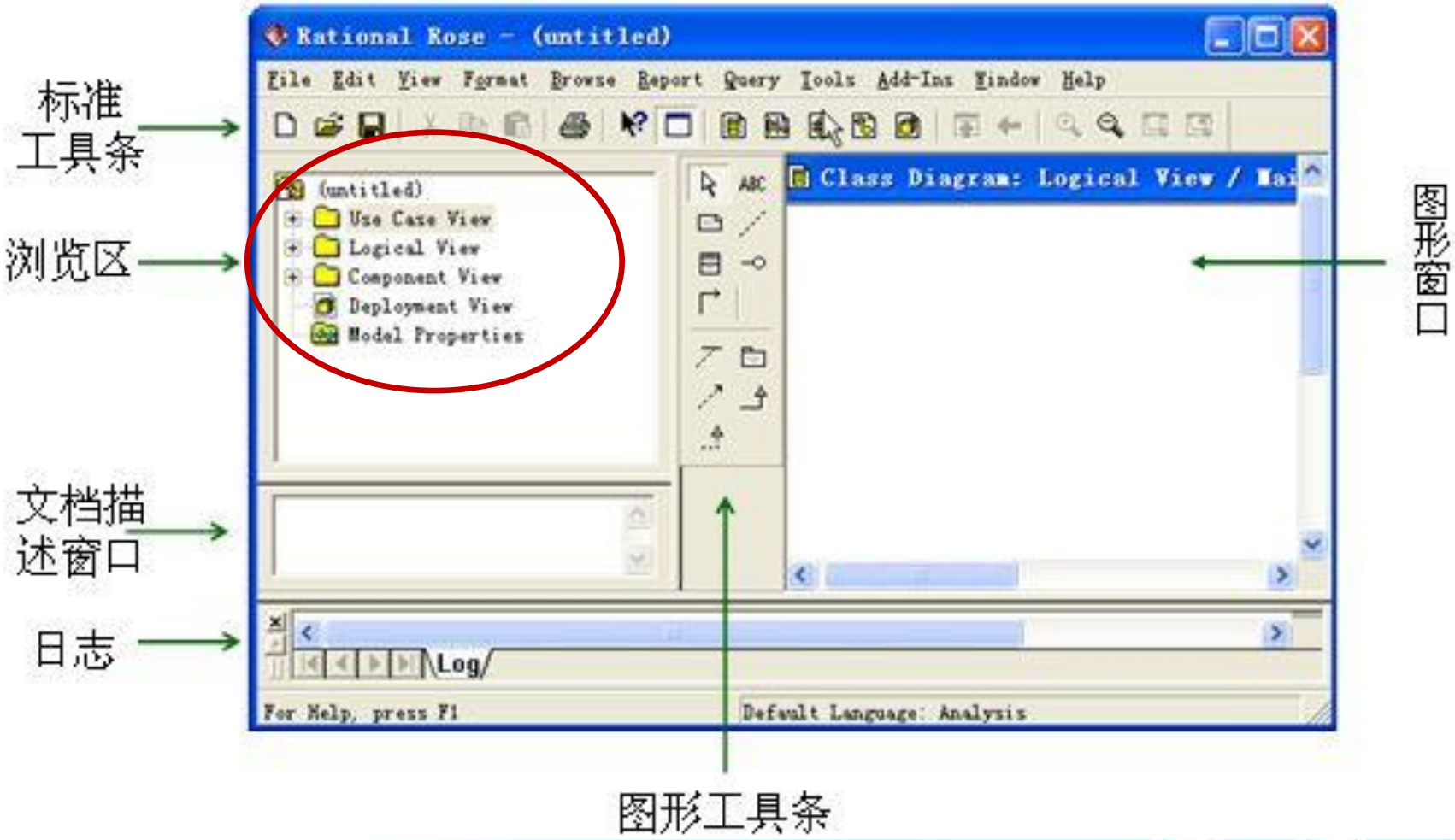
Deployment View（部署视图）

- 用途：显示系统的物理部署，并描述位于节点实例上的运行组件实例的部署情况
- 组成：部署图
- 使用者
  - 开发人员
  - 系统集成人员
  - 测试人员



典型UML建模工具：Rational Rose

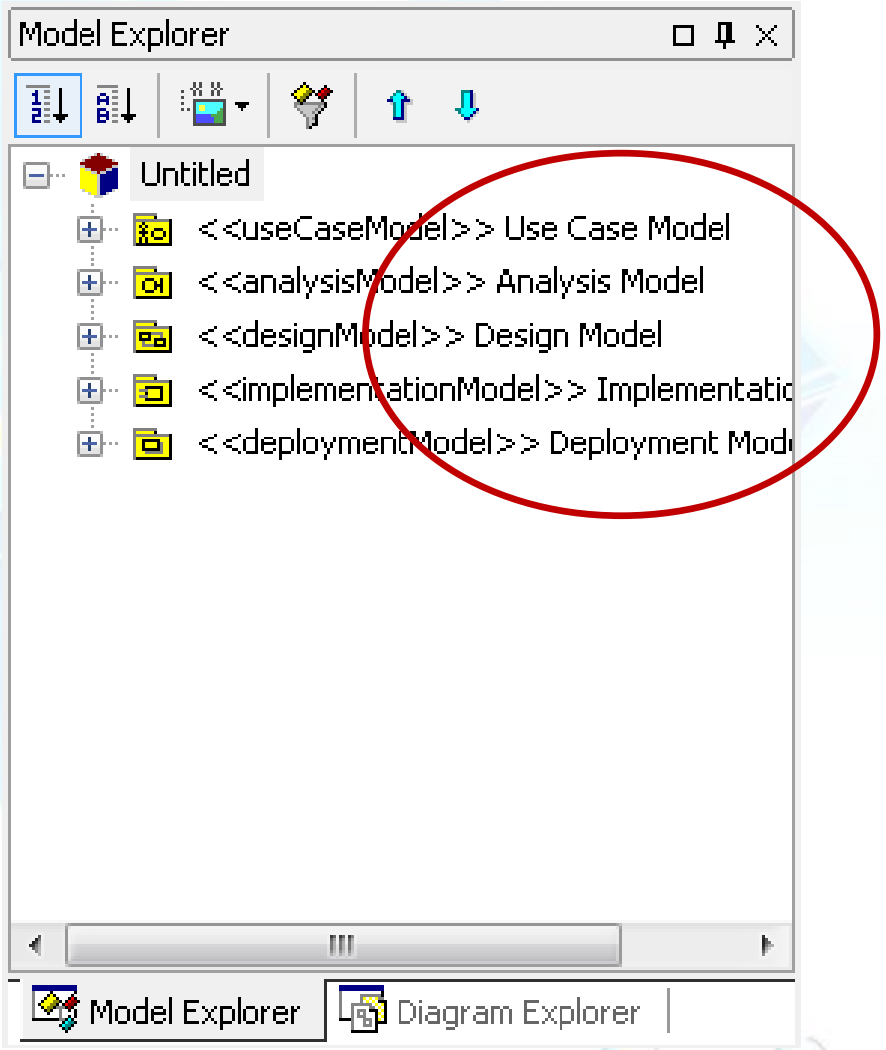
- **4个 View**



概述—UML统一建模语言回顾

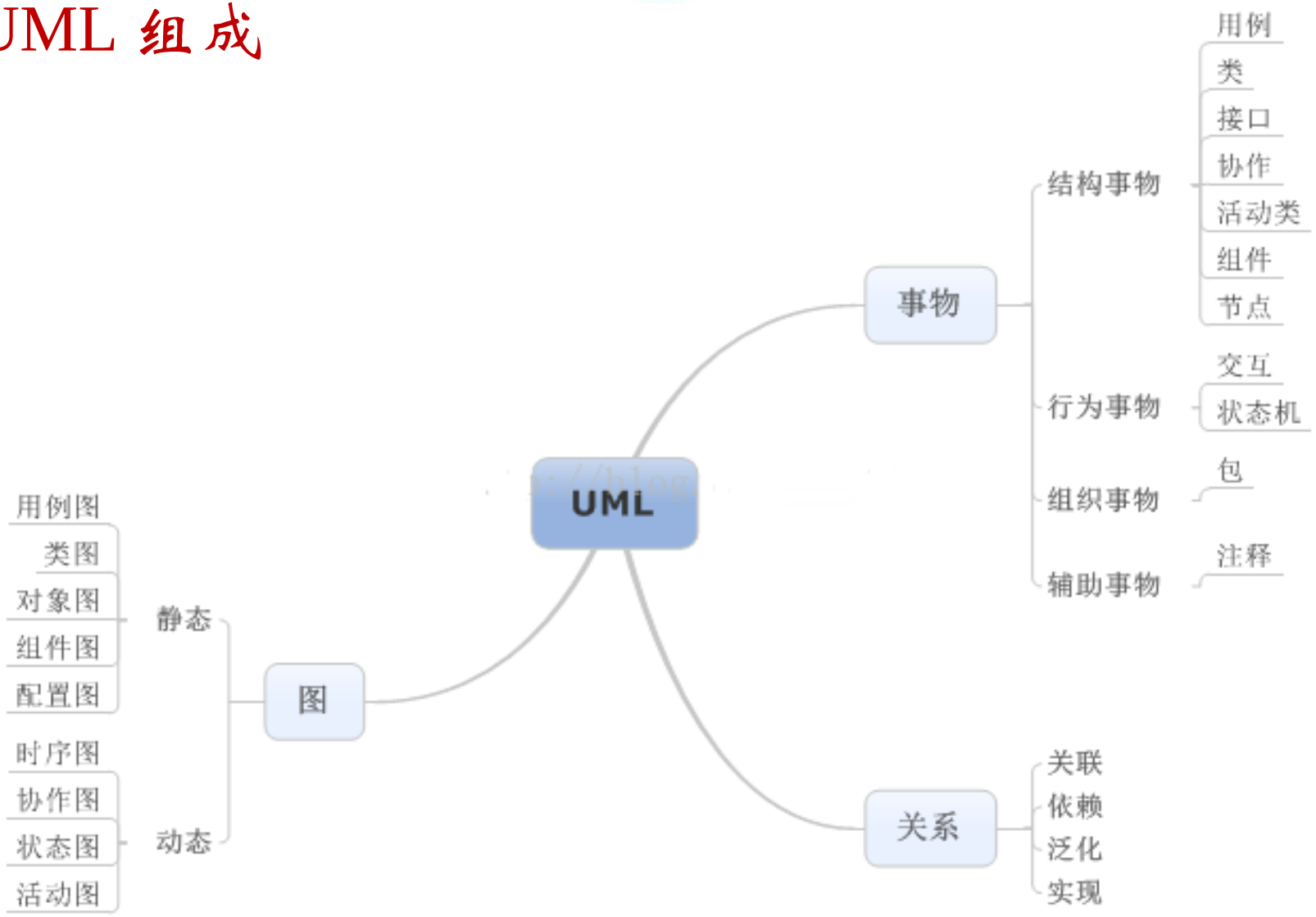
典型UML建模工具： StarUML

- 5个 Model



概述—UML统一建模语言回顾

UML 组成



概述—UML统一建模语言回顾

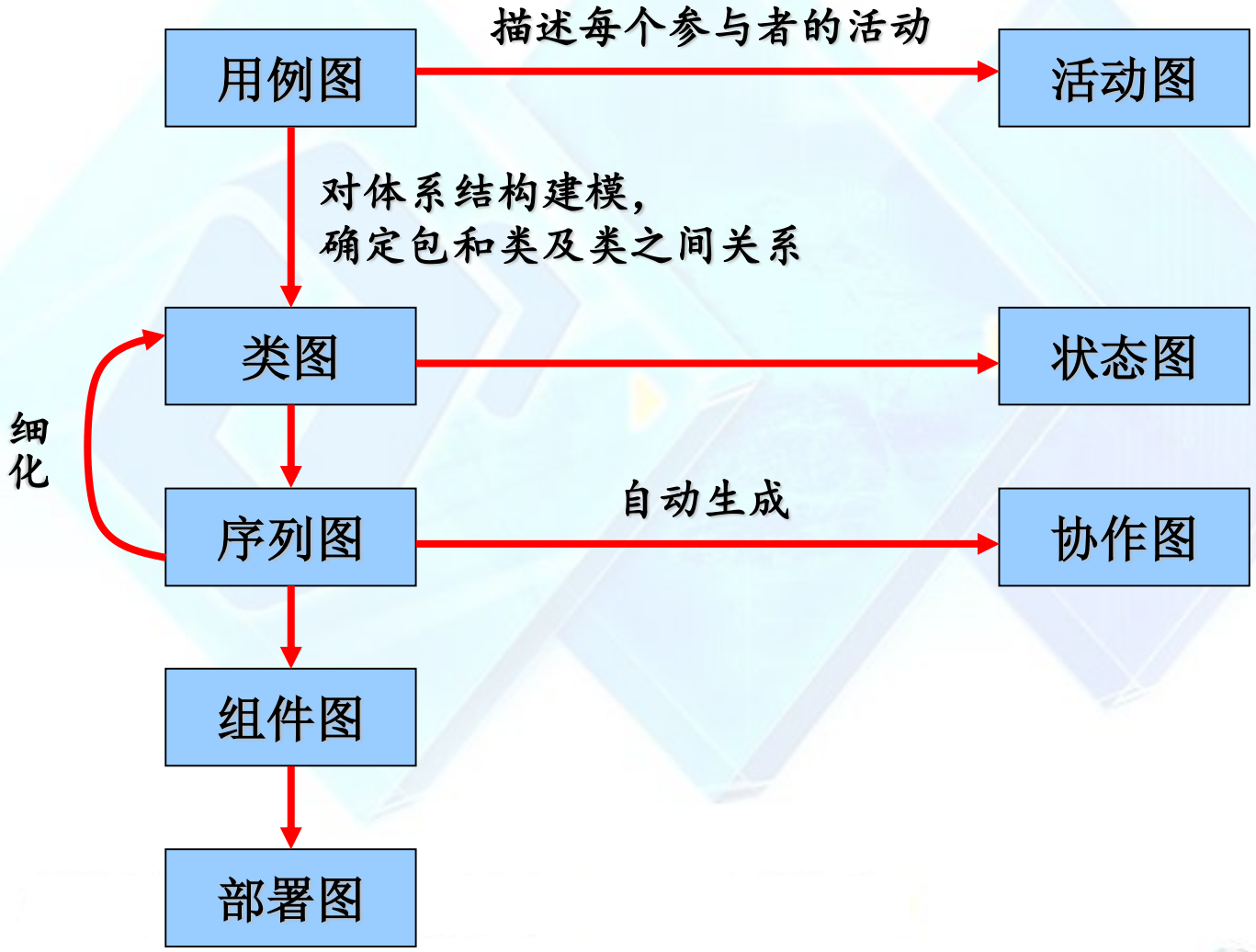
UML中的模型图（9种）

- 类图 (class diagram)
- 对象图 (object diagram)
- 用例图 (use case diagram)
- 时序图 (sequence diagram)
- 协作图 (collaboration diagram)
- 状态图 (statechart diagram)
- 活动图 (activity diagram)
- 组件图 (component diagram)
- 部署图 (deployment diagram)

* 包图（结构化建模用）

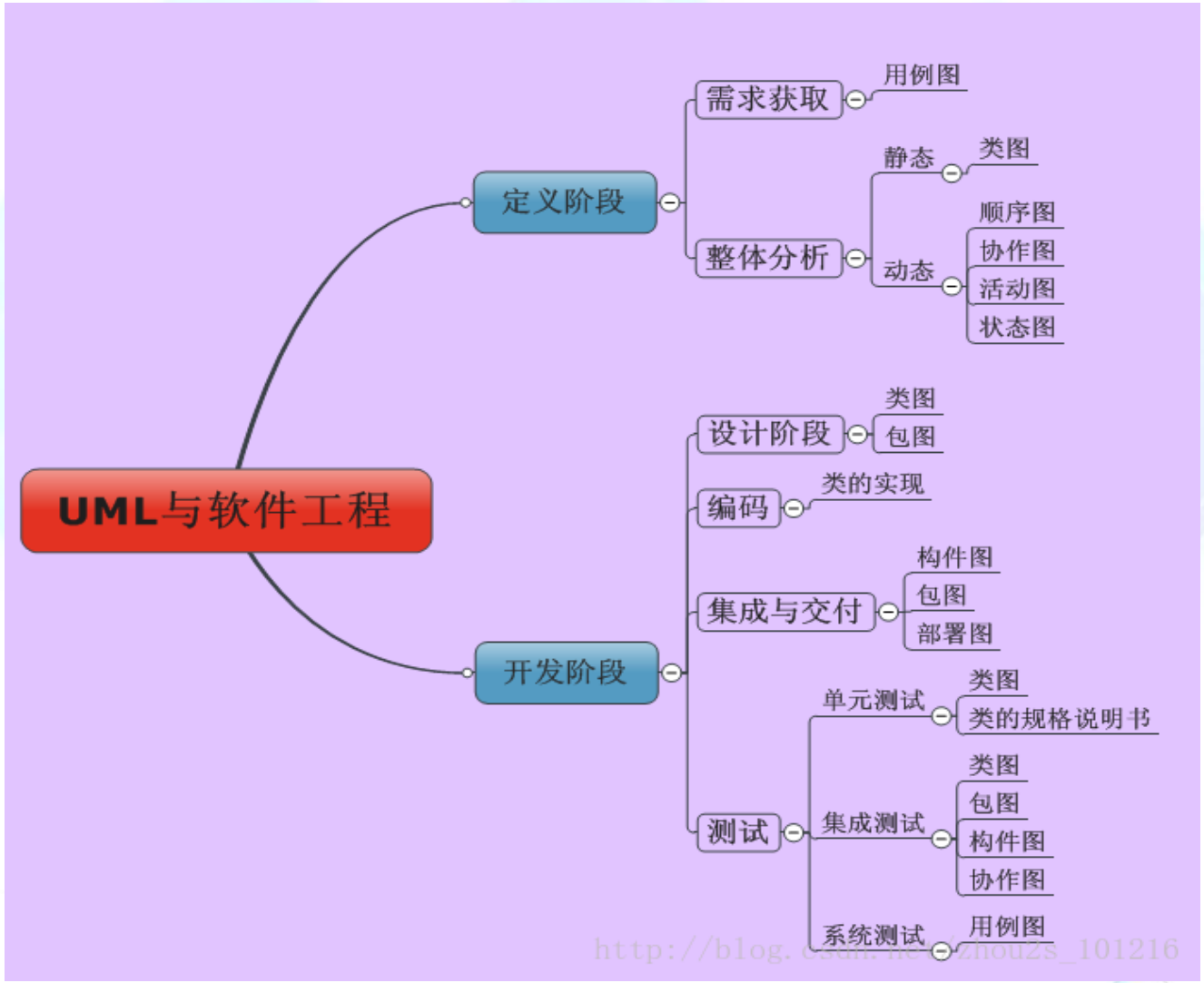
概述—UML统一建模语言回顾

UML图之间的关系



概述—UML统一建模语言回顾

UML图与软件工程的关系



概述—UML统一建模语言回顾

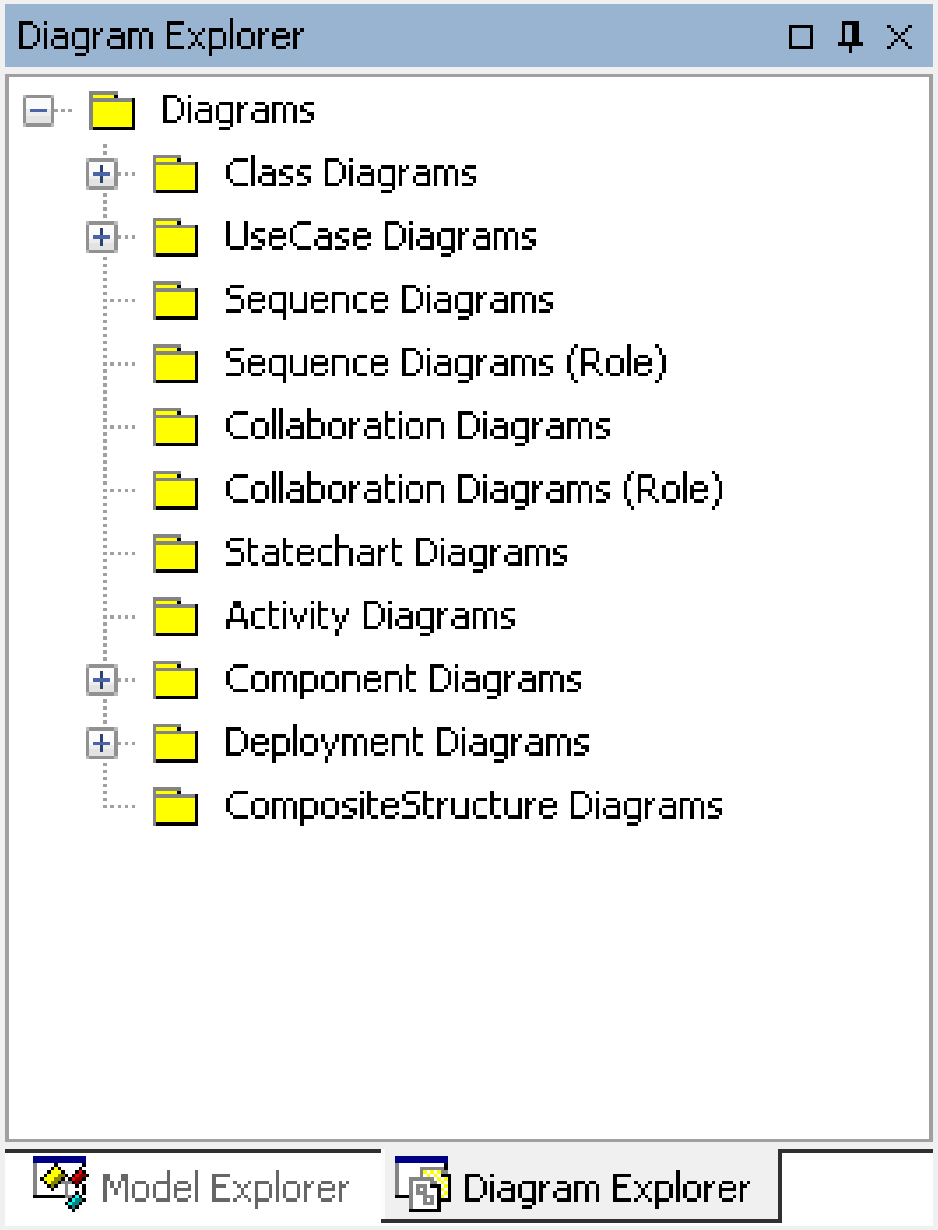
UML图与使用者之间的关系

UML图的使用人员					
人员 / 图	系统用户	分析人员	设计人员	开发人员	测试人员
用例图					
类图					
对象图					
序列图					
协作图					
状态图					
活动图					
构件图					
部署图					

概述—UML统一建模语言回顾

StarUML支持UML图

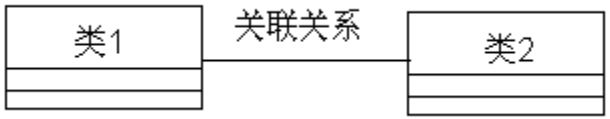
- **Diagram**

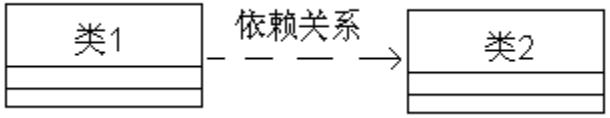


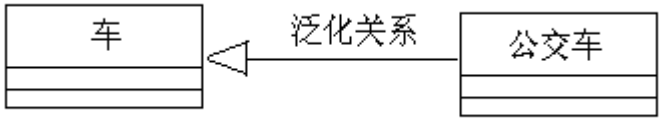
概述—UML统一建模语言回顾

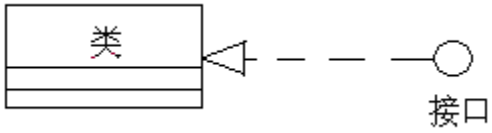
UML中的关系

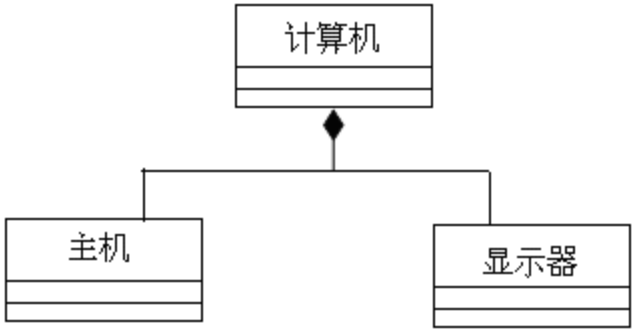
- 关联
- 依赖
- 泛化
- 实现
- 聚合/组合







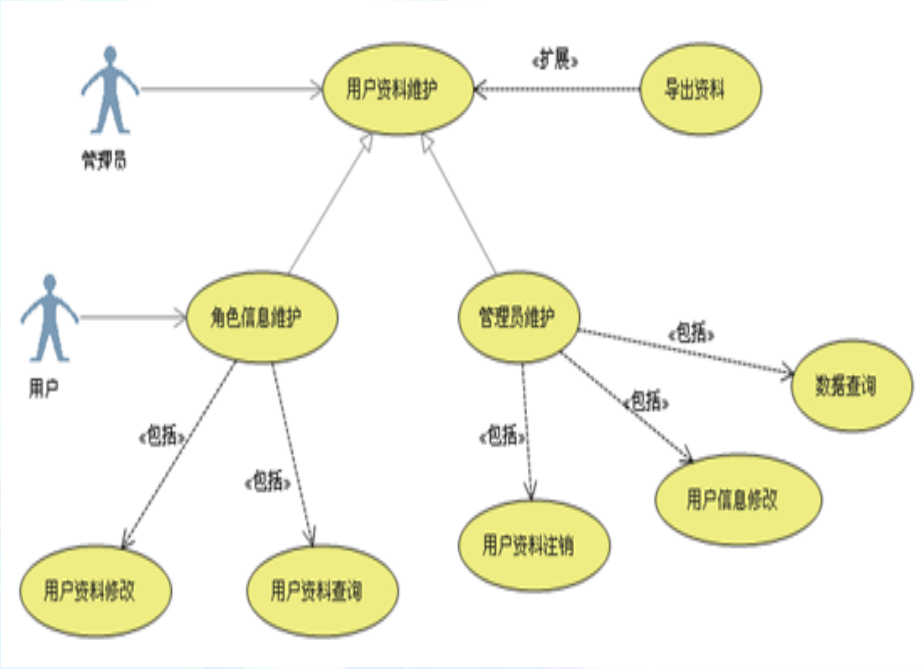


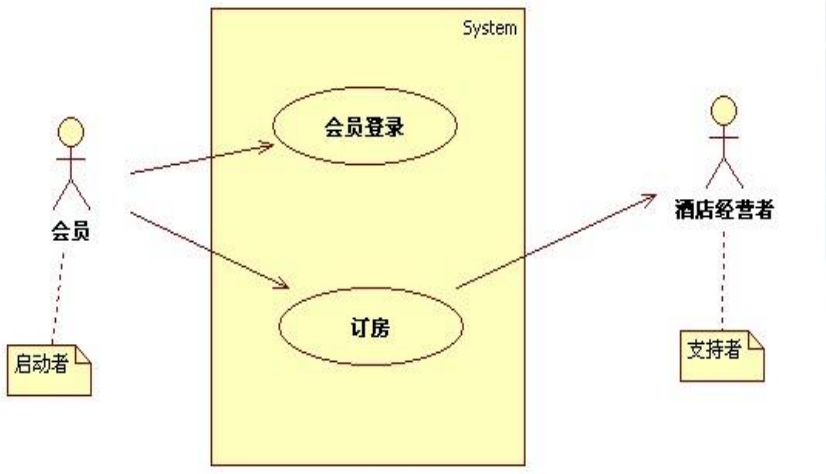


概述—UML统一建模语言回顾

用例图（UseCaseDiagram）

- 从用户的角度描述了系统的功能，并指出各个功能的执行者，强调用户的使用者，系统为执行者完成哪些功能

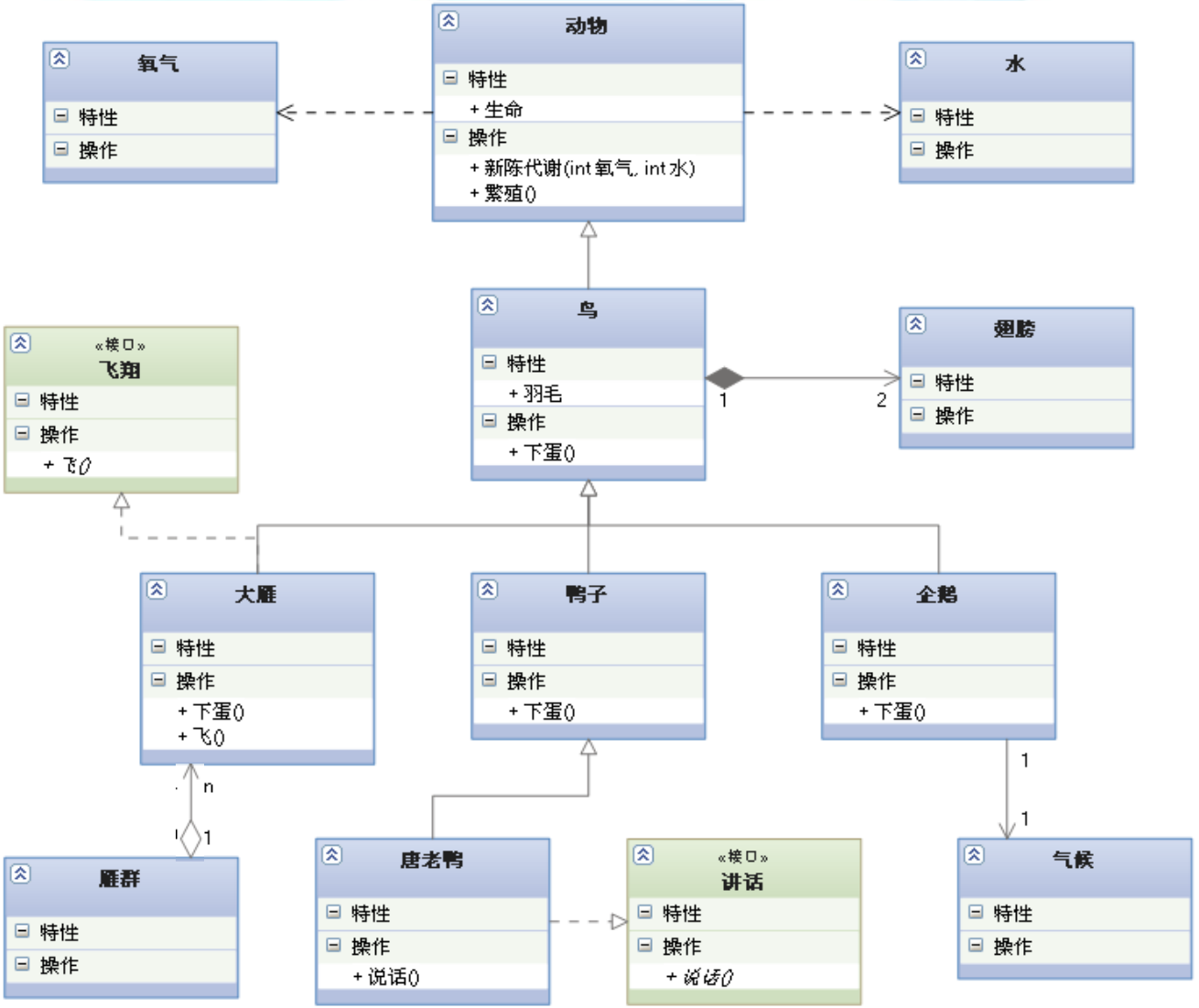




概述—UML统一建模语言回顾

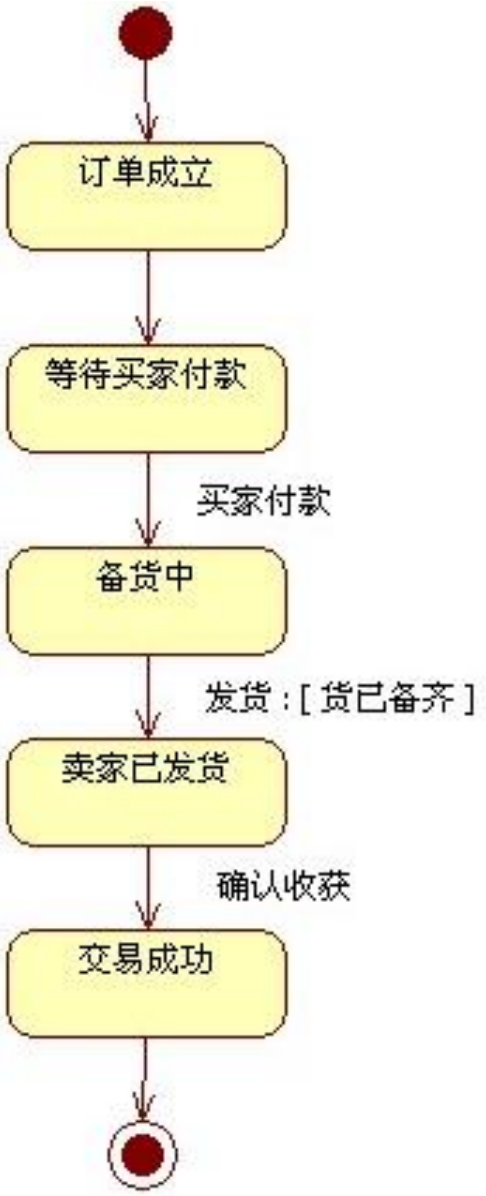
类图（ClassDiagram）

- 描述类的内部结构和类与类之间的关系，是一种静态结构图



状态图（StatechartDiagram）

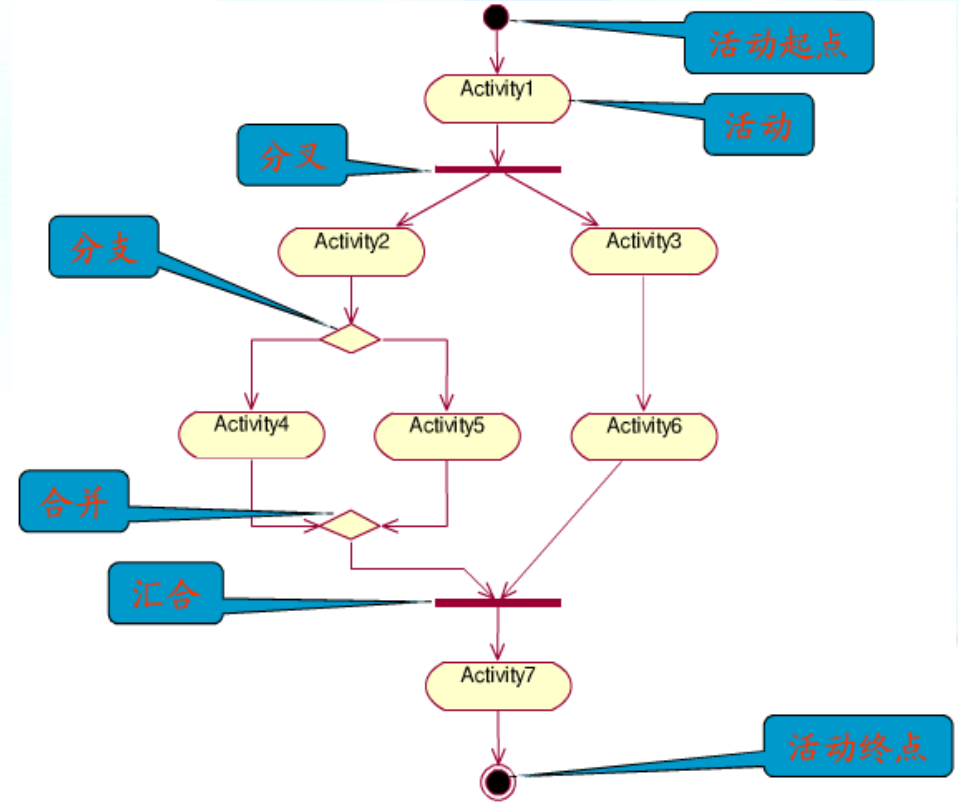
- 一种由状态、变迁、事件和活动组成的状态机，用来描述类的对象所有可能的状态以及时间发生时状态的转移条件

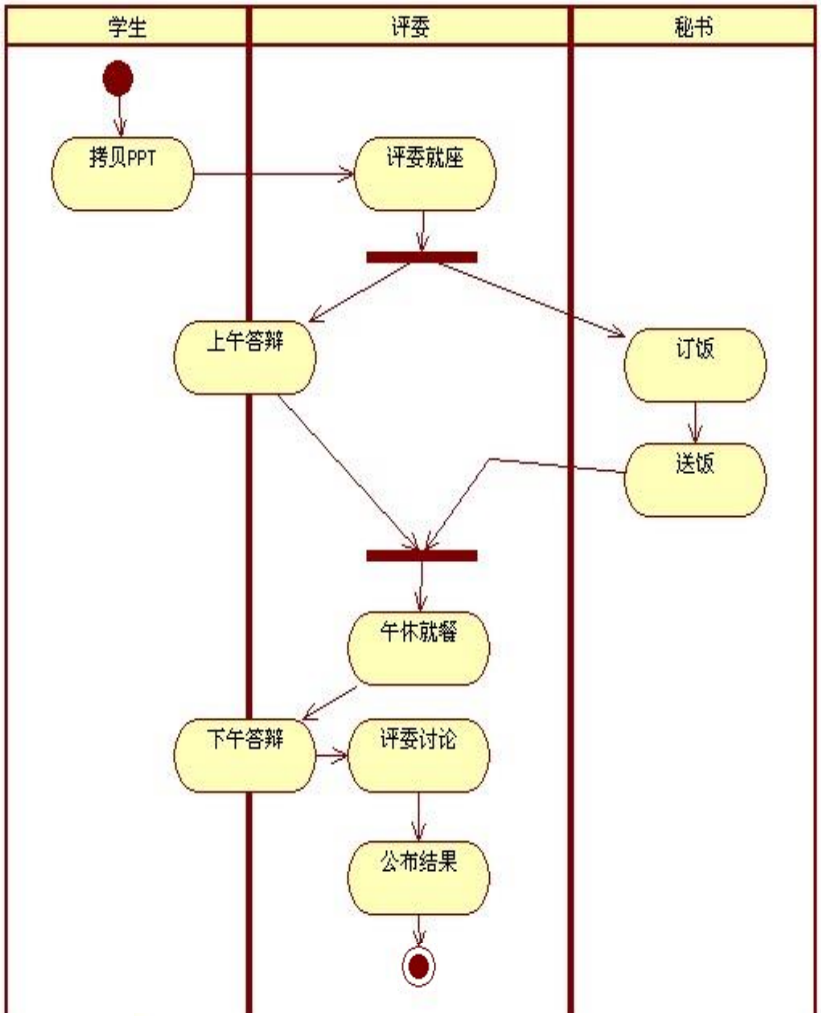


概述—UML统一建模语言回顾

活动图（ActivityDiagram）

- 是状态图的一种特殊情况，这些状态大都处于活动状态。本质是一种流程图，它描述了活动到活动的控制流

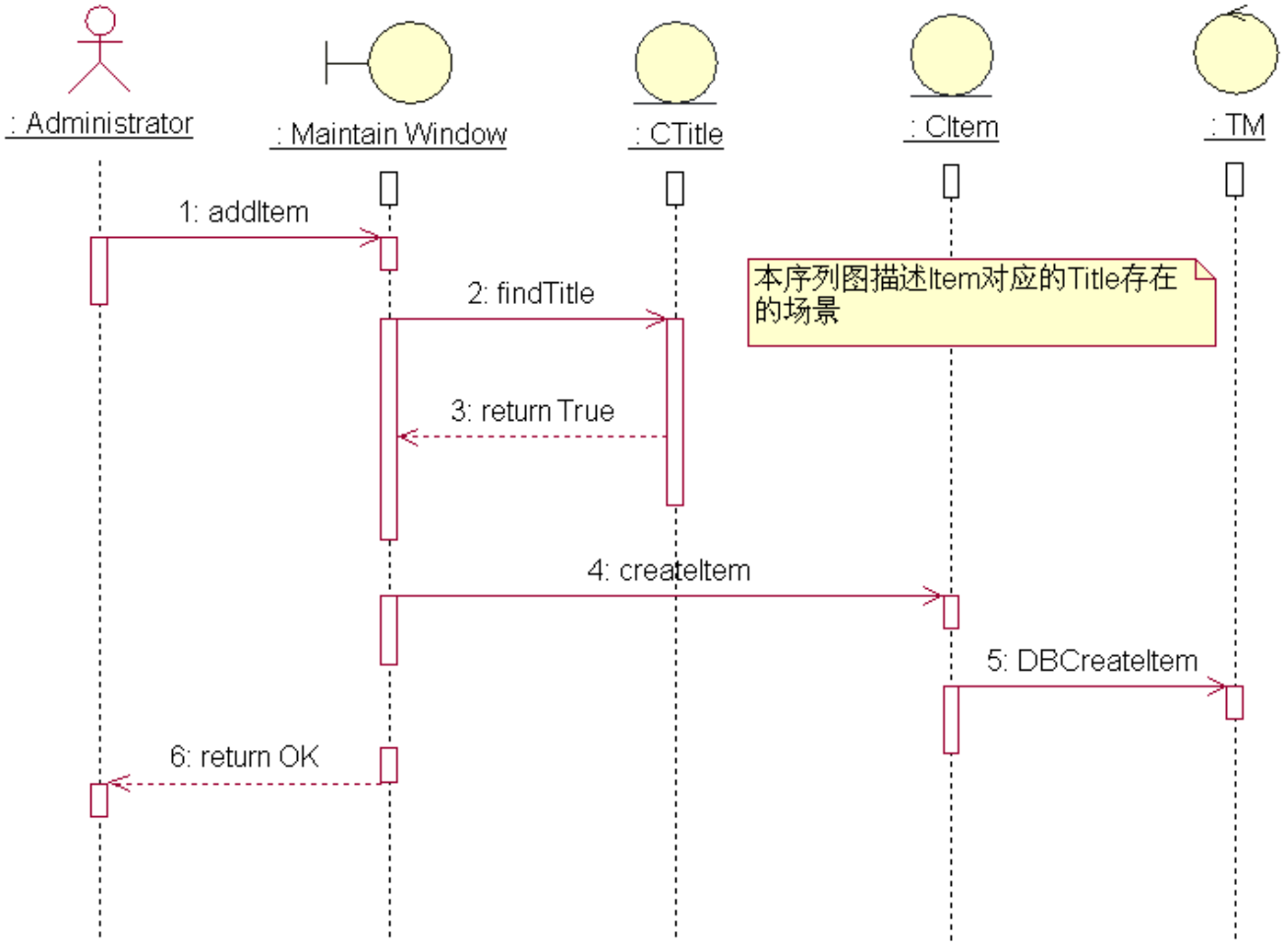




概述—UML统一建模语言回顾

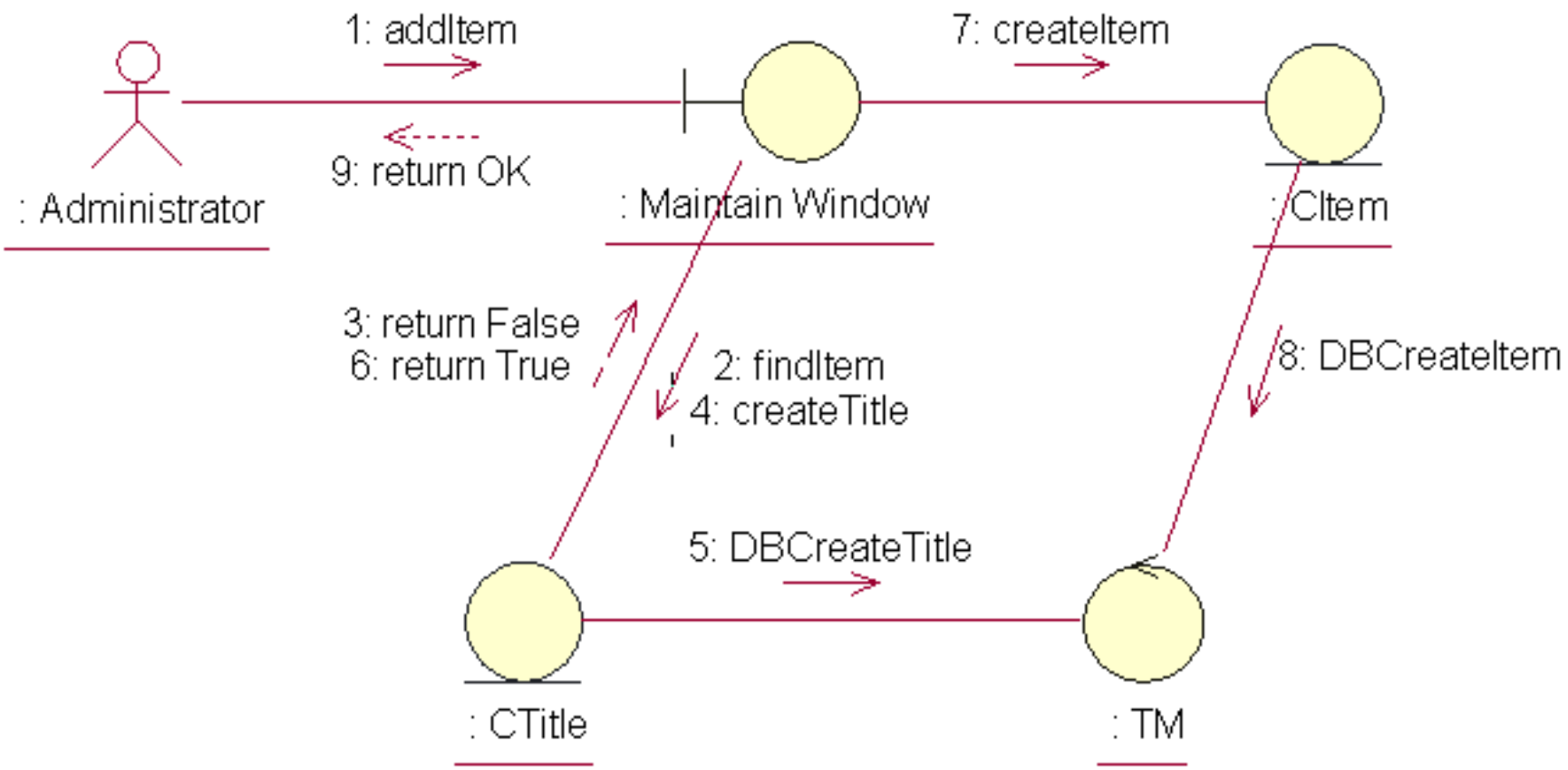
时序图（SequenceDiagram）

- 交互图的一种，描述了对象之间消息发送的先后顺序，强调时间顺序。
- 序列图的主要用途是把用例表达的需求，转化为进一步、更加正式层次的精细表达



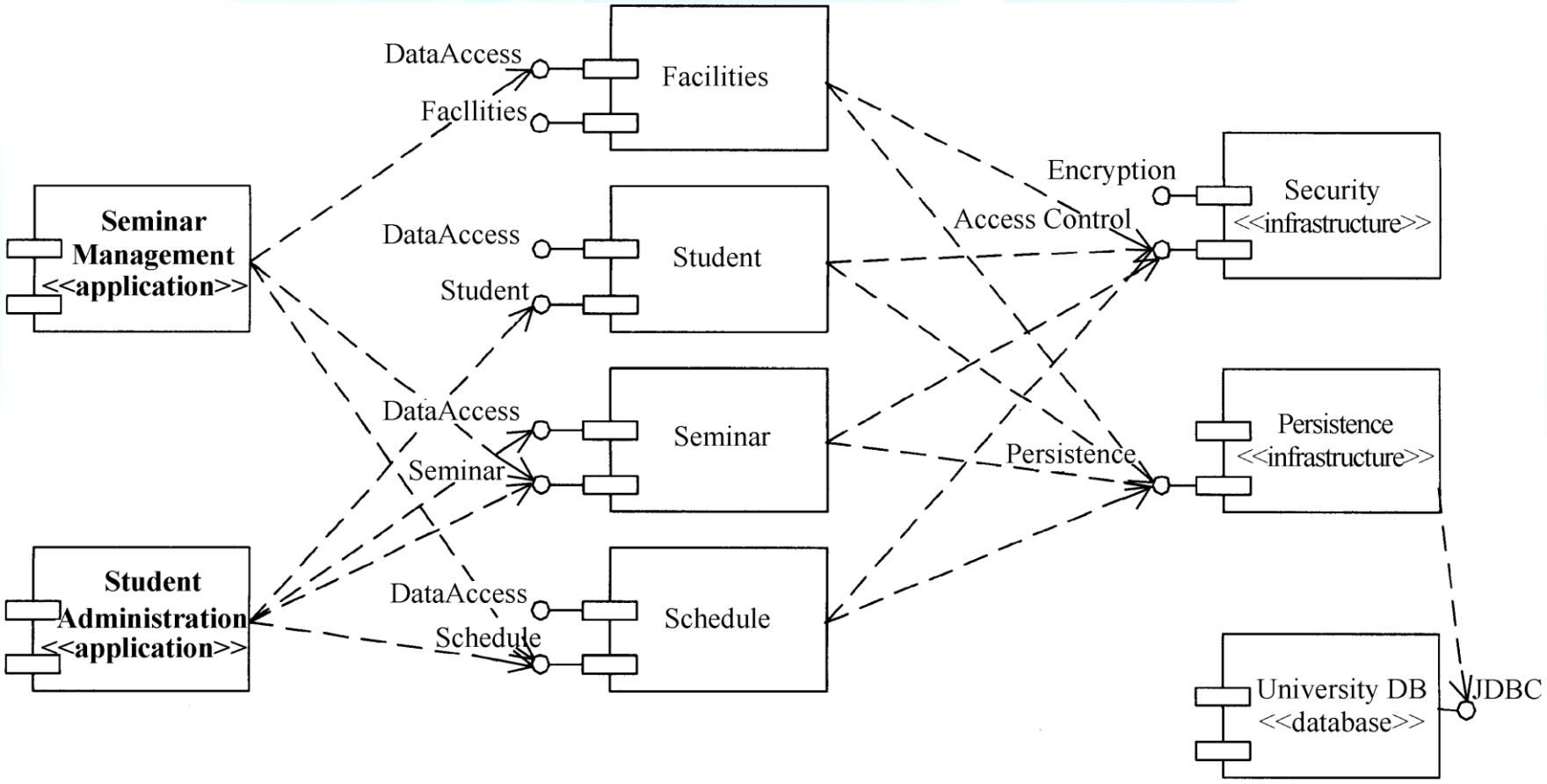
协作图（CollaborationDiagram）

- 交互图的一种，描述了收发消息的对象的组织关系，强调对象之间的合作关系



组件图（ComponentDiagram）

- 用来表示系统中组件与组件之间，类或接口与组件之间的关系图



概述—UML统一建模语言回顾

部署图（DeploymentDiagram）

- 描述了系统运行时进行处理的结点以及在结点上活动的构件的配置。强调了物理设备以及之间的连接关系