



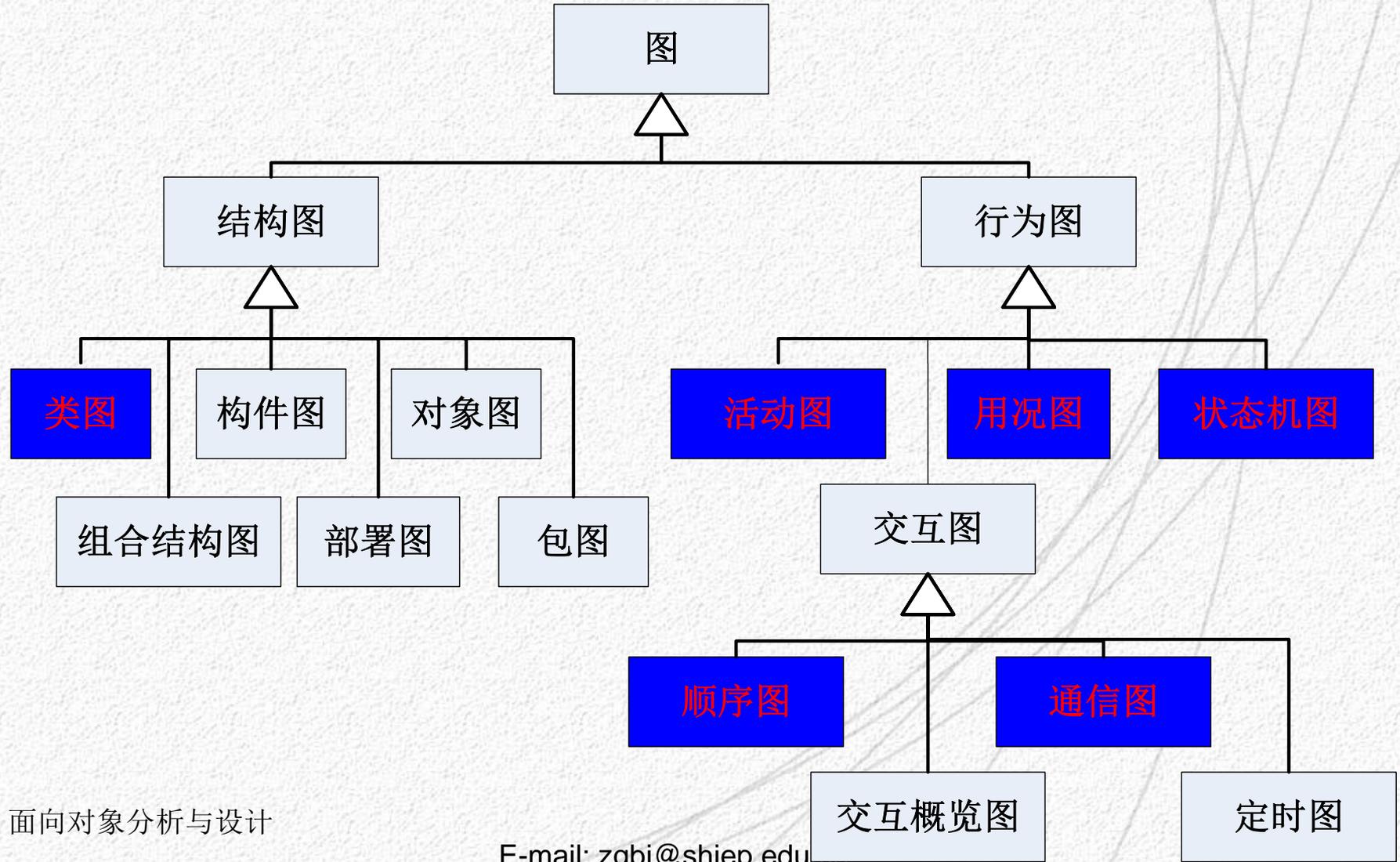
上海電力學院

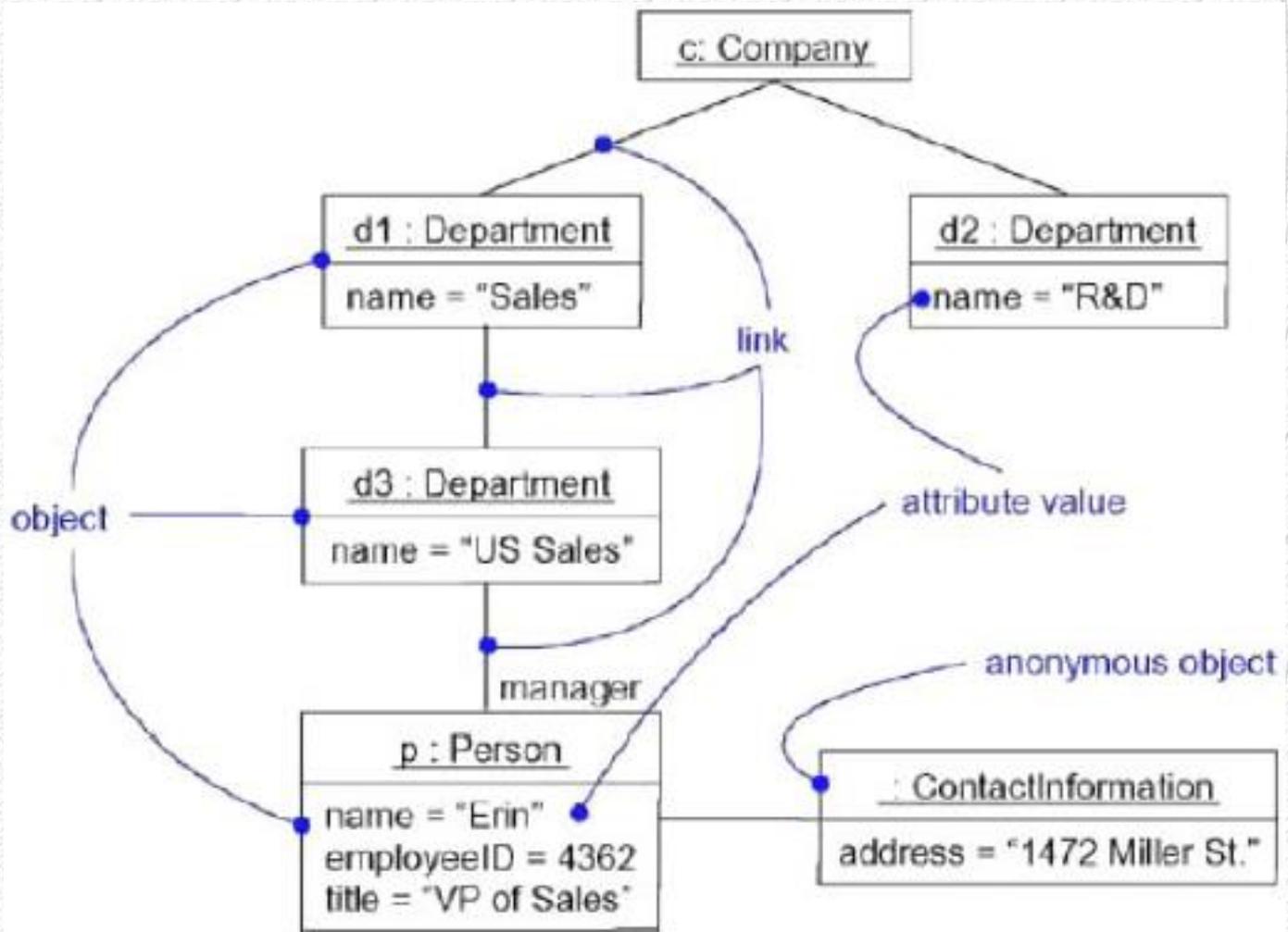
Shanghai University of Electric Power

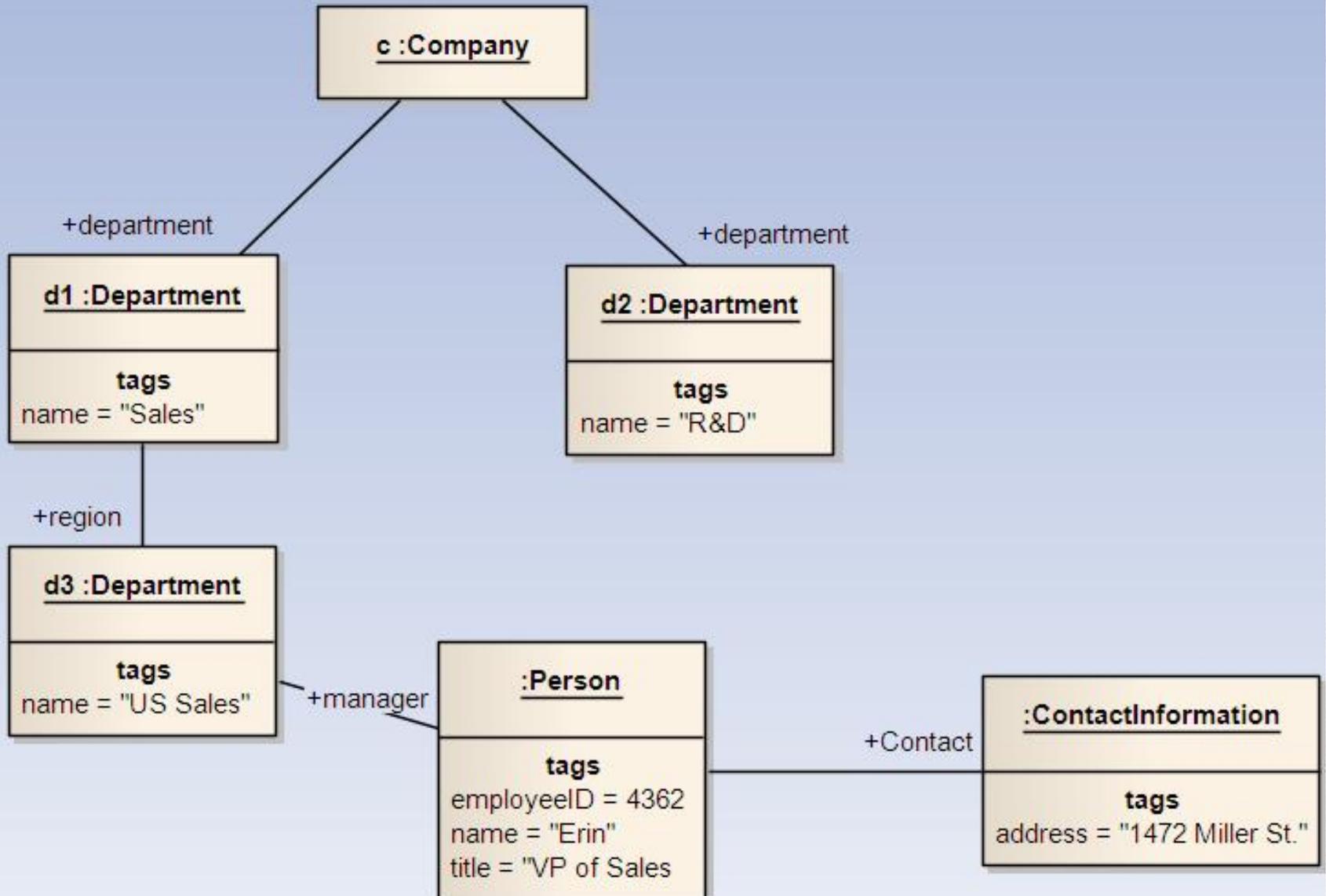
面向对象分析与设计

第七章 UML其他辅助图

www.shiep.edu.cn

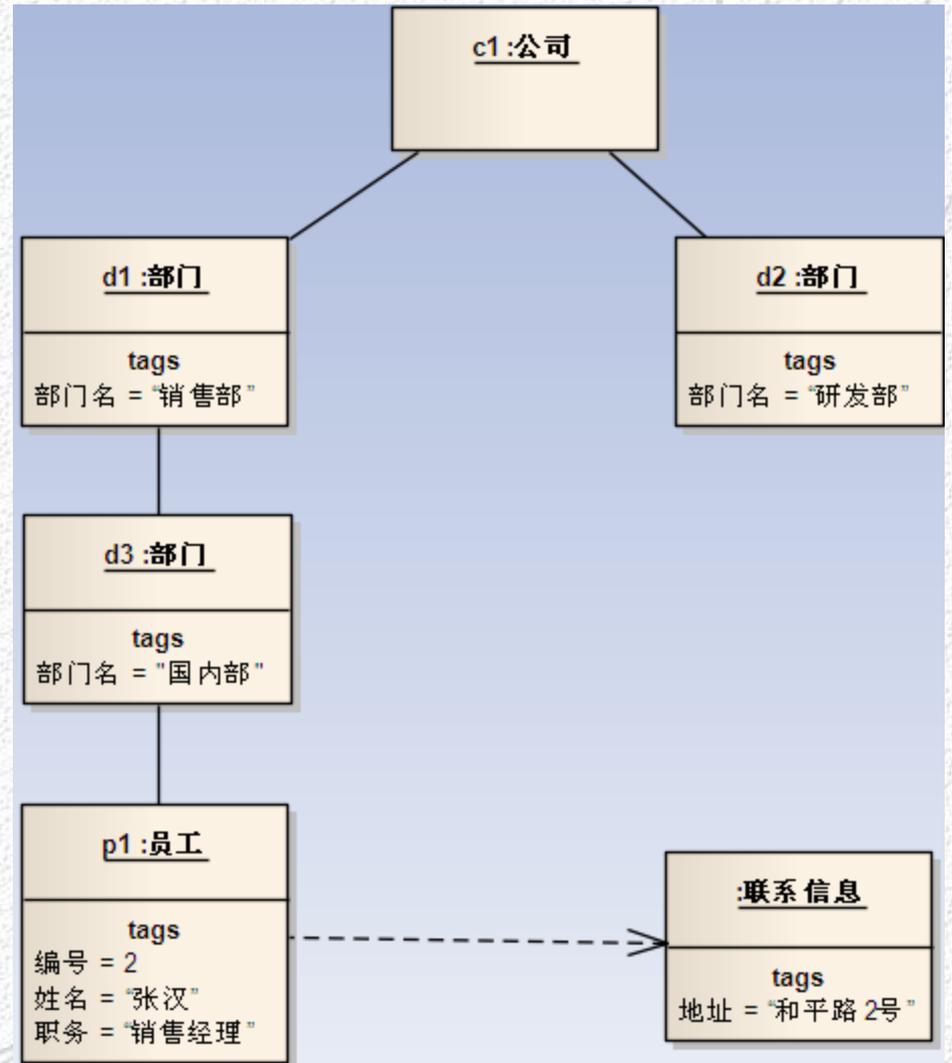
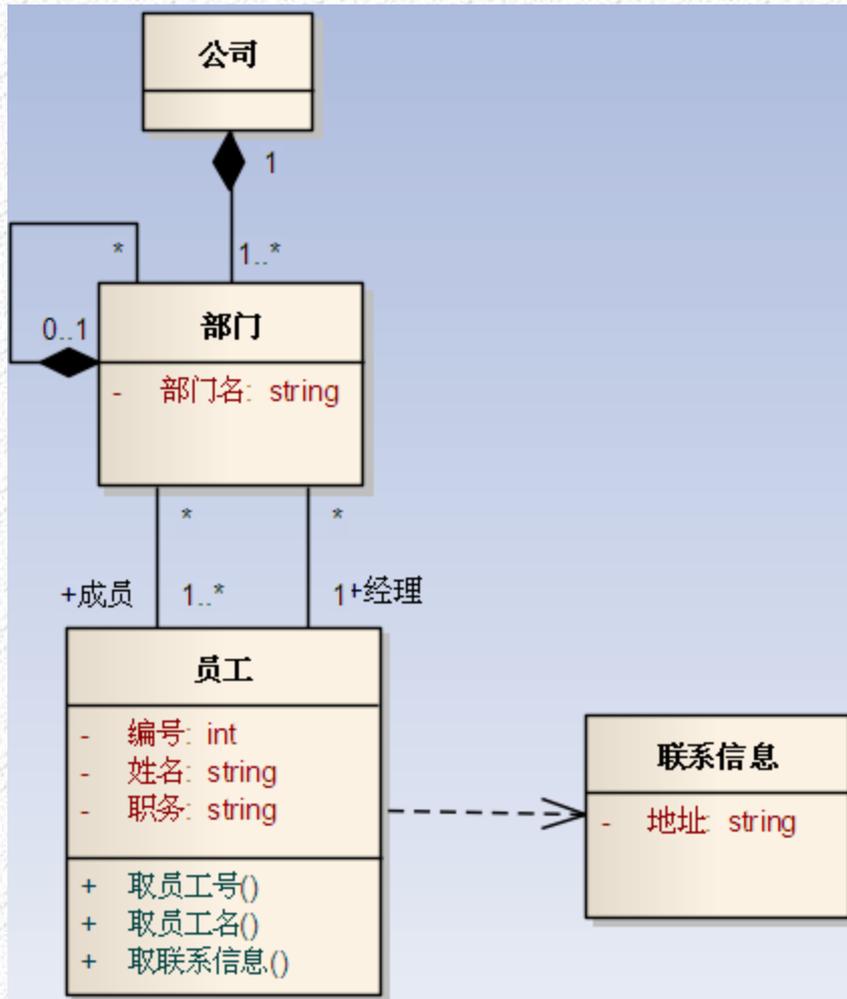






■ 对象图

- 对象图(object diagram)是表示在某一时间点上一组对象以及它们之间的关系的图。在图形上，对象图是顶点和弧的集合。
- 对象图一般包括：
 - 对象
 - 链
 - 也可以包含注解和约束
- 对象图是类图的一个实例





■ 用途

- 捕捉实例和连接
- 在分析和设计阶段创建
- 捕获交互的静态部分
- 详细描述瞬态图



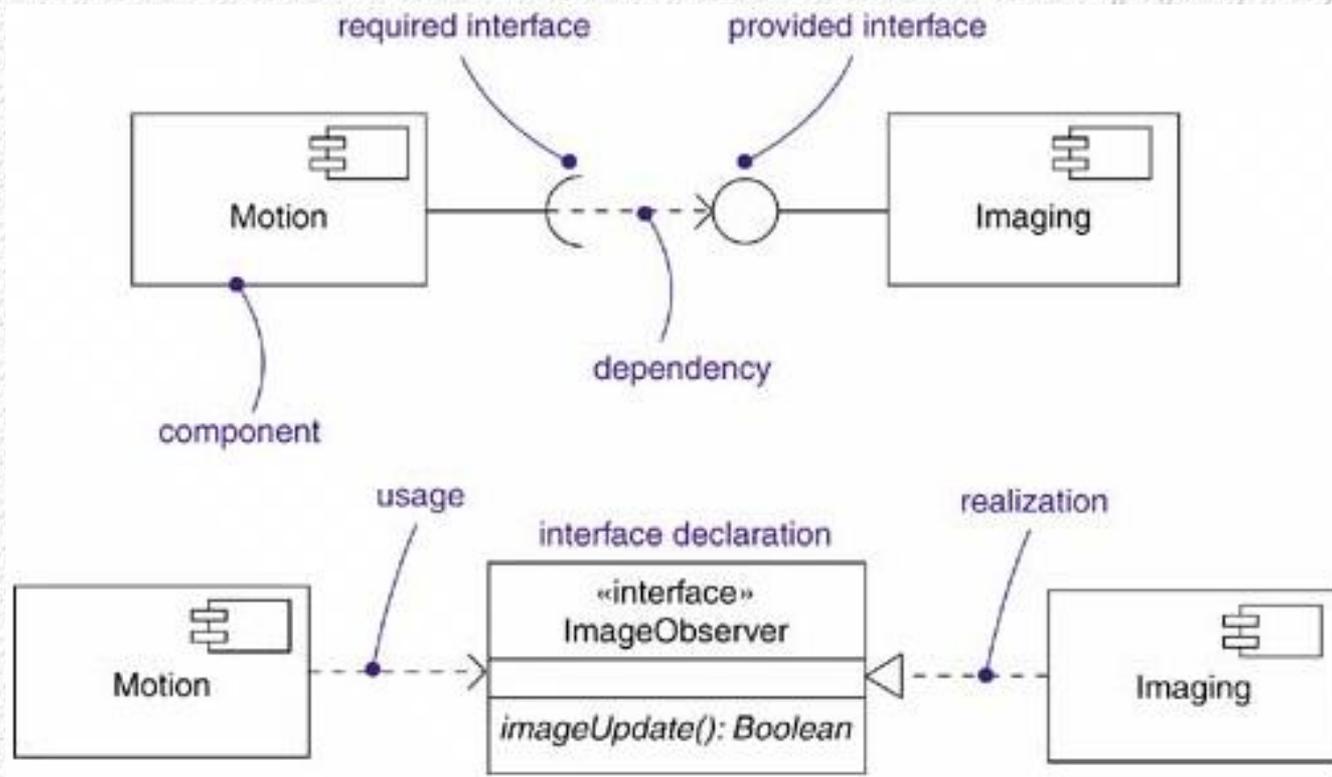
■ 构件图

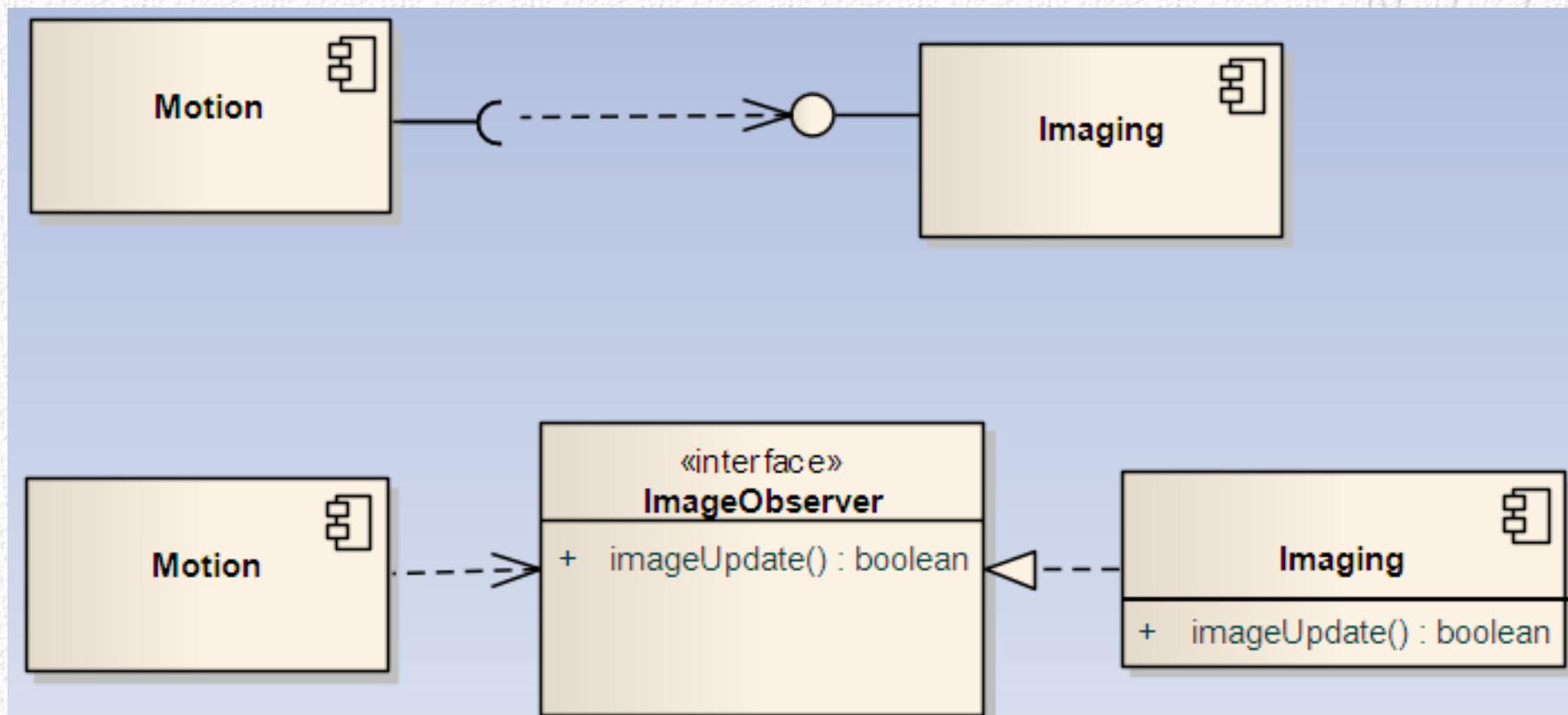
- 构件图是对面向对象系统的物理方面建模时使用的两种图之一。
- 构件图描述软件构件以及构件之间的关系
- 构件本身是代码的物理模块
- 构件图中通常包括三类元素
 - 构件
 - 接口
 - 关系

- 构件和接口之间的关系是非常重要的。
- 基于构件来构造系统，通过描述接口来分解系统，然后提供实现这些接口的构件和通过访问接口获得服务的其他构件。
- 构件所实现的接口称为供接口(**provided interface**)。一个构件可以声明许多供接口
- 构件所使用的接口称为需接口(**required interface**)。一个构件可以遵从许多需接口

■ 构件和接口关系的表示方式

- 简略形式（棒棒糖和插口）
- 展开形式





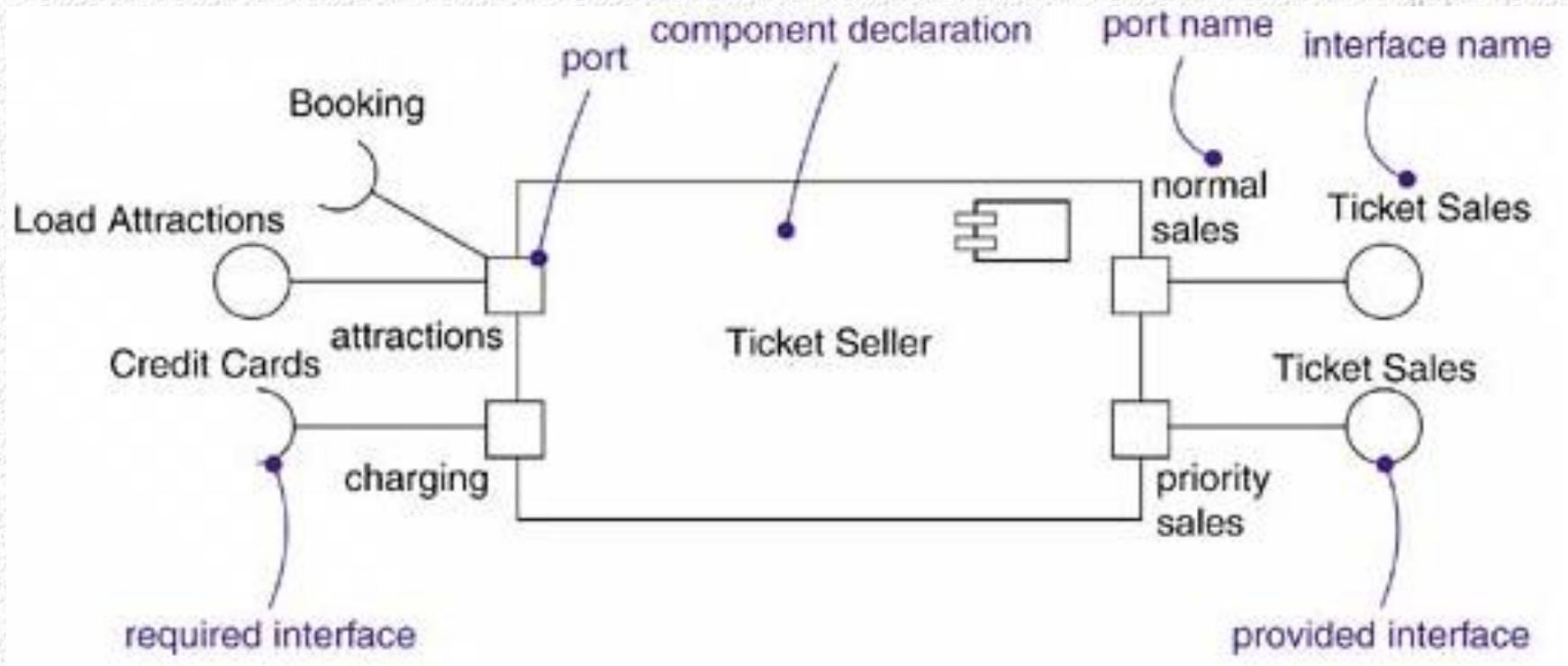


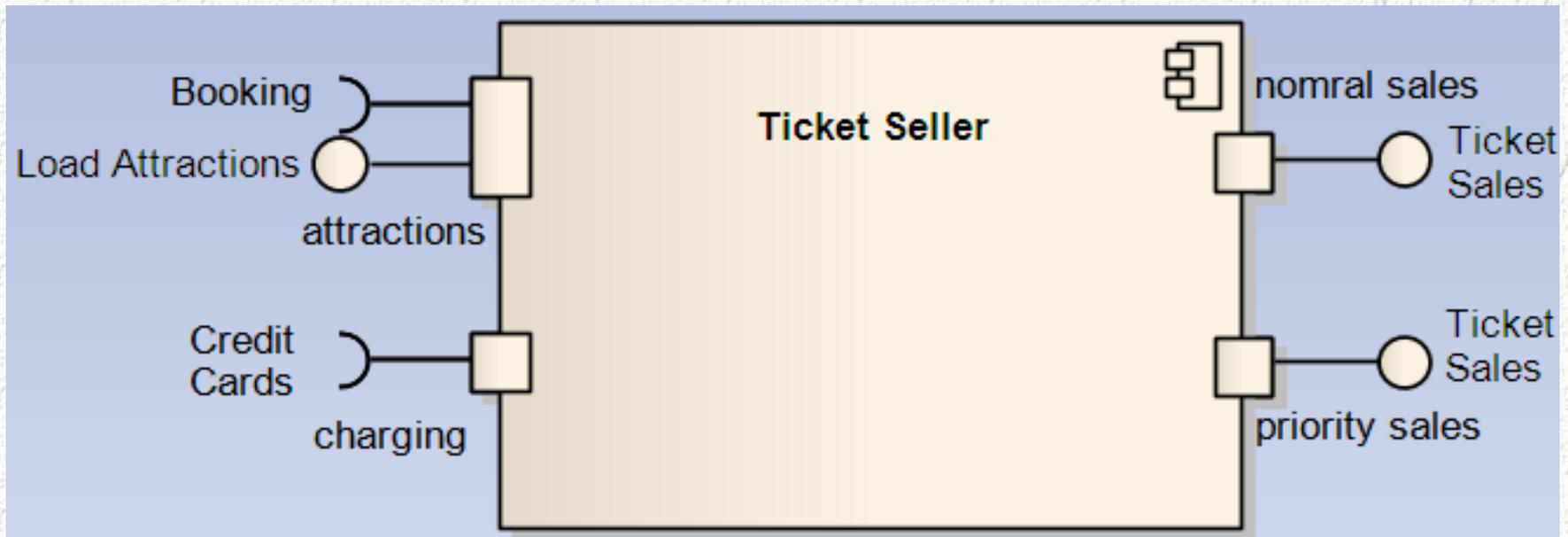
■ 可替换性

- 构件是可替换的，一个构件可以用遵循相同接口的其他构件来替换
- 构件是系统的一部分，构件很少单独存在

■ 端口

- 端口是一个被封装的构件的对外窗口
- 端口被表示成跨立于构件边界上的方块，表示穿过构件的封闭边界的洞。供接口和需接口都附着到端口符号上。供接口表示一个可以通过那个端口来请求的服务，需接口表示一个该端口需要从其他构件获得的服务。每个端口都有一个名字，可以通过构件和端口名来唯一标识。
- 端口是构件的一部分，端口实例随着它们所属构件的实例被创建和撤销。
- 端口也有多重性，以指明在构件实例中特定端口实例的可能数目



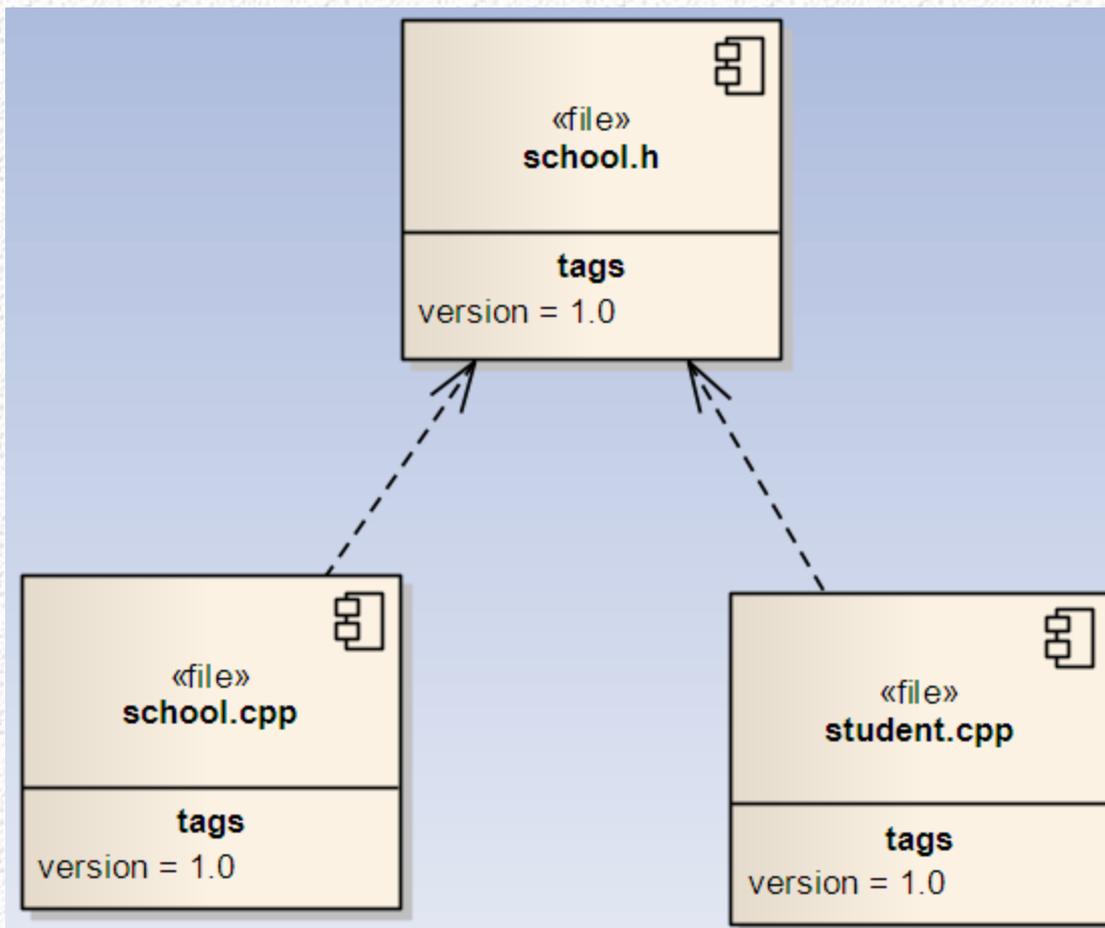


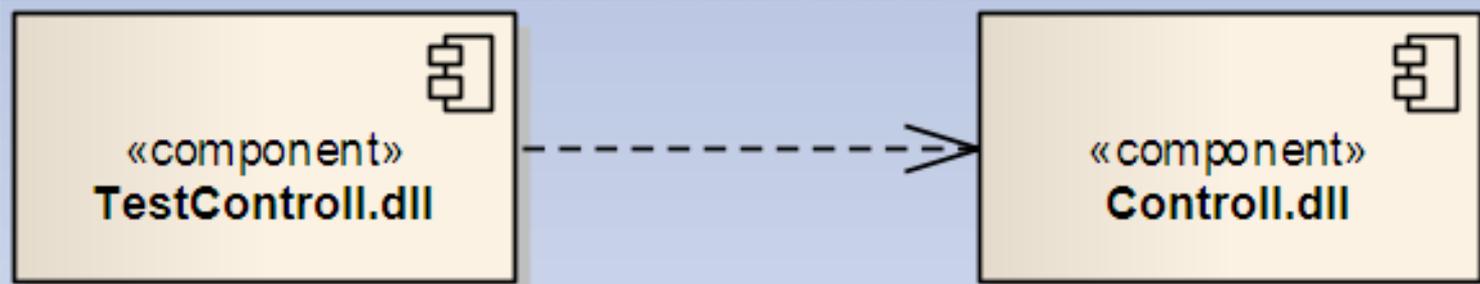
■ 内部结构

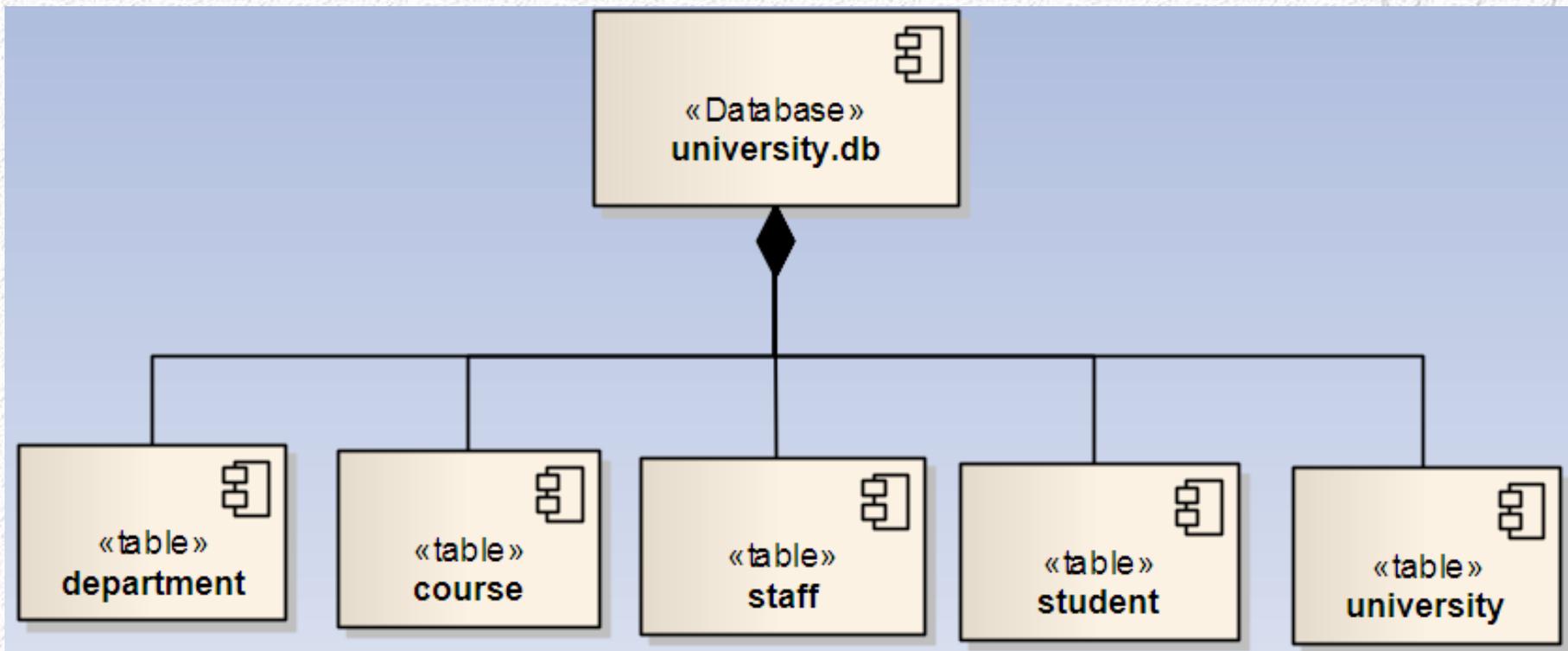
- 在一些大型系统中，理想的做法是用一些小的构件作为构造块来组建一些大构件
- 构件的内部结构是一些小的部件，这些部件和它们之间的连接一起组合成构件的实现

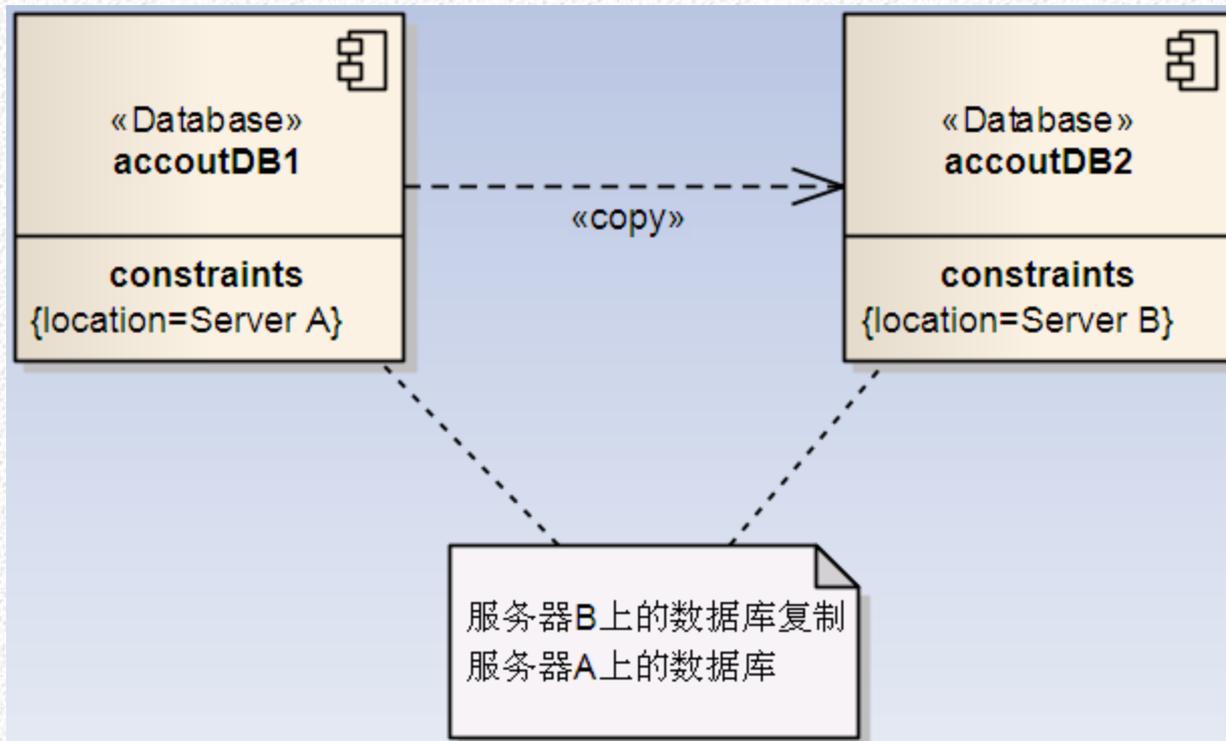
■ 用途

- 为源代码建模
- 为可执行版本建模
- 为数据库建模
- 为自适应系统建模





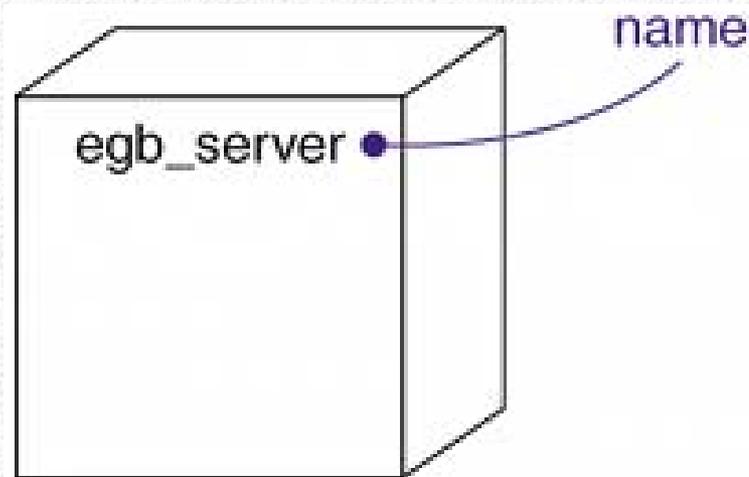




■ 部署图(Deployment Diagram)

- 部署图是面向对象系统的物理方面建模时使用的两种图之一（另一种图为构件图）
- 部署图显示了运行软件系统的物理硬件，以及如何将软件部署到硬件上。
- 部署图通常包括两种元素
 - 结点
 - 关系

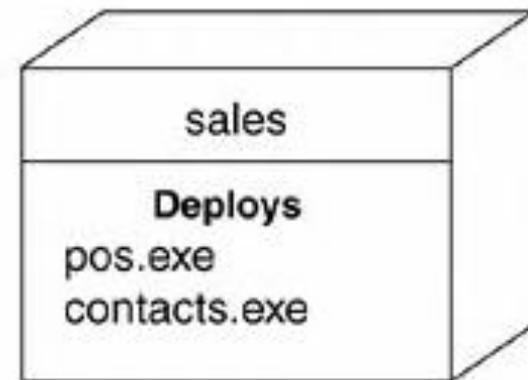
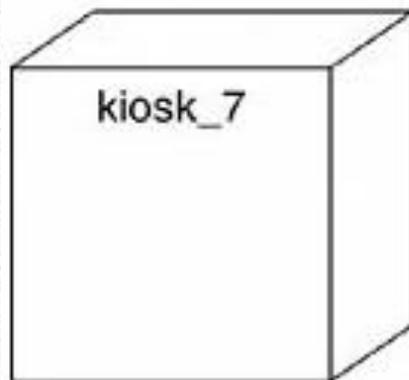
- 结点(node)是存在于运行时并代表一项计算资源的物理元素，一般至少拥有一些内存，而且常常具有处理能力。在图形上，把结点画成一个立方体



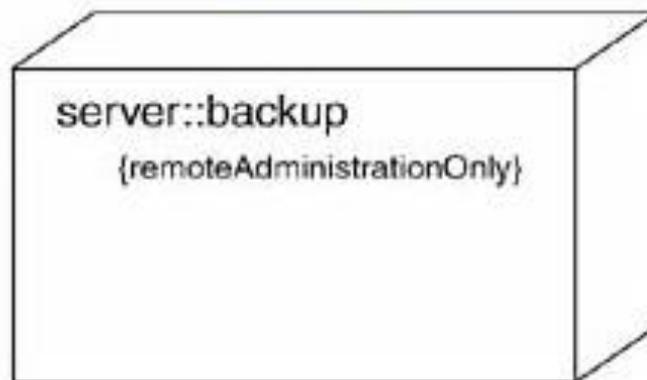
■ 名称

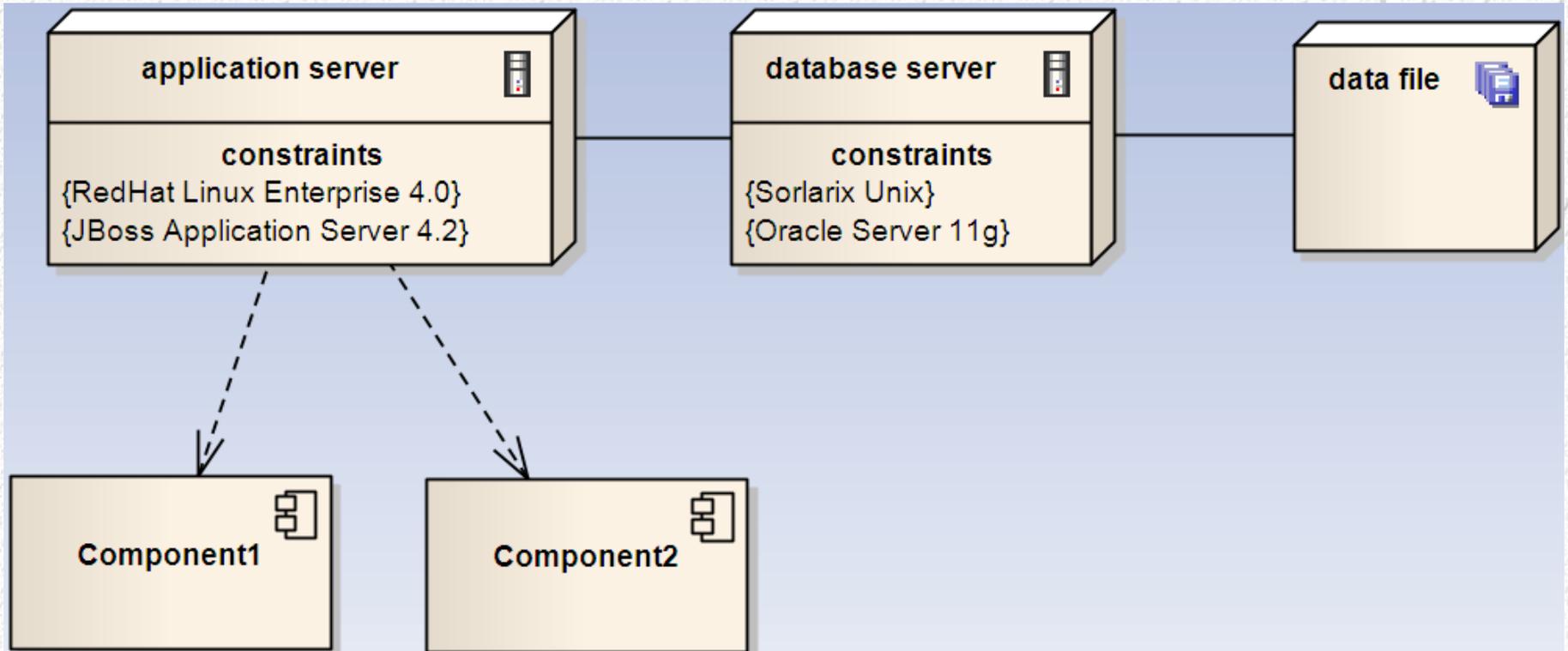
- 简单名
- 限定名

simple names



qualified names

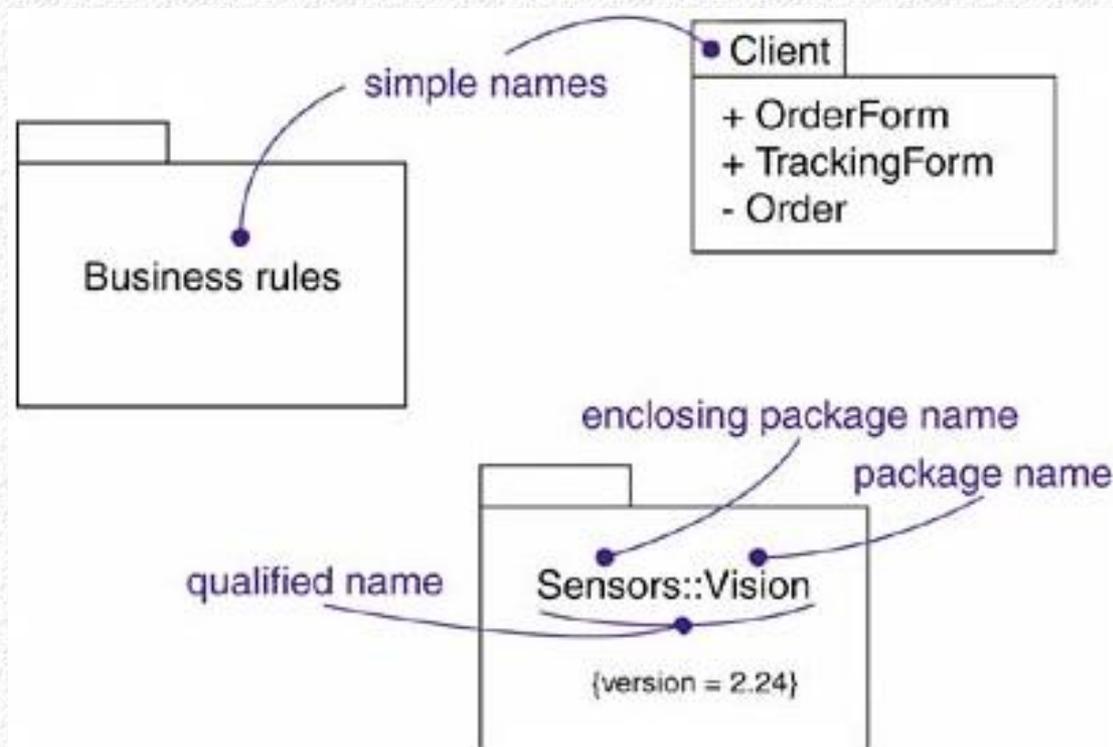


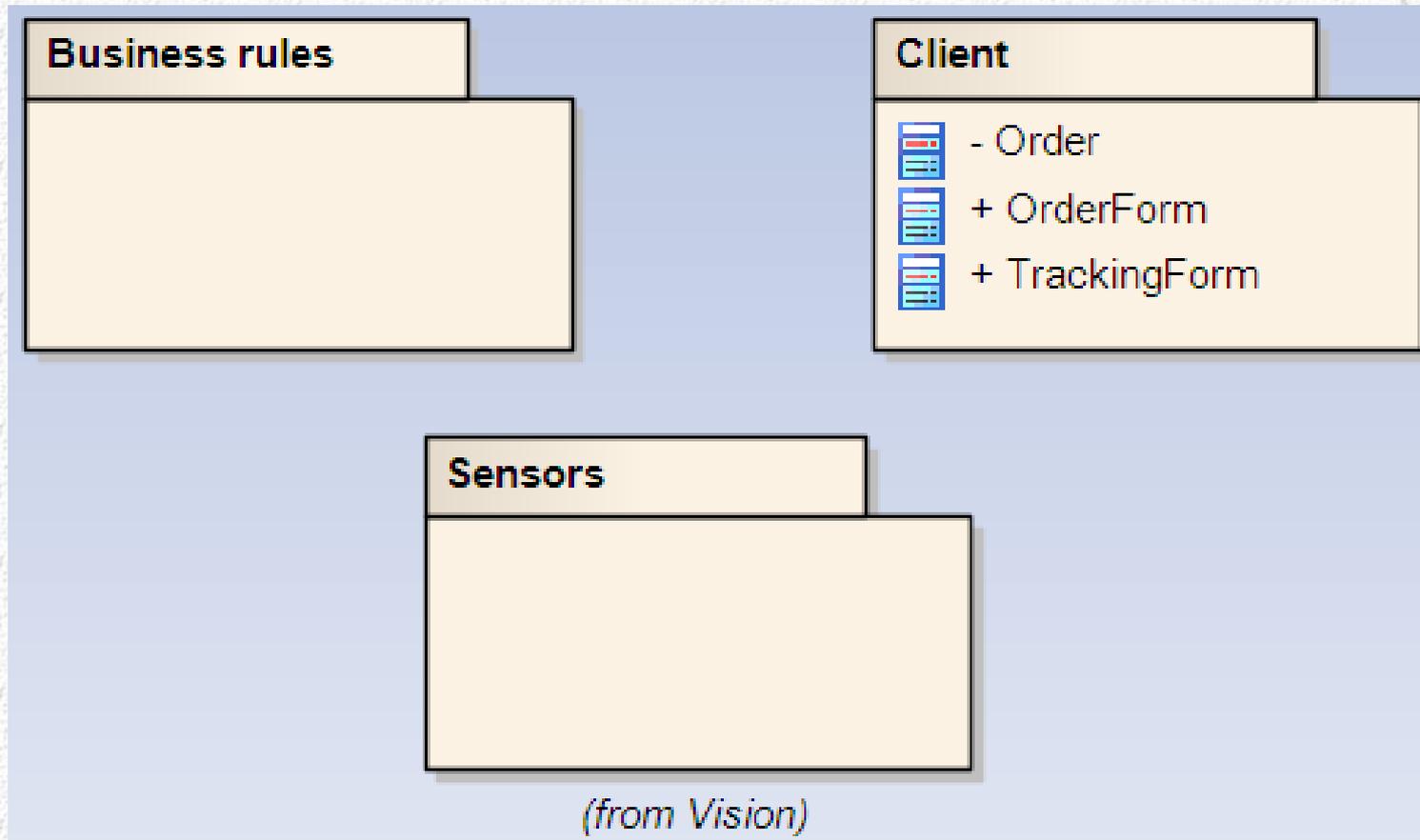


■ 用途

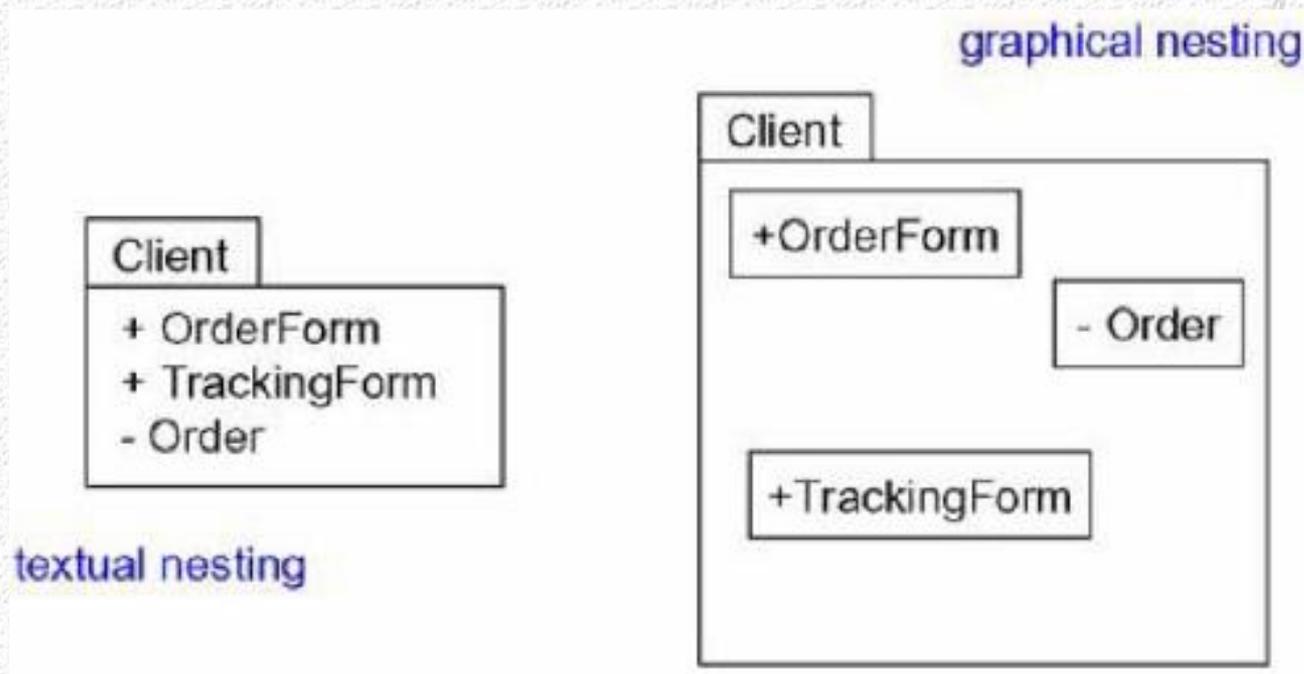
- 为嵌入式系统建模
- 为客户/服务器系统建模
- 为完全的分布式系统建模

- 包(package)是用于把模型本身组织成层次结构的通用机制，它不能执行。
- 名称：简单名、限定名



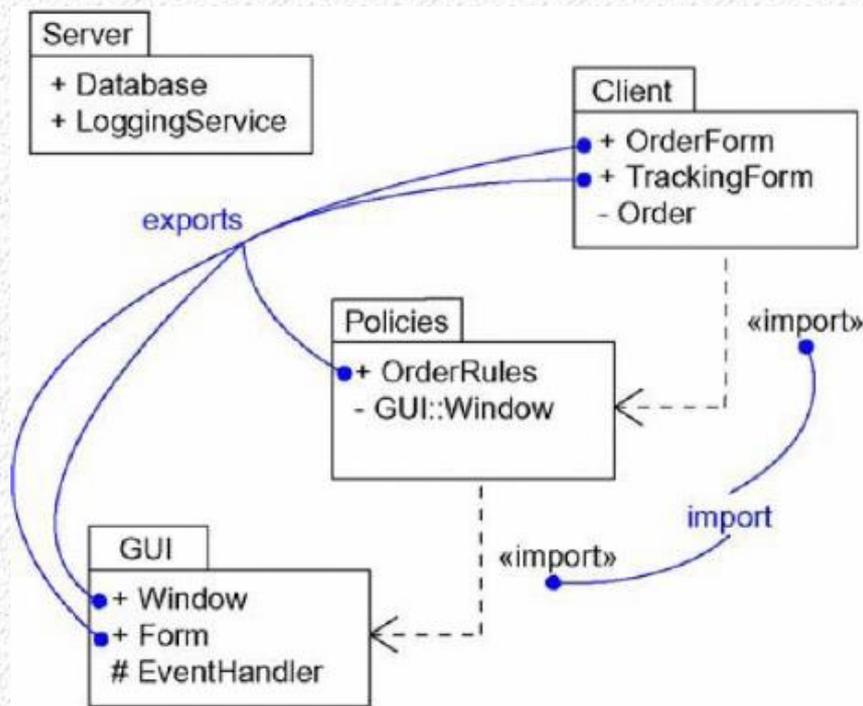


- 拥有的元素：可以是类、接口、构件、结点、协作、用况和图，甚至可以是其他包
- 可见性



■ 引入与示出

- 引入(**import**)把来自目标包中的公共元素添加到进行引入的包的公共命名空间中
- 包的公共部分成为它的引出(**export**)



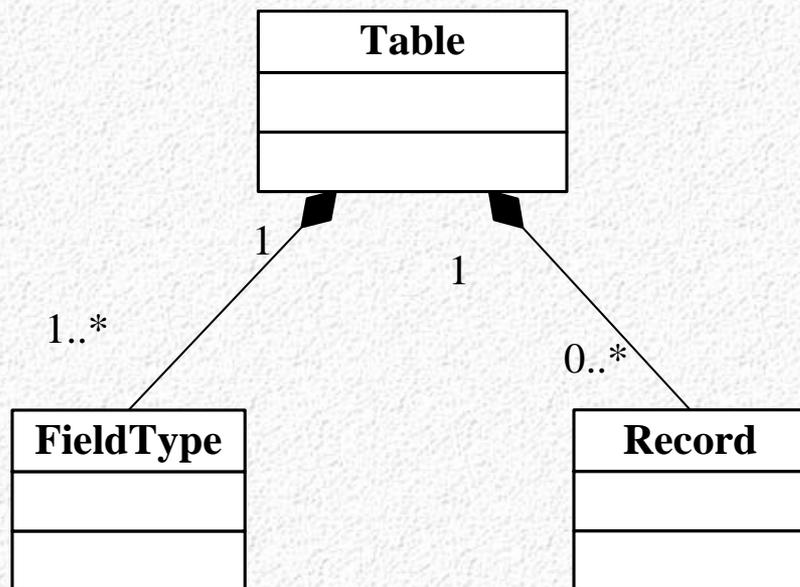


■ 用途

- 包图可用于任何类型的**UML**分类器进行分组，但一般更多用于对类或用例进行分类

- 組合結構圖是一種高級視圖，其顯示了如下內容：
 - **內部結構：** 顯示包含在類里的各個成員，以及各個成員之間的關係。
 - **如何使用類：** 顯示類如何通過端口作用於系統。
 - **合作：** 顯示系統中一組對象共同協作完成某件事。

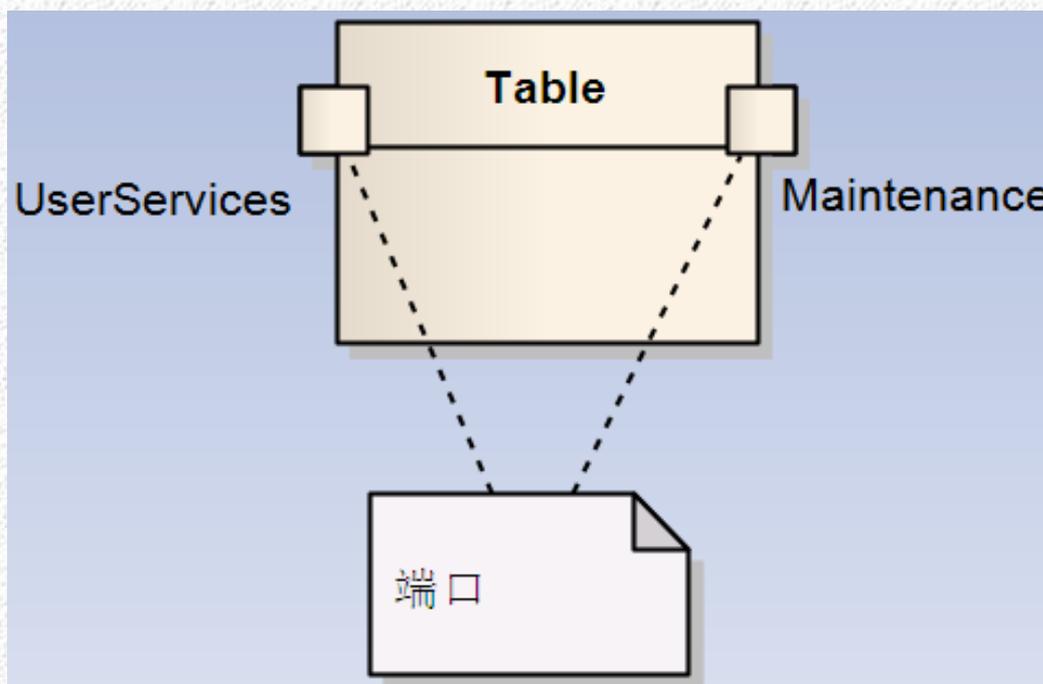
- 在介绍类图时，曾介绍类之间的关系，包括关联和组合。组合结构图提供了显示这些关系的替代方式。描述了类图中的组合关系，通过组合关系显示**Table**包含**Record**和**FieldType**类型的对象。





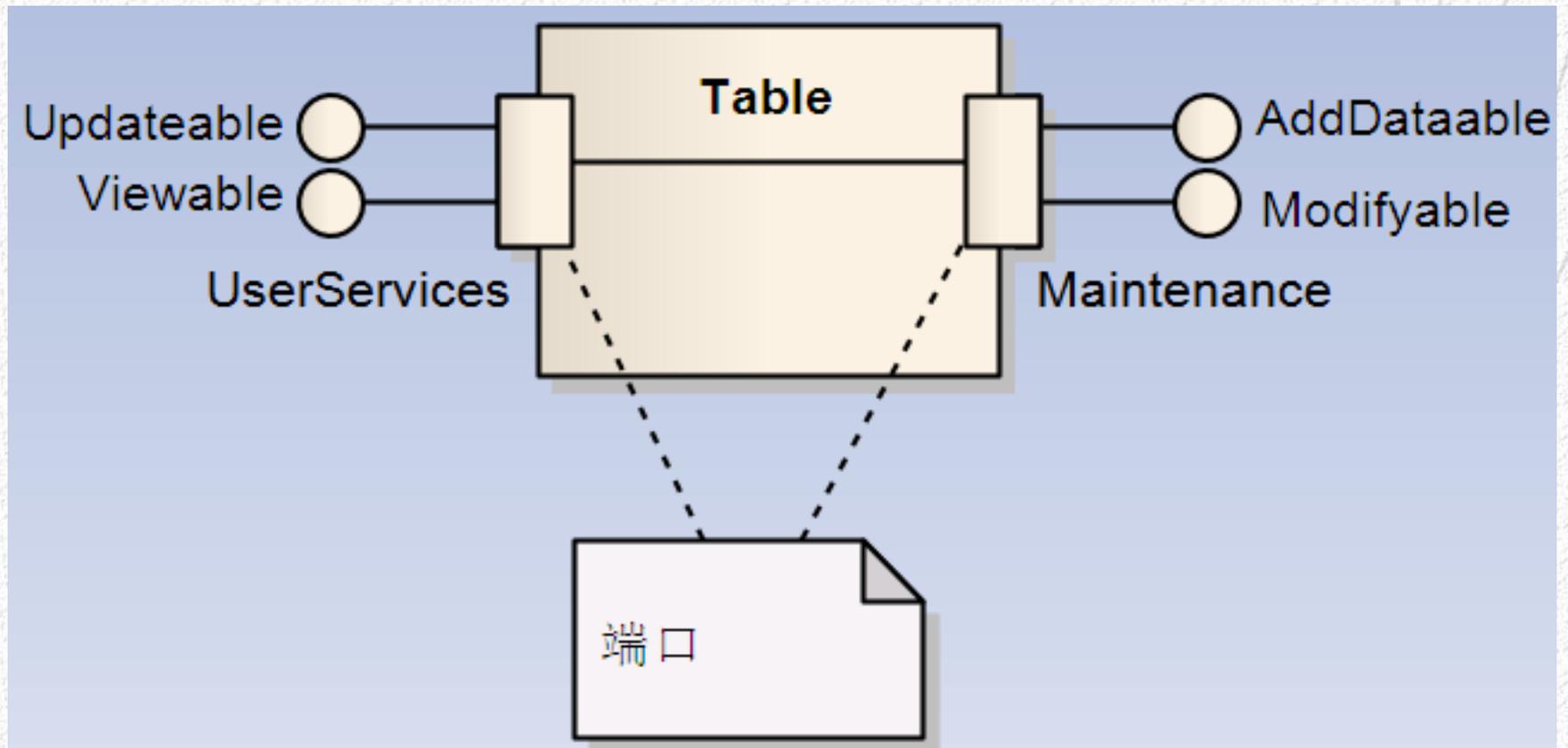
- 类的内部结构主要关注于类的内容，为了显示一个类如何被其他类所用，组合结构图引入了端口。端口侧重于类的外部，显示一个类如何被其他类所使用。
- 端口是类与外部世界之间的交互点。端口通常是通过不同类型的客户，以表示类的不同使用方法。例如，一个Table类可以有两种不同的用法：
 1. 用户查看及编辑此Table。
 2. 管理员则可以修改数据、添加数据，以及修改Table结构。

- 类的每一种使用方法以不同的端口表示，端口以类边界上的一个小矩形表示，如上图，在端口的旁边添加一个名称，以显示端口的目的

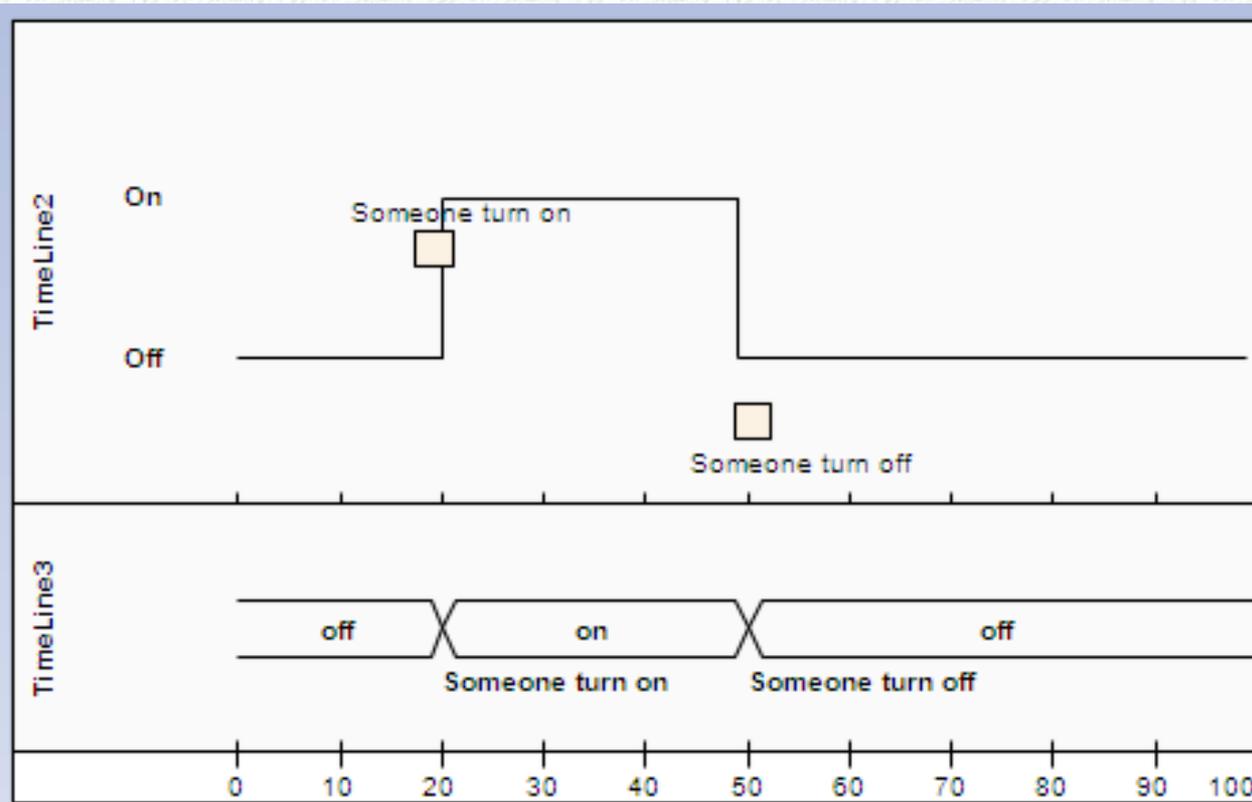




回顾类图，类能实现接口。当类实现一个接口时，该接口称为此类的**提供接口**，一个提供接口可以使其他类通过该接口访问此类。与提供接口相反，类还可以有**需求接口**。需求接口是类运行的接口，即该类需要一个实现该接口的类来为其提供服务。如下图所示的 Table 类实现了 updateable 和 Viewable 接口，让其他类通过这些接口修改及查看 Table 类。这些接口与端口 Userservices 关联；而 Maintenance 端口则与 Modifyable 接口和 AddDataable 接口关联，其中 Modifyable 接口是一个需求接口，是 Table 用于修改表结构的服务， AddDataable 接口是提供接口，以便管理向表中添加数据。

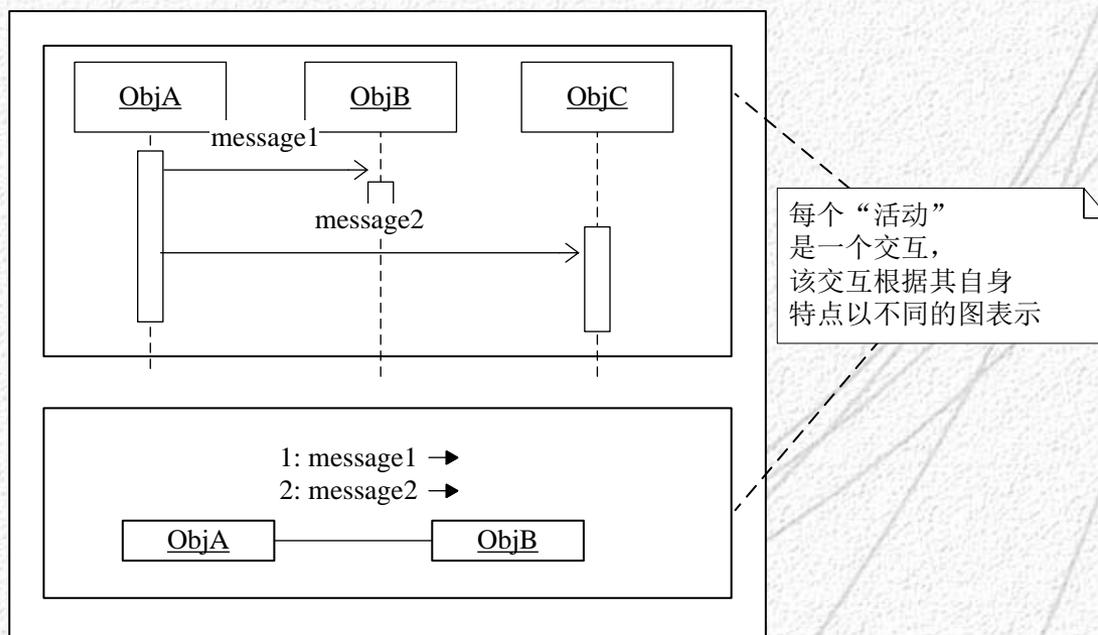


- 定时图是用来描述对象的状态（或某数值）随时间变化而变化的图，一般软件开发中很少会用到。



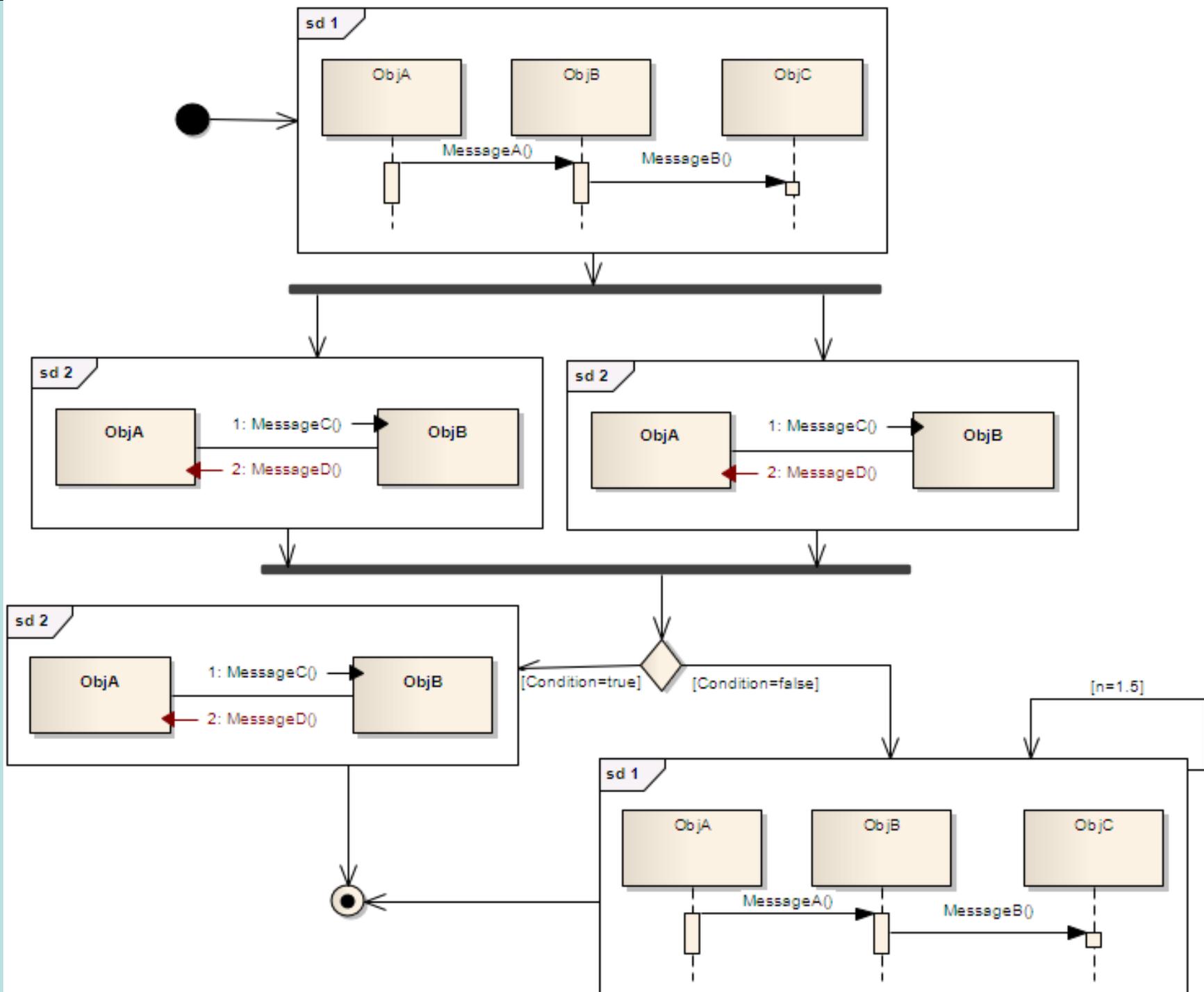
- 顺序图、通信图和时序图主要关注特定交互的具体细节，而交互概况图则将各种不同的交互结合在一起，形成针对系统某种特定要点的交互整体图。
- 交互概况图的外观与活动图类似，只是将活动图中的动作元素改为交互概况图的交互关系。如果概况图内的一个交互涉及时序，则使用时序图；如果概况图中的另一个交互可能需要关注消息次序，则可以使用顺序图。交互概况图将系统内单独的交互结合起来，并针对每个特定交互使用最合理的表示法，以显示出它们如何协同工作来实现系统的主要功能。

- 交互概況圖不僅外觀上與活動圖類似，甚至在理解上也可以以活動圖標準，只是以交互代替了活動圖中的動作。交互概況圖中每個完整的交互都根據其自身的特点，以不同的交互圖來表示，如圖所示。





- 交互概况图与活动图一样，都是从**初始节点开始**，并以**最终节点结束**。在这两个节点之间的控制通过两者之间的所有交互。并且，交互之间不局限于简单按序的活动，它可以有判断、并行动作甚至循环，





■ 说明:

- 从初始节点开始，控制流执行第一个顺序图表示的交互，然后并行执行两个通信图表示的交互，然后合并控制流，并在判断节点处根据判断条件值执行不同的交互：当条件为真时执行通信图表示的下一个交互，交互完成后结束，而条件为假时执行下一个顺序图表示的交互，该交互在结束之前将循环执行 **5** 次。

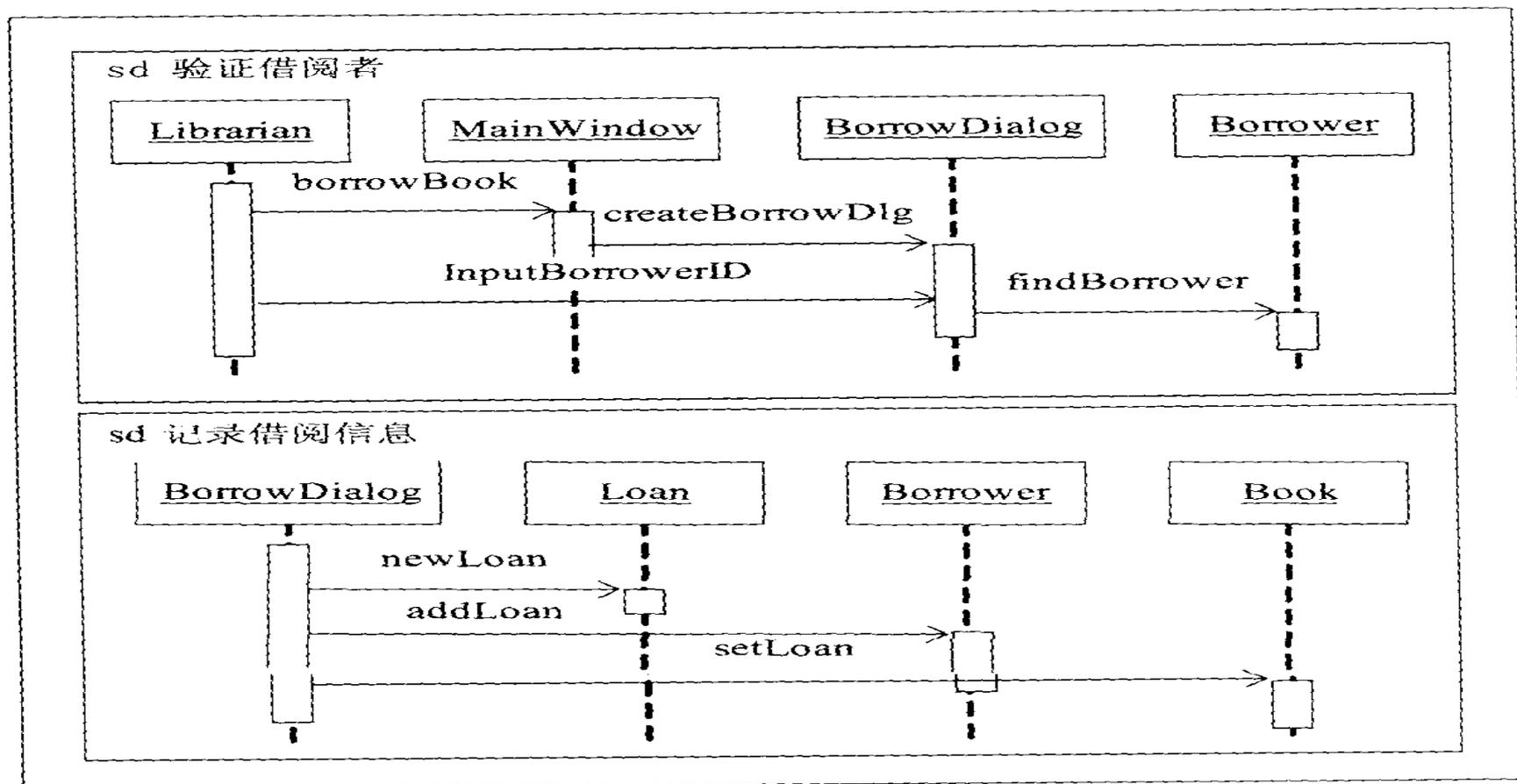
为用例建模交互概况图

- 在使用交互概况图为用例建模前，先回顾一下活动图。活动图以活动、状态、控制流等方式描述了系统的用例，而交互概况图是以顺序图、通信图和时序图描述用例，这两者以不同的方式描述了相同的内容。交互概况图以活动图的形式，交互图的内容对用例进行描述。

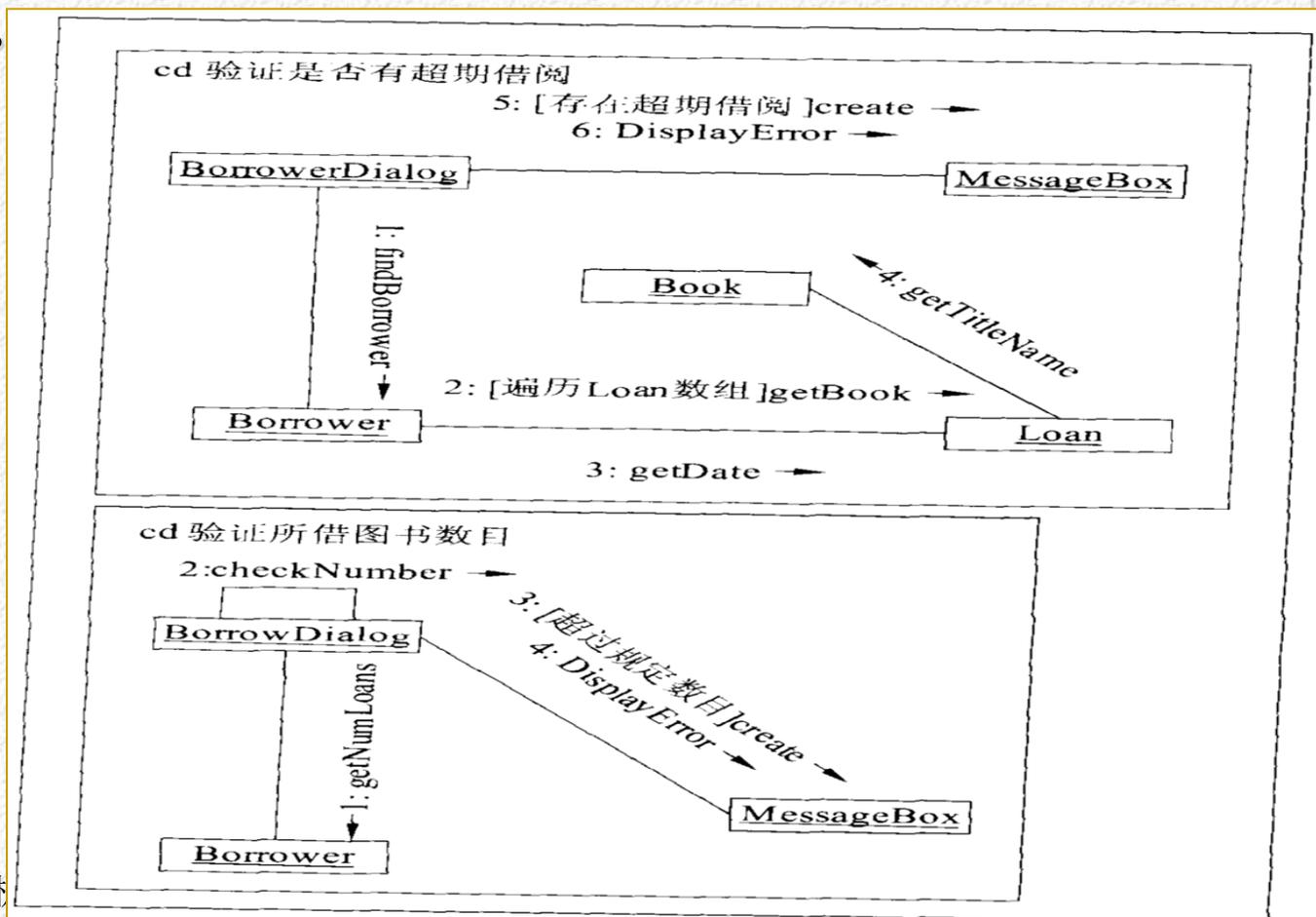


- 以交互概况图为用例建模时，首先必须将用例分解成单独的交互，并确定最有效表示交互的图类型。例如，对“**图书管理系统**”中的**借书用例**的基本操作流程而言，它可以分为如下几个交互：
 - 验证借阅者身份
 - 检验借阅者是否有超期的借阅信息
 - 获取借阅的图书信息
 - 检验借阅者借阅的图书数目
 - 记录借阅信息

- 對於交互“**驗證借閱者身份**”和“**記錄借閱信息**”而言，消息的次序比任何其他因素都重要，因此對這些交互使用順序圖。此處可以重用建模順序圖中的相關步驟，如圖所示。



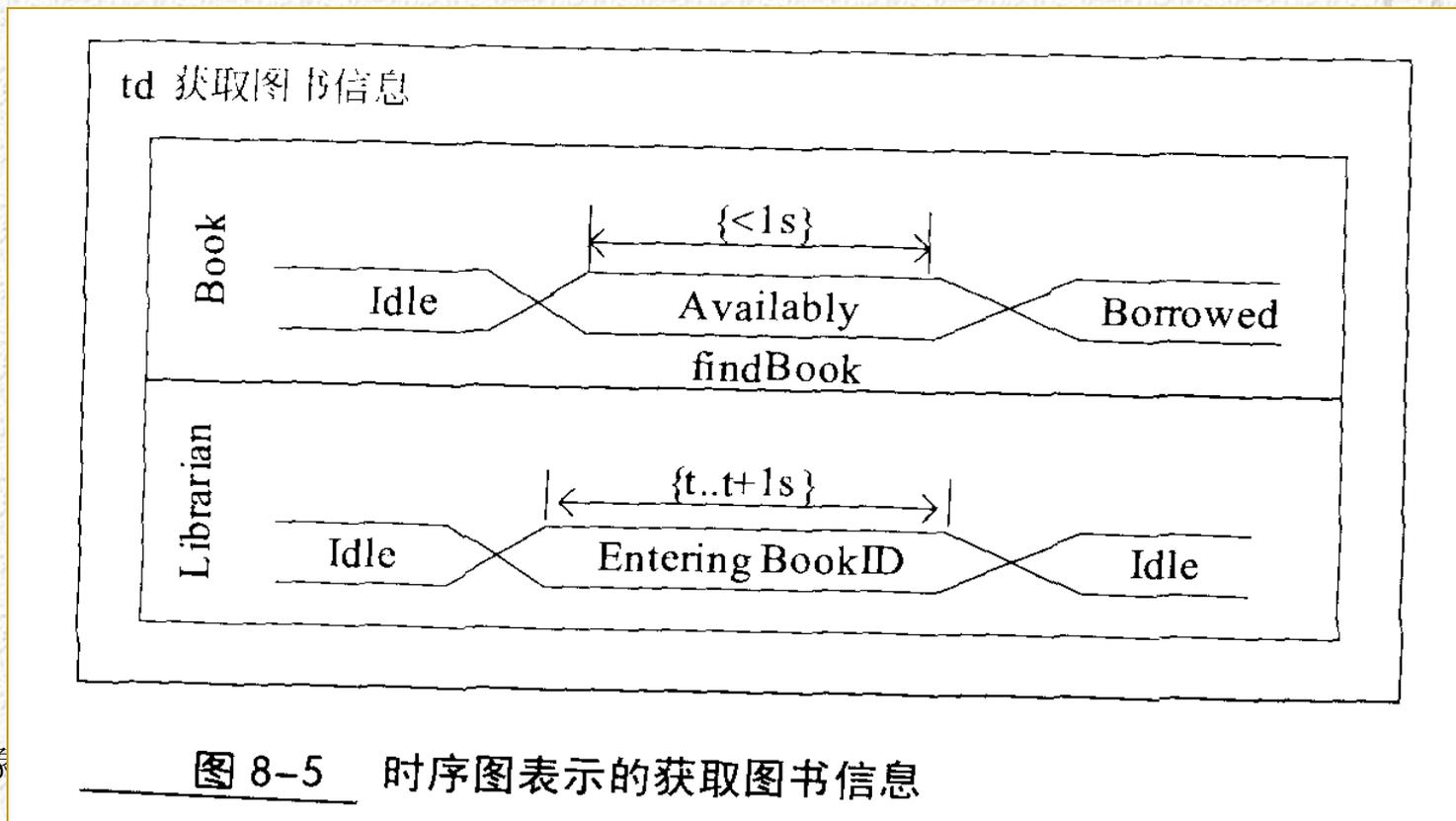
- 为了使交互概况图中的交互多样化，“检验借阅者借阅的图书数目”和“检验借阅者是否有超期的借阅信息”交互将以通信图表示，如图所示。



面向对象分析

图 8-4 通信图表示的验证借阅是否超期和超过规定数目

- 假设“获取借阅的图书信息”交互对时间非常敏感，它要求整个交互要在 1 秒内完成。这部分交互主要关注时序，并且交互概况图能包含任何不同的交互图类型，因此这部分交互在交互概况图中可以用定时图表示。





- 在分析交互概況图中的各个交互后，下一步就是将所有的交互根据操作步骤，使用控制线将各个交互连接起来形成一幅图——交互概況图。

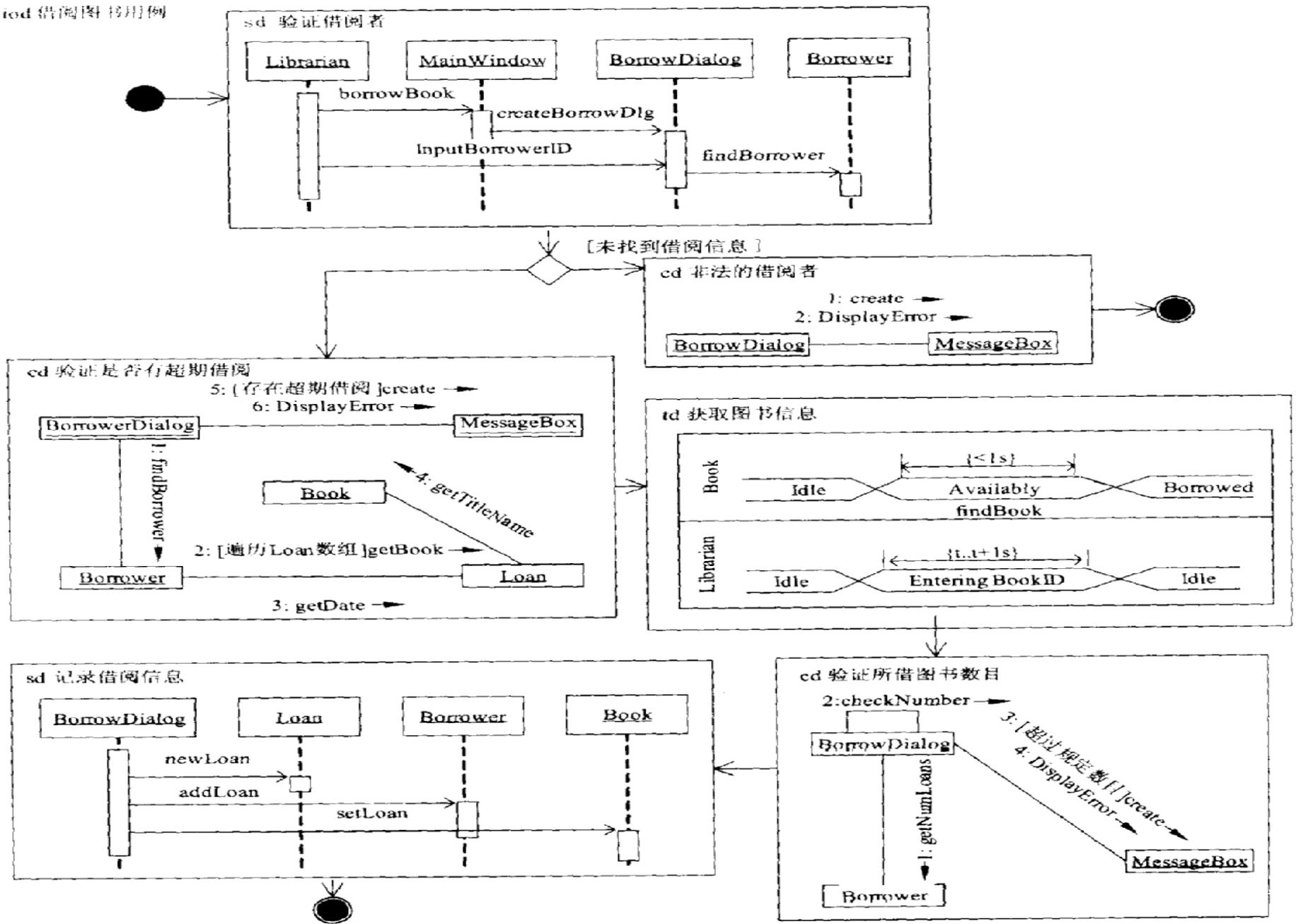


图 8-6 借阅图书用例的交互概况图