基于UML的构件建模

北京大学　信息科学技术学院　软件所

麻志毅

第一部分 UML 2.0

- 构件规约
- 制品
- 部署
- 对体系结构建模

第二部分 一种建立面向构件的系统分析模型的方法

- 构件元模型以及构件规约
- 建立面向构件的系统分析模型过程指南

第一部分 UML 2.0

- **UML支持对逻辑构件（如业务构件、过程构件）和物理构件（如EJB构件、CORBA构件、COM+构件和.NET构件以及WSDL构件）的规约；**
- **UML支持对实现构件的制品、对可执行的构件的部署和对构件执行在其上的节点的规约。**

——这意味着在基于构件的开发中，在不同的阶段有着不同的构件模型，供不同的开发人员使用。

1 构件规约

1.1 构件图

1.1.1 构件的定义

在《计算机百科全书》中，把构件定义为在构件软件系统中具有相对独立功能、可以明确辨识、接口由契约指定、语境有明显依赖关系、可独立部署且多由第三方提供的可组装软件实体。

按照UML2.0的定义,构件是系统的模块化部分，它封装了自己的内容，且它的声明在其环境中是可以替换的；构件利用提供接口和请求接口定义自身的行为，它起类型的作用。

上述定义有如下几方面的含义：

1) 一个构件表示系统的一个模块部分，而且是一个自包含的单元，它封装了其内部成分的状态和行为。

2) 按照提供和请求接口，构件定义其行为；

3) 构件是可替换的单元，在设计时和运行时基于接口的兼容性，若一个构件能提供功能与另一个构件相同的功能，则前者就能替换后者。

4) 构件起类型的作用，可执行的构件是可实例化的。

由于在UML2.0中，构件继承了类，故：

5）构件具有属性、操作和可见性，并能参与关联和泛化。

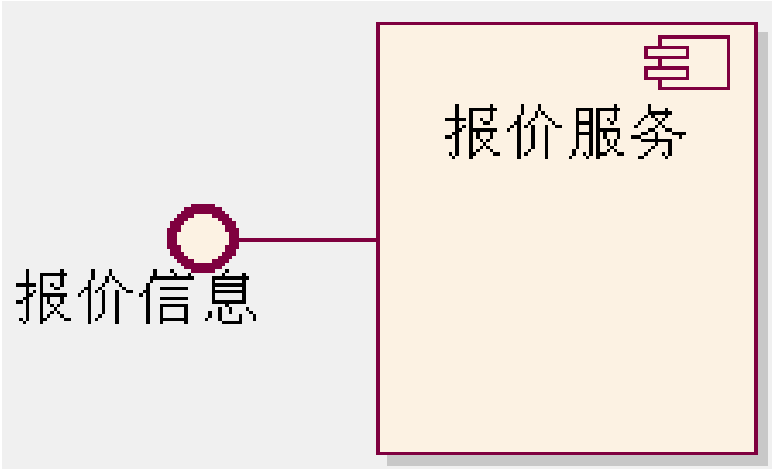
UML2.0把构件分为基本构件和包装构件。

基本构件注重于把构件定义为在系统中可执行的元素。

包装构件扩展了基本构件的概念，它注重于把构件定义为一组相关的元素，这组元素为开发过程的一部分。也即, 包装构件定义了构件的命名空间方面。在构件的命名空间中，可以包括类、接口、构件、包、用况、依赖（如映射）和制品。按照这种扩展，构件也具有如下的含义：

6）可以用构件来装配大粒度的构件，方法为把所复用的构件作为大粒度构件的成分，并把它们的请求和提供接口连接在一起。

构件的表示法



<<component>>
订单

<<provided interfaces>>
订单登记
开帐单
<<required interfaces>>
发票
建立(…)
登记费(…)

1.1.2 构件的接口

接口是表示对一组相关的操作进行声明的一种建模元素，它指定了一种契约，这契约必须由实现这个接口的构件的任何实例完成。

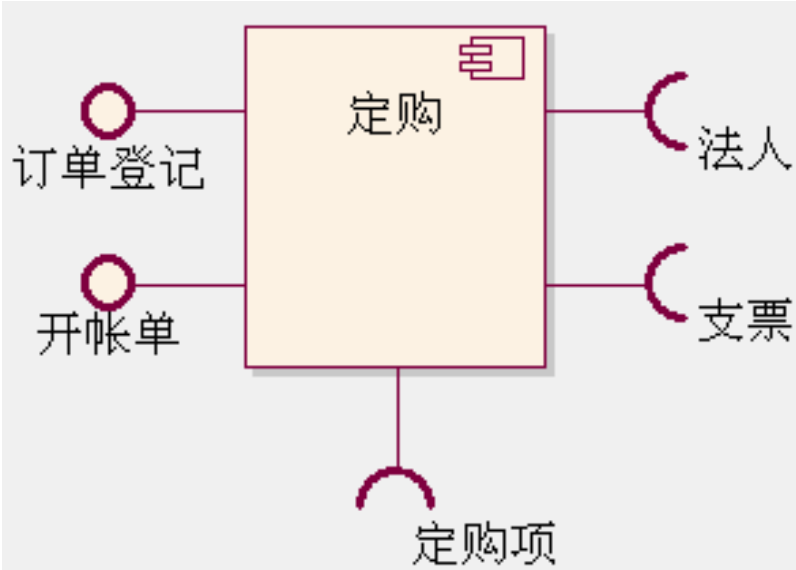
可以按各种约束（如前置和后置条件）的形式把一个接口与一个职责相关联，可以对通过这个接口的交互规定次序。

接口分提供接口和请求接口。

把构件实现的接口称为**提供接口**，这意味着构件的提供接口是给其它构件提供服务的。构件可以直接实现提供接口，构件的子构件也可以实现提供接口。实现接口的构件支持由该接口所拥有的特征，此外，构件必须与接口拥有的约束相容。

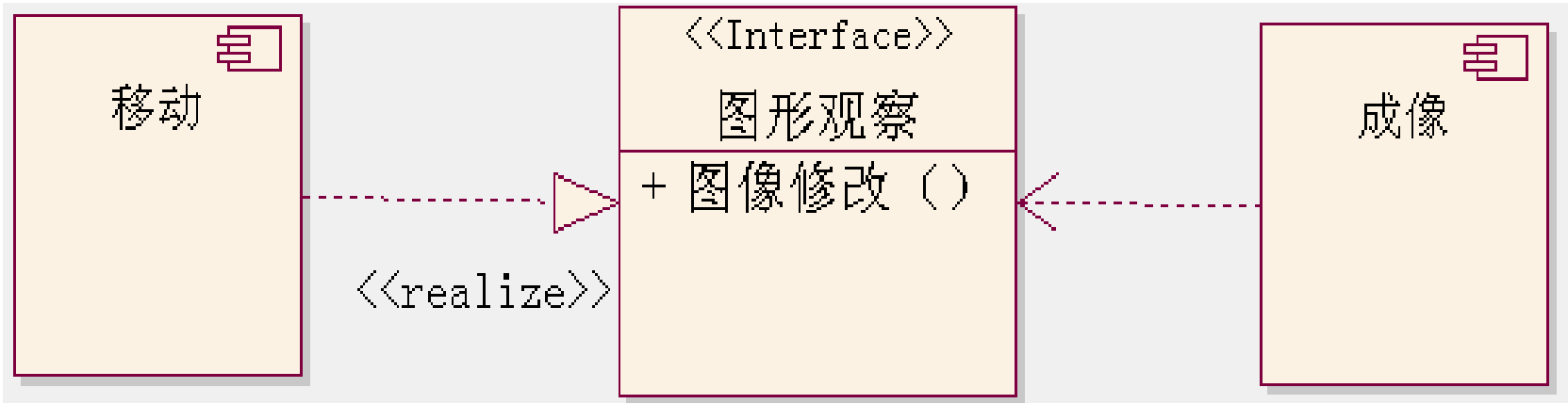
构件使用的接口被称为**请求接口**，即构件向其它构件请求服务时要遵循的接口。

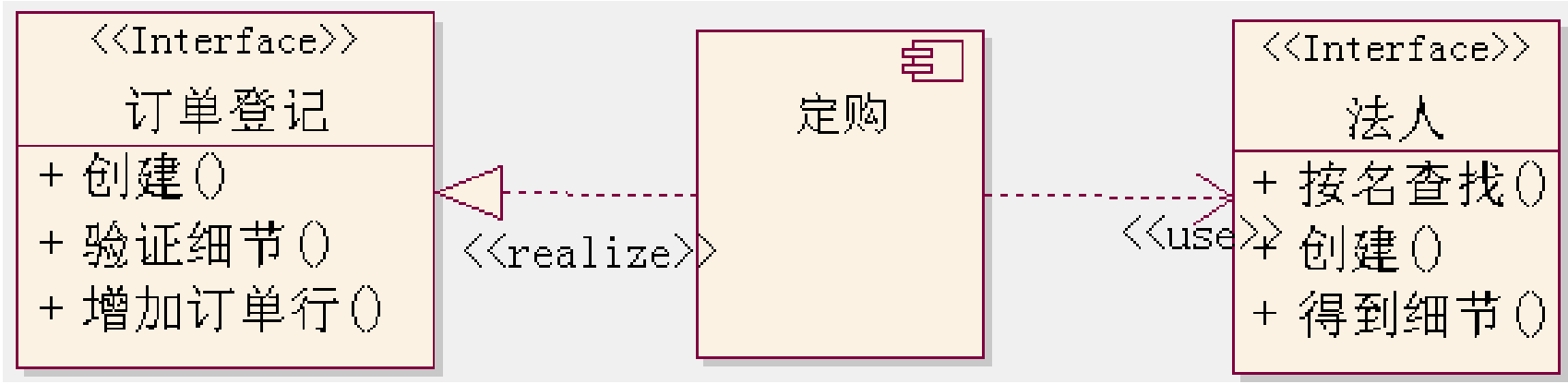
一个构件可以实现多个接口，一个构件可以请求多个接口，一个接口可以由多个不同的构件实现。



只要一个构件服从由另一个构件的提供和请求接口表达的约束，它就能和这样的环境进行交互。

通过添加新增功能的新构件类型，能够扩展系统。

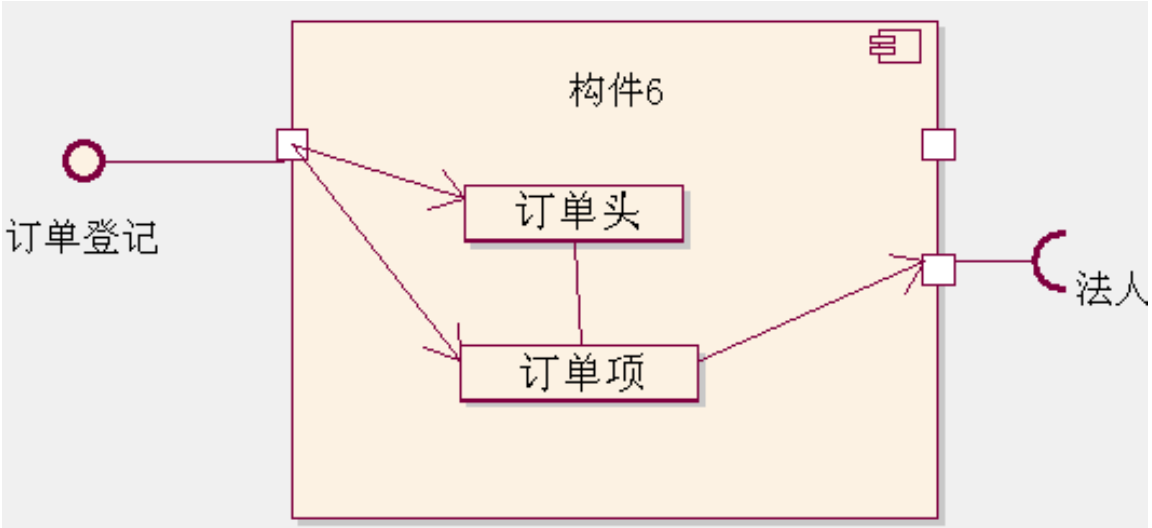

移动
<<Interface>>
图形观察
+ 图像修改 ()
成像
<<realize>>


<<Interface>>
订单登记
+ 创建()
+ 验证细节()
+ 增加订单行()
<<realize>>
定购
<<Interface>>
法人
+ 按名查找()
+ 创建()
+ 得到细节()
<<use>>

1．1．3构件的端口

接口对声明一个构件的总的行为来说是有用的，构件的实现仅需保证要实现在全部的提供接口中的操作。

使用端口是为了能进一步控制这种实现。



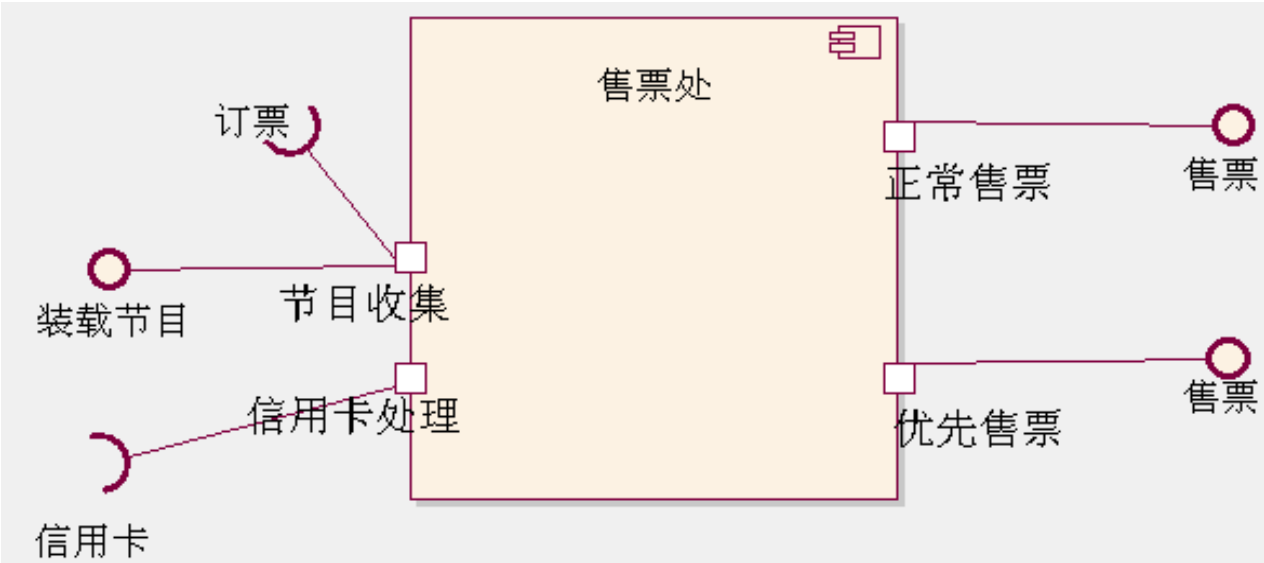
端口是一个封装构件的显式的对外窗口，是有标识的。在一个封装的构件中，所有进出构件的消息（调用或信号）都要通过端口。

可用一个显式的对象实现端口，它也可能只是一个虚拟的概念。

端口允许把构件的接口分成离散的组，并独立使用。

端口是构件的一部分，端口的实例随着它们所属的构件的实例一起被创建和撤消。

端口也可以具有多重性，用于指定在构件实例中特定端口的可能数目。构件实例中的每一个端口都有一个端口实例的排列。虽然排列中的端口都满足同样的接口并接受同种类的请求服务，但它们可能有不同的状态和数据值。例如，具有较高优先级的端口实例优先被满足。



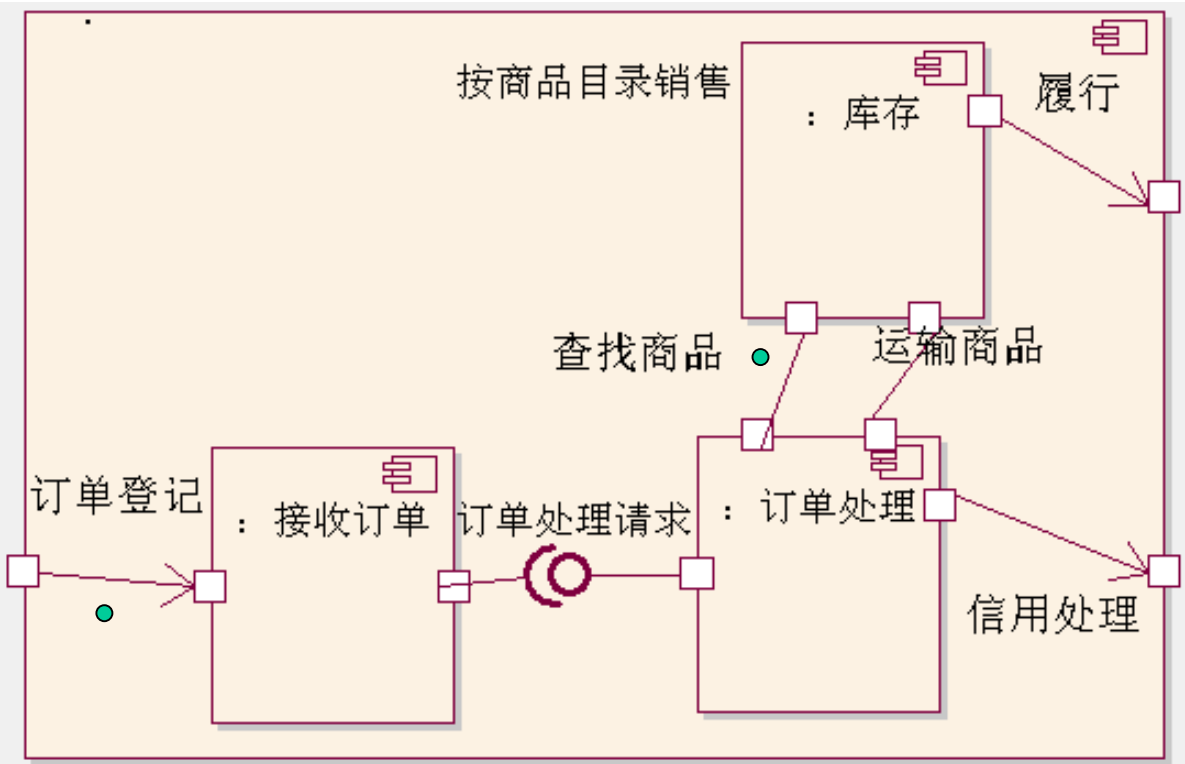
端口“节目收集”处定义了一个协议。下图的装配连接子“订单请求处理”处也定义了一个协议。

1．1．4 连接件

如果一个端口提供一个特定的接口而另一个端口需要这个接口，且接口是兼容的，那么这两个端口便是可连接的。

连接端口意味着请求端口要调用提供端口中的操作，以得到服务。设立端口和接口的优点在于在设计时，两个构件彼此不需要了解对方的内部，只要它们的接口是相互兼容的即可。如果需要使用新的构件，可以把这些端口重新连接到其它提供相同接口的构件上。

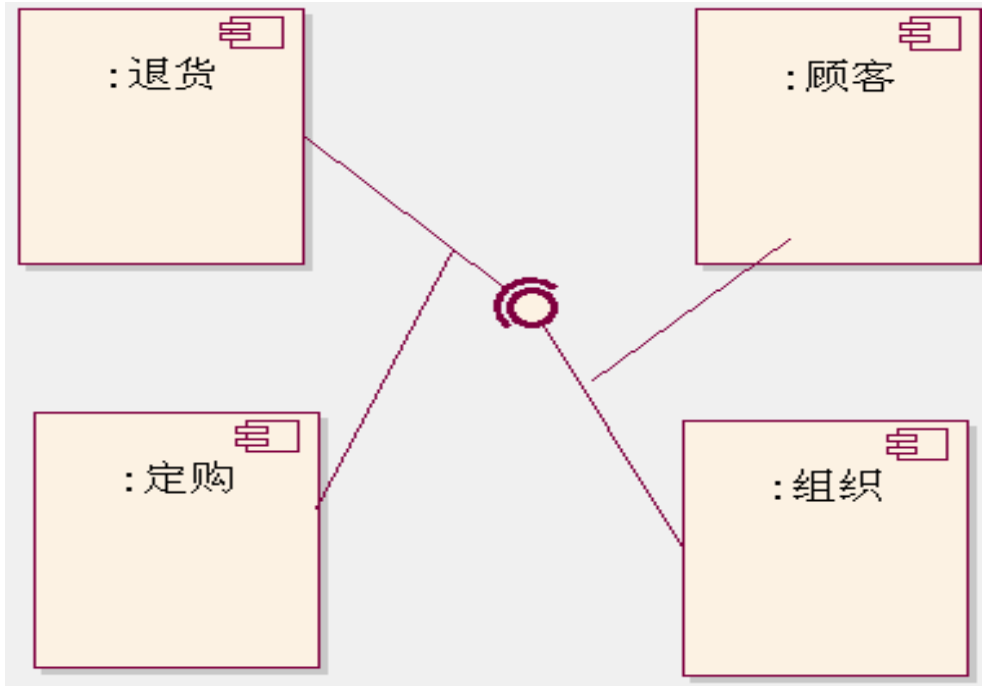
把两个端口之间的衔接称作为连接件。



为了连接构件或把端口与构件内的部件相连，UML定义了两种连接件：委托连接件和装配连接件。

装配连接件是两个构件间的连接件，它定义一个构件提供服务，供另一个构件请求。它是定义从一个请求接口或端口到一个提供接口或端口的连接件。

装配连接件的执行时语义，是信号沿着一个连接件的实例传递，该信号起源于一个请求端口，并被交付到一个提供端口。从一个单请求接口或端口到在不同构件上的提供接口的多个连接件指明，要处理信号的实例将在执行时决定。类似地，连接到一个单提供端口的多个请求端口指明，这个请求可以起源于不同的构件类型实例。



委托连接件把外部的构件契约（与端口指定的一样）链接到行为的内部实现（由构件的部件完成），它是行为的声明，在构件实例级别上使用。

用委托连接件对行为的层次分解建模，由构件提供的服务最终可以由嵌套在构件内部（可多层）的构件实现。

一个端口可以委托在从属构件上的一组端口，这组端口必须共同地提供所需要功能。

1．1．6 构件的内部结构

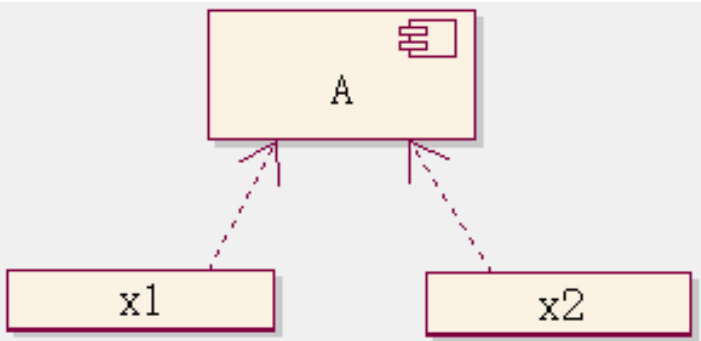
构件由实现它的一个或多个部件组成，把这些部件以及其间的关系称之为它的内部结构。

装配连接件和委托连接件：内外交互

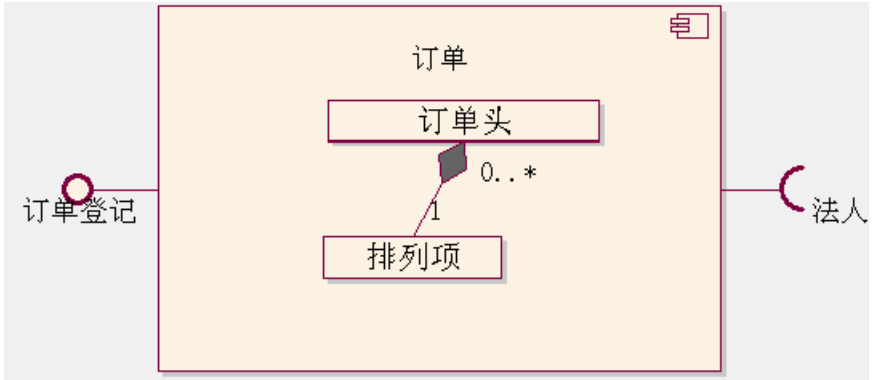
组合结构图、交互图、活动图：用于展示如何在内部实现外部行为。

构件与实现它的制品

1、使用依赖关系



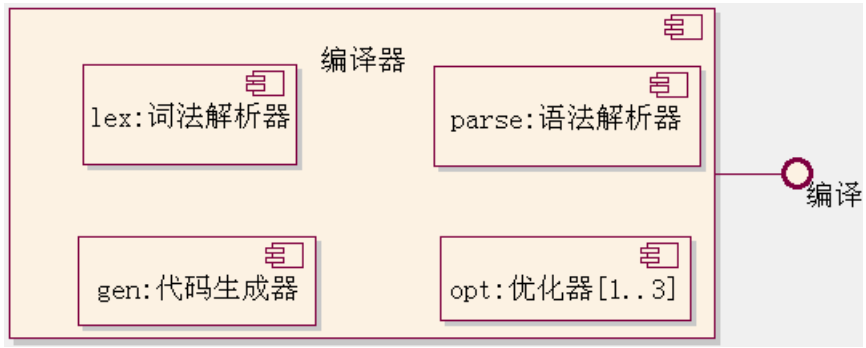
2、在构件图符的内部展示



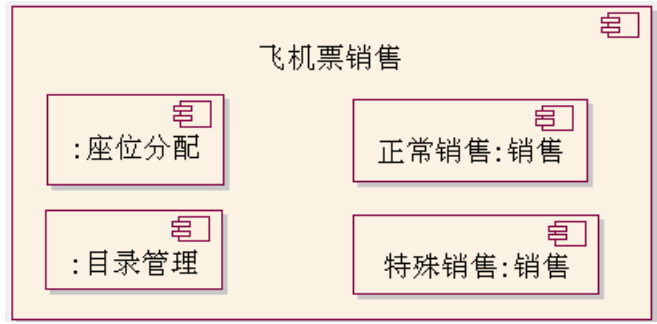
3、分栏展示

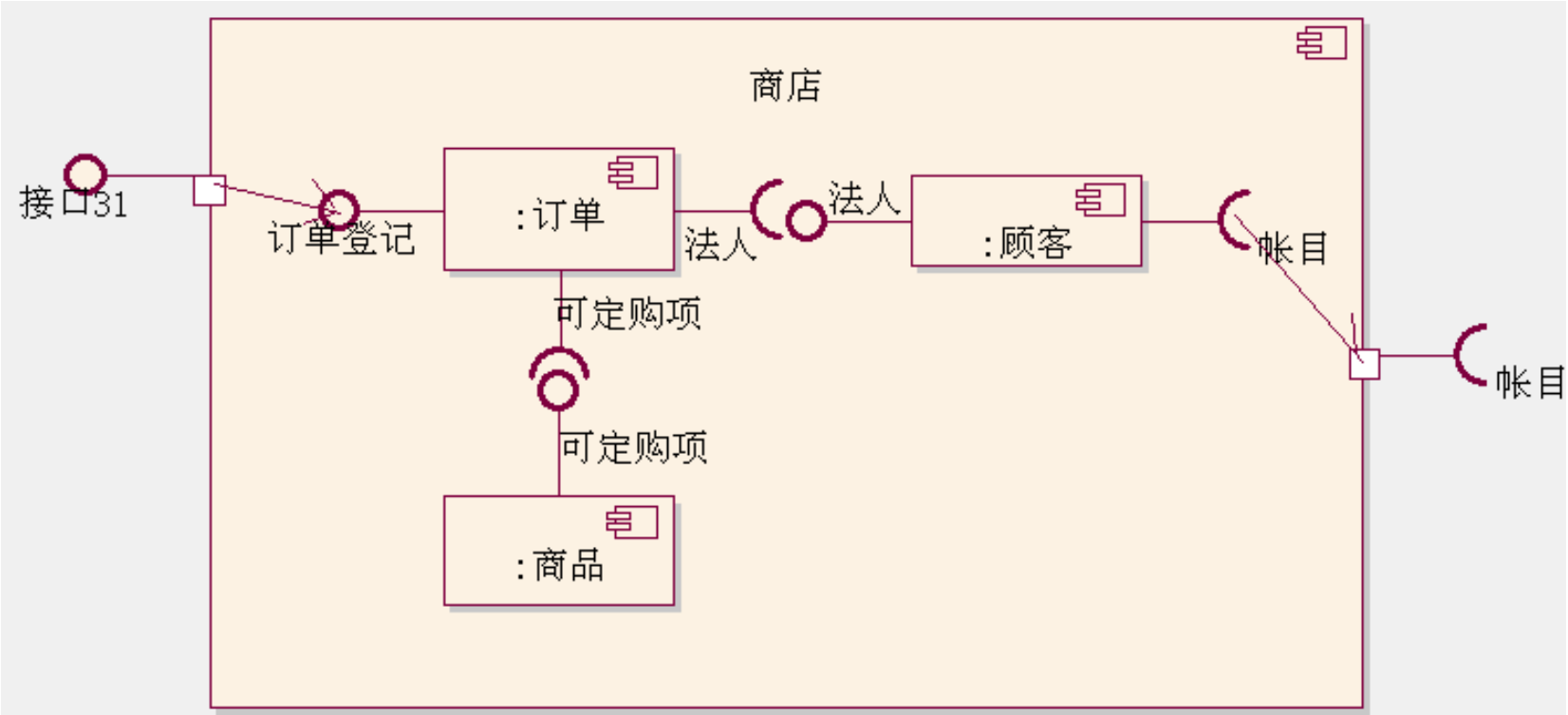
<<component>> 订单
<<provided interfaces>> 订单登记 开帐单 <<required interfaces>> 发票 建立(…) 登记费(…)
<<realization>> 订单头 排列项
<<artifact>> Order.jar

例题1：一个编译器的多个完整版本能通过不同级别的优化配置而成；在一个给定的版本中，在运行时可以选择一个适当的优化器。



例题2：对同类型的部件的处理



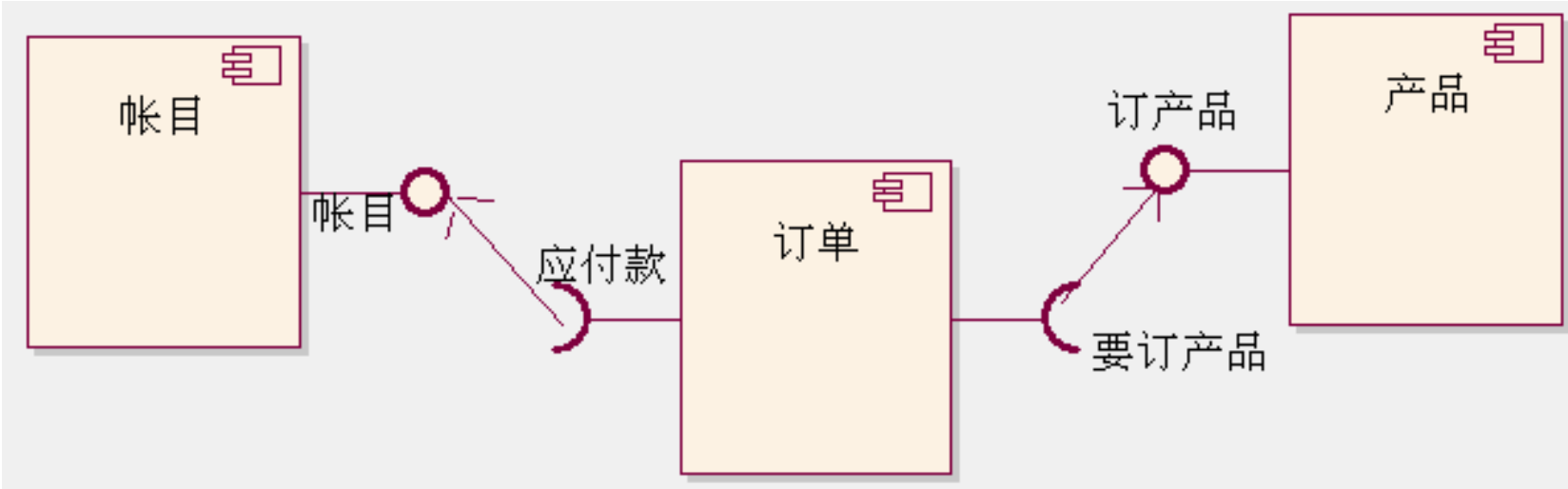


例题：展示构件的内部结构（或白盒视图），也展示了如何在构件的内部实现外部的行为 。

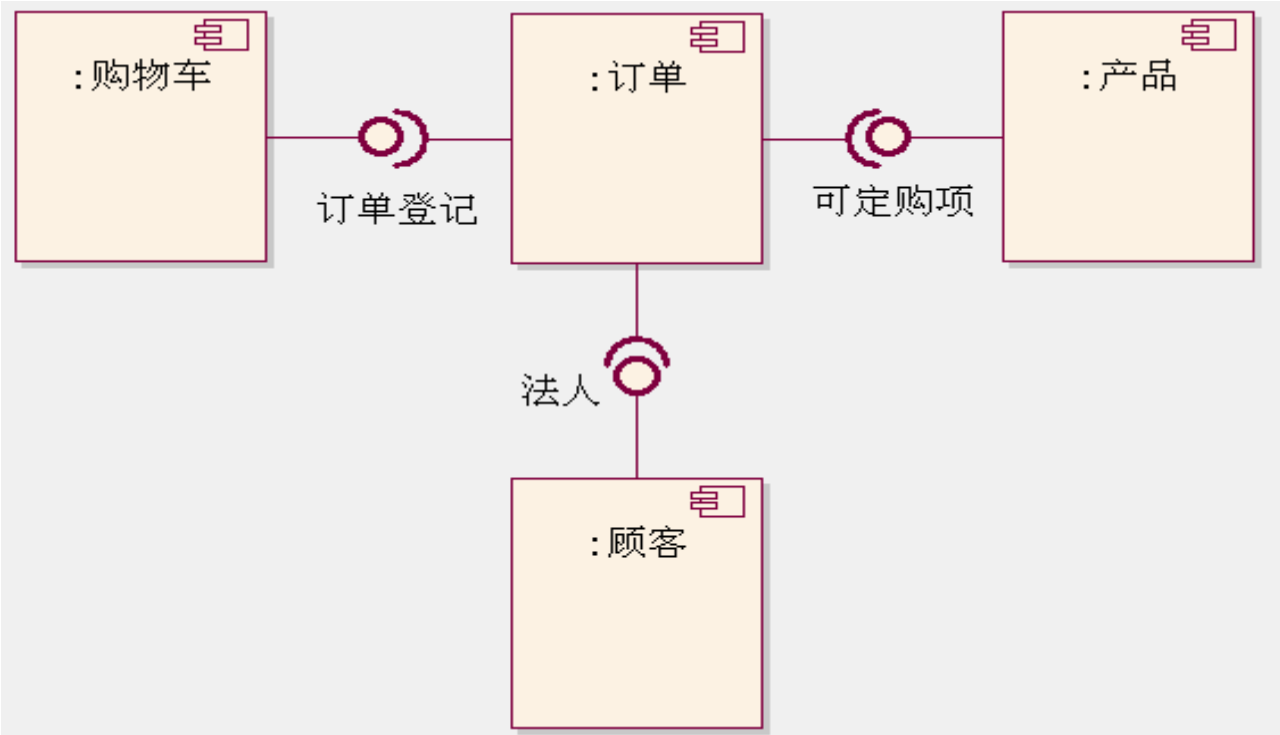
构件内部的部件可以为类或构件。

1．2 构件间的关系

（1）关联



构件、提供接口和请求接口以及通过依赖的线接示例



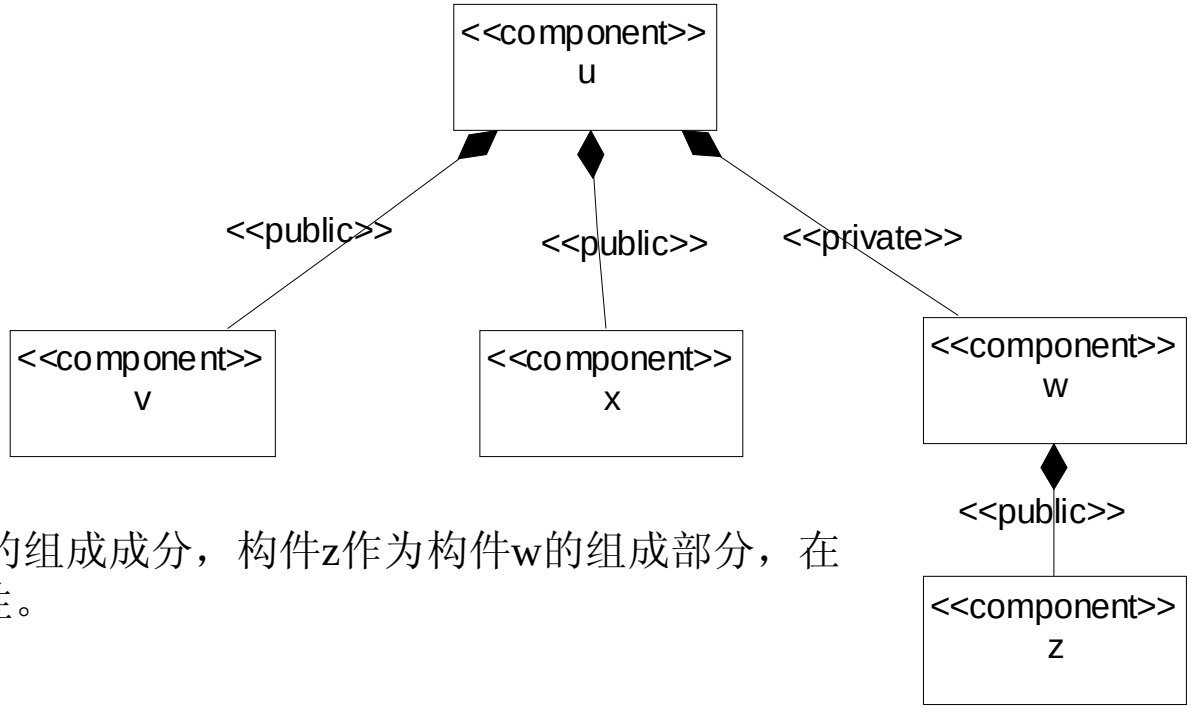
在组合结构图中，通过定义内部构件和连接件，可以在角色或实例级详细描述衔接。装配连接件是关联的实例。

一旦构件之间建立的关联，在构件运行时，一个构件的实例就可以实例化另一个构件，或访问另一个构件的实例。

一个构件实例可创建几个其它构件的实例，可能进而访问，也可能供其它实例访问。

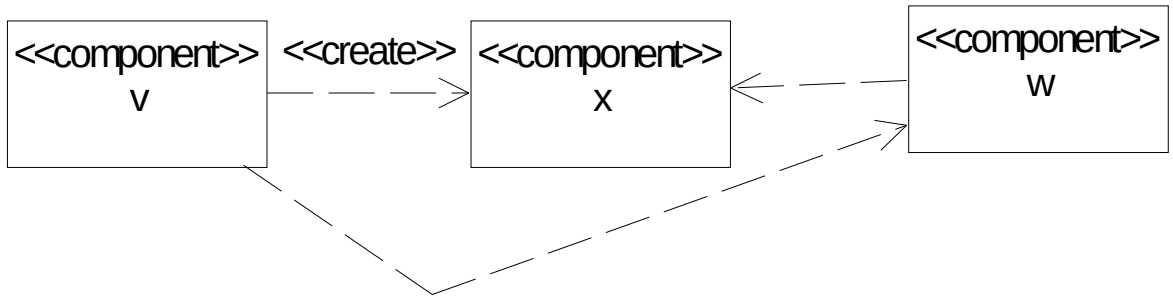
一个构件实例不创建另一个构件的实例，而通过获取的标识的方式再进行访问，但要遵从构件间的可见性。

（2）聚合



构件v、x和w作为构件u的组成成分，构件z作为构件w的组成部分，在组成关系上还标有对外可见性。

（3） 依赖



构件间的访问关系实际上就是构件接口间的依赖（通常在结构图上），在结构上定义构件间的线接。但这种关系是一种概貌性的关系，一般是中间结果的产物，最终要精确到关键间的关联或聚合。

（4） 继承

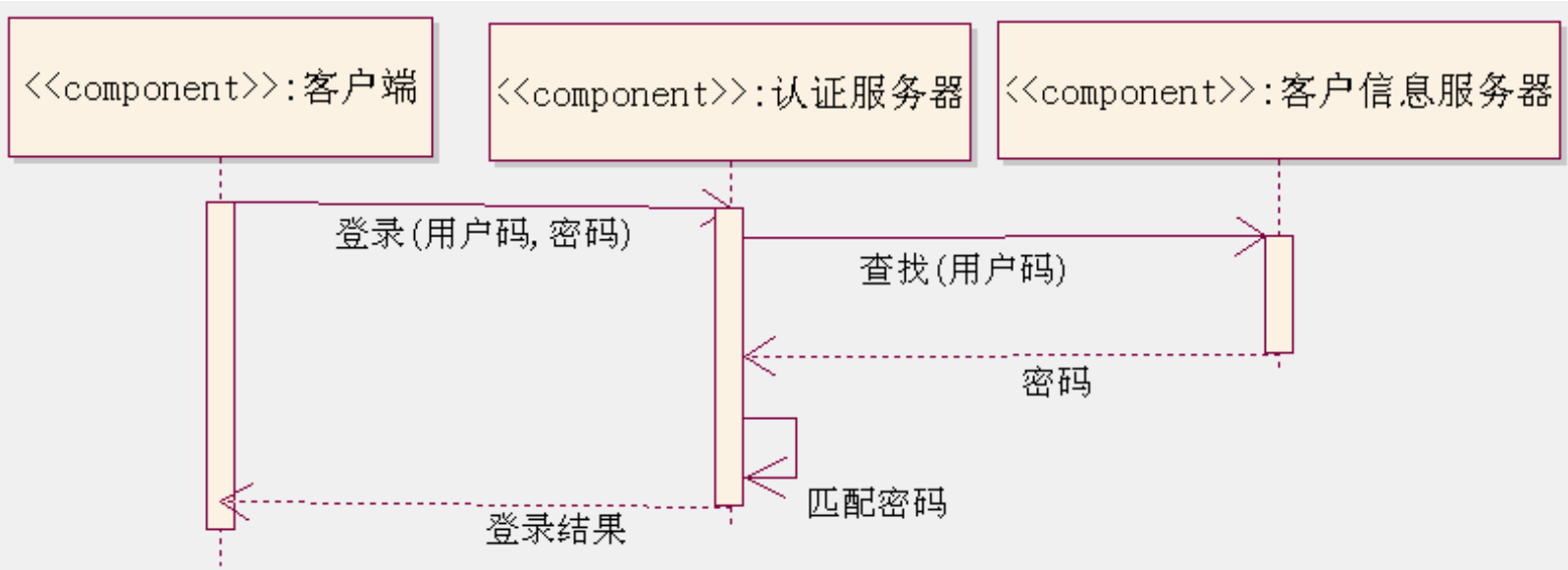
构件之间可以有泛化关系，但只是应用在规约阶段。

特殊构件继承一般构件的规约，并对所继承的规约可进行修改，但不能添加不变式。为了保证特化在逻辑上与泛化相符合，新操作的前置条件必须等于或弱于旧操作的前置条件，且新操作的后置条件必须等于或强于旧操作的后置条件。也即，特化构件的行为模型必须在协议和结构上与泛化构件相符合。在状态机上，特化的要保留泛化的东西。

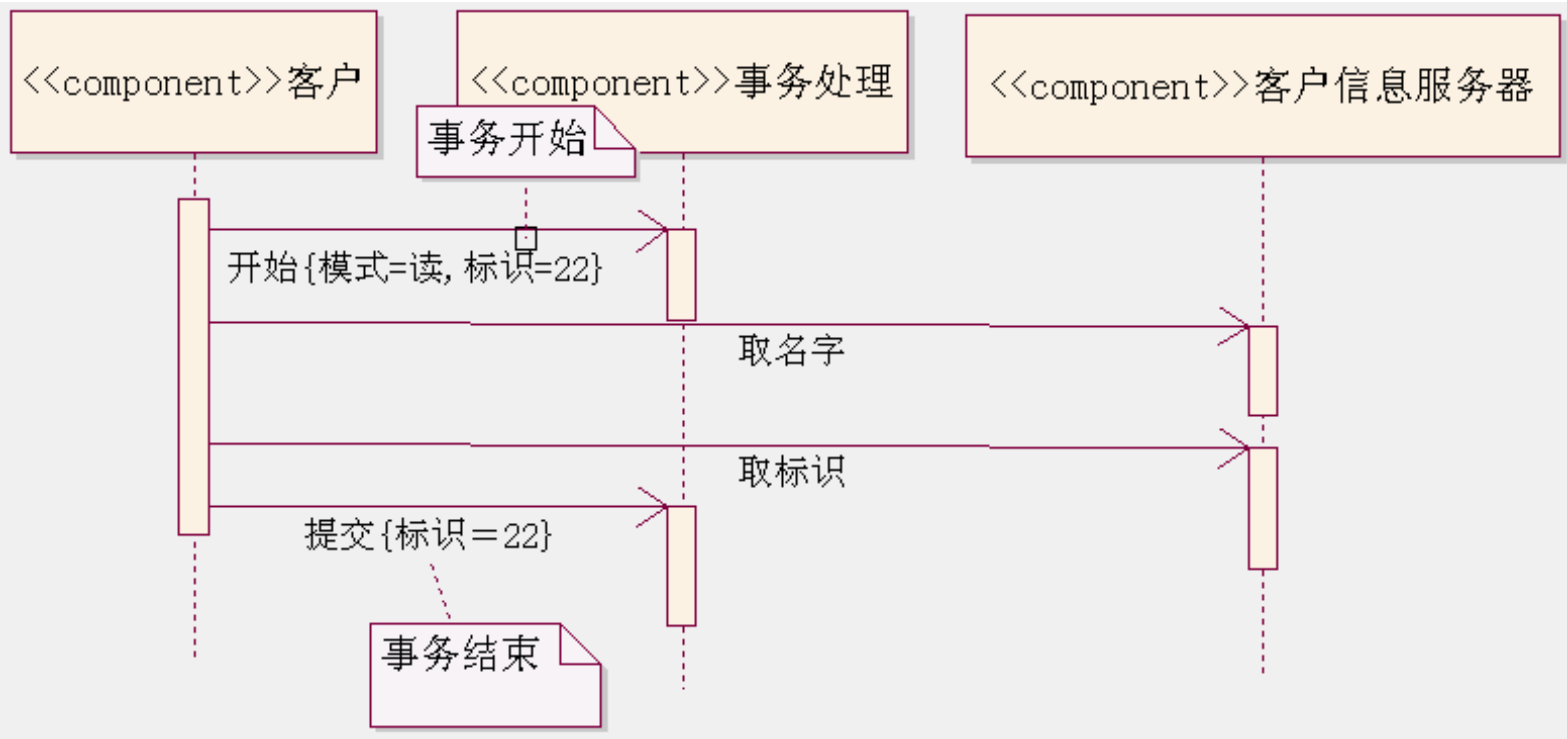
1.3 对构件的动态行为的描述

- 可以在接口、端口上和构件本身上附属诸如协议状态机这样的行为；
- 可以通过按调用操作的顺序显式地给出动态约束，以较精确地定义外部行为；
- 可以把其它的行为（如用况、活动或交互规约）与接口或连接件相关联，以定义协作间的“契约”。

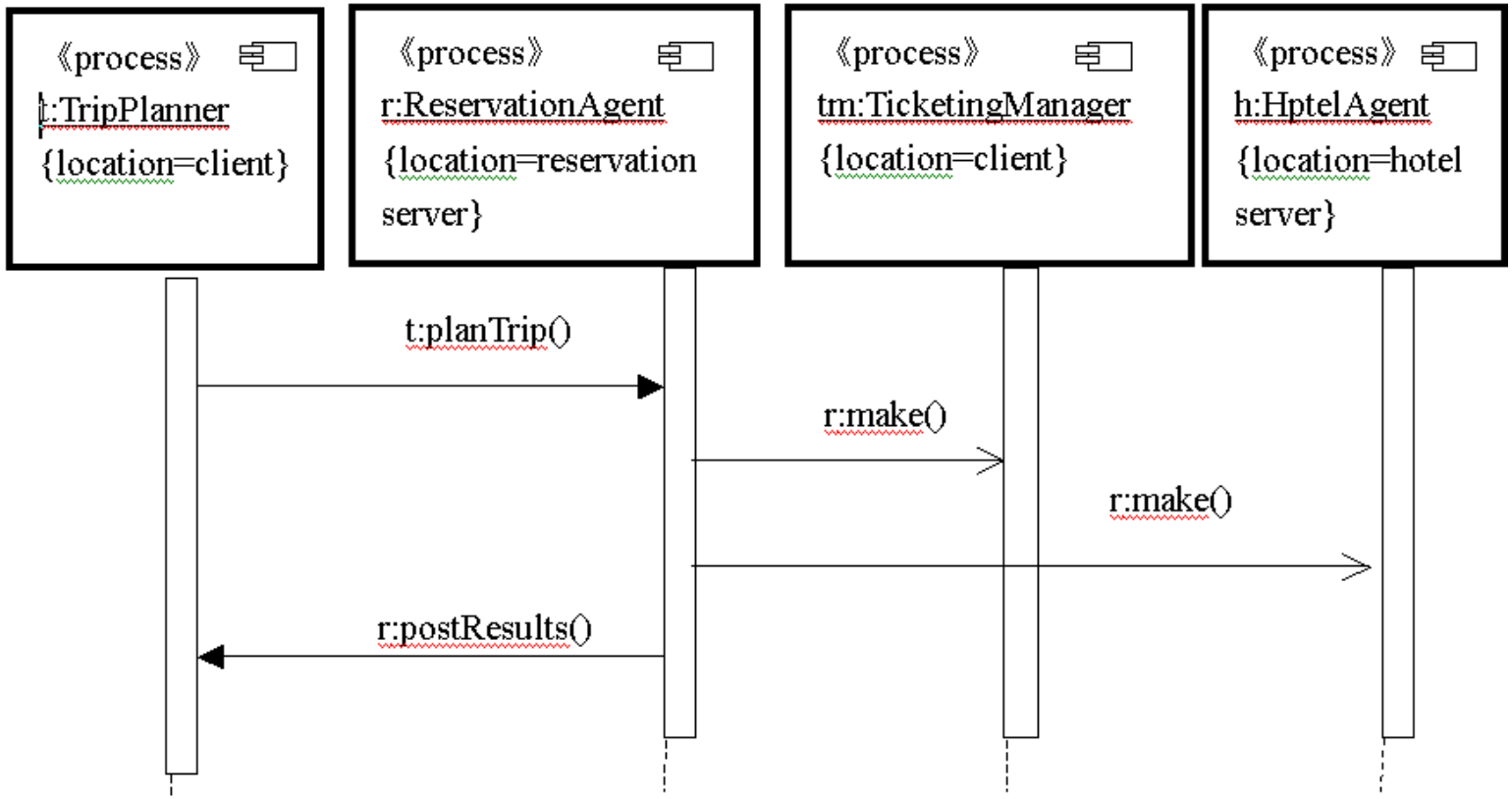
1.3.1用交互图描述构件间的交互行为



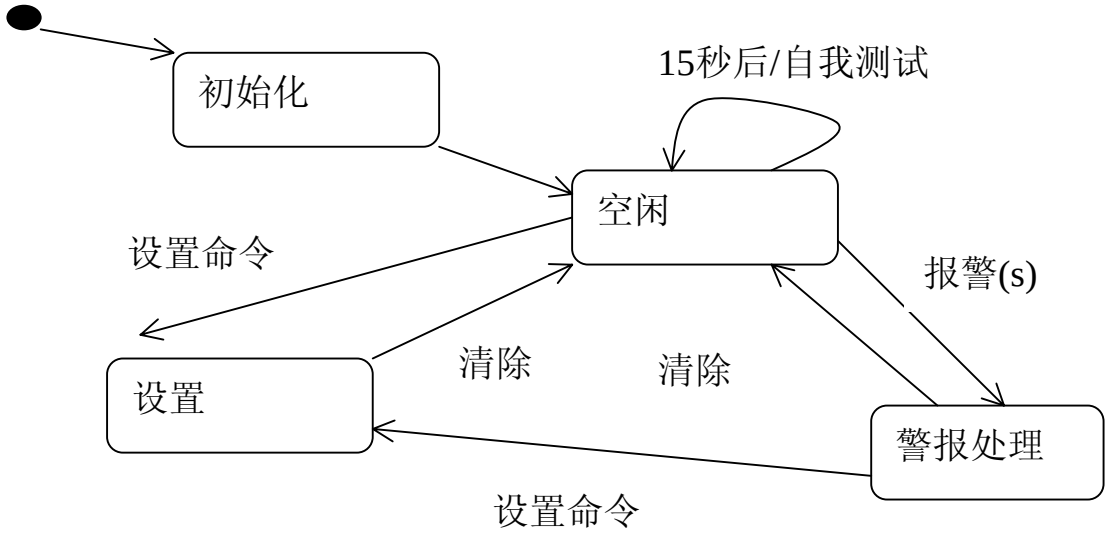
用构件间的交互图来对事务建模



若要对体系结构的进程情况建模，可使用顺序图。下面对一个分布式的旅游计划系统的体系结构的进程间的交互建模。



1.3.2用状态图描述构件的状态行为



2 制品图

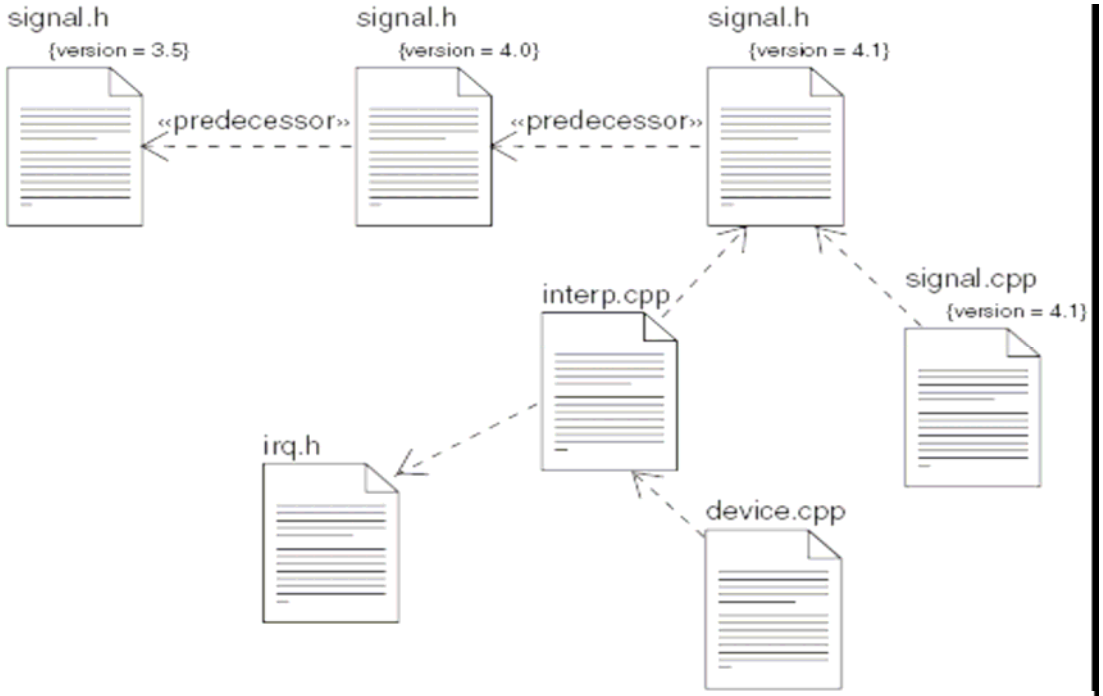
用于对构件的物理实现进行建模

1)对构件的源代码建模

■识别出一组相关源代码文件集合。

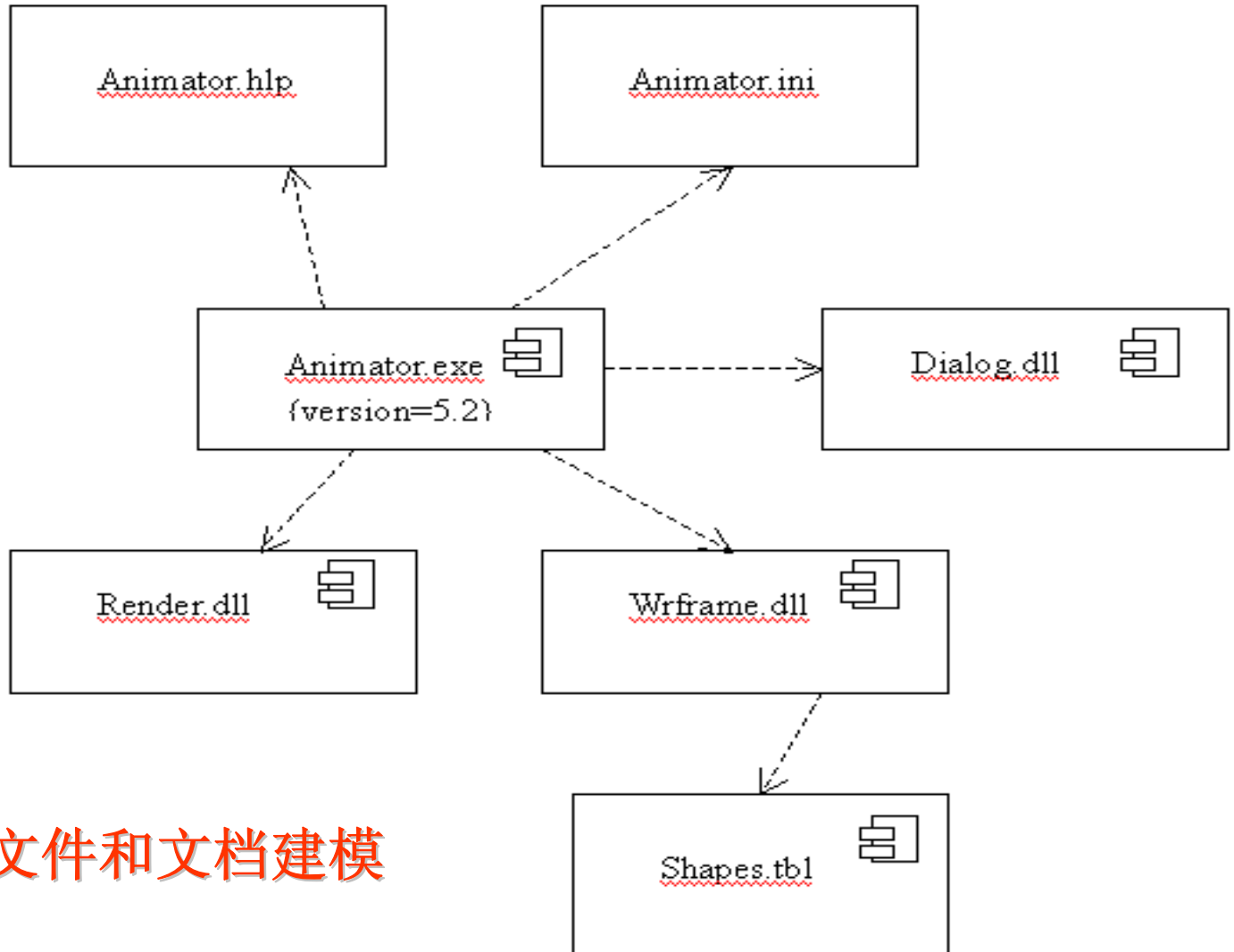
■考虑设置一个标记值，用它给出源代码文件的版本号、作者名或最后修改日期等信息。

■用依赖关系对这些文件之间的编译依赖关系建模。



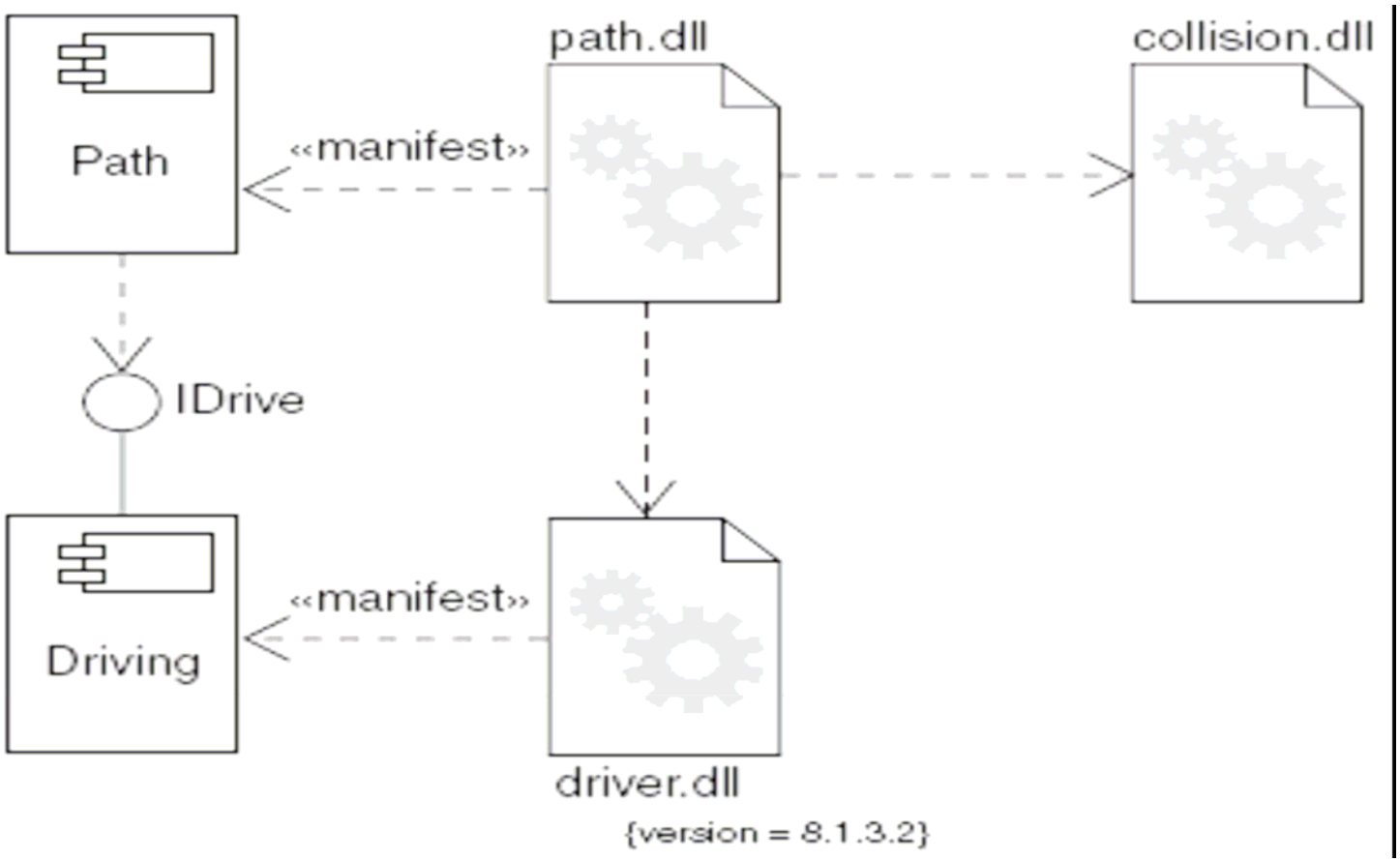
2)对可执行体的发布建模

- 识别要建模的制品集合。通常，它应包括一个结点上的部分或全部制品，或者集跨越系统中所有结点的这样制品集的分布；
- 考虑该集合中各制品的衍型；
- 考虑制品之间的关系。



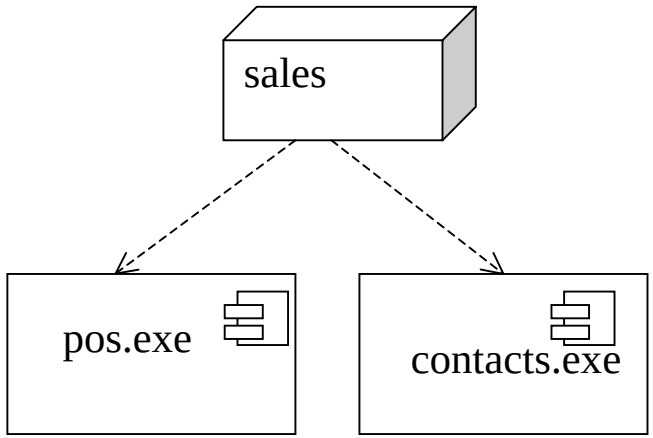
对表、文件和文档建模

利用承载依赖对构件和实现体建模



3 部署图

一个节点上可以有一个或多个构件，一个构件也可以部署在一个或多个节点上。如左图所示，节点与它所部署的构件之间的关系可用依赖关系显式地加以表示。



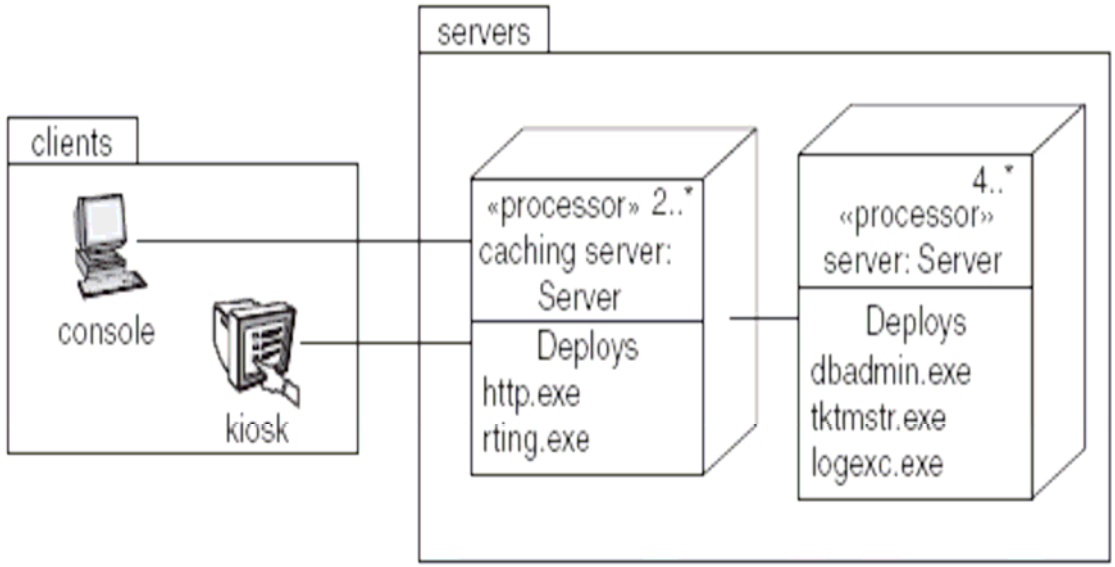
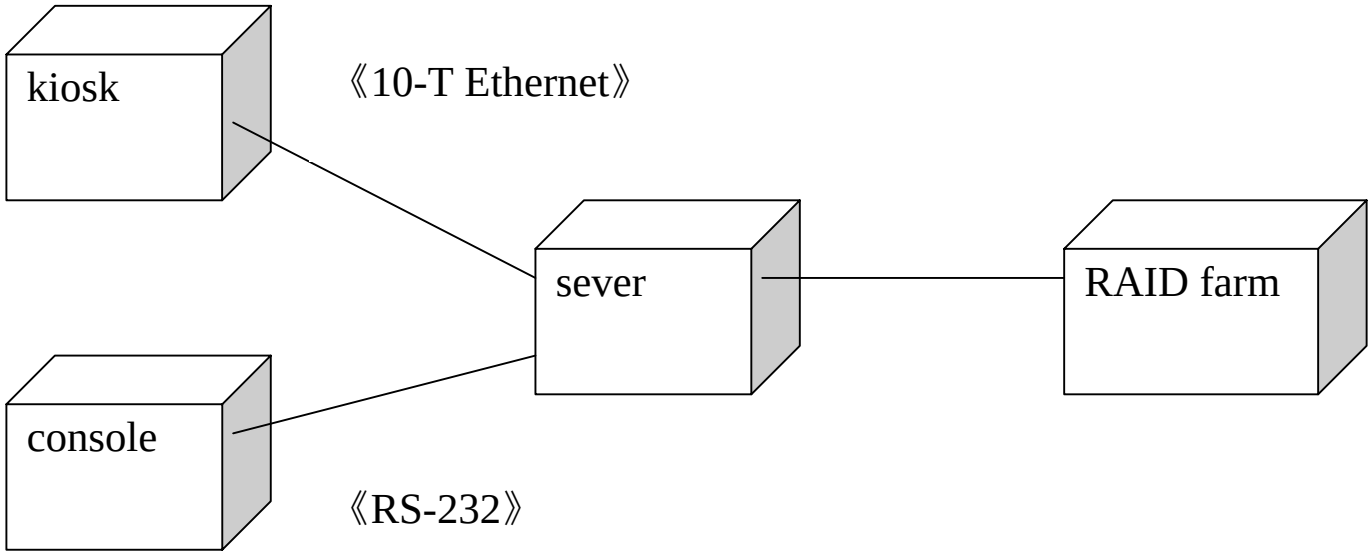
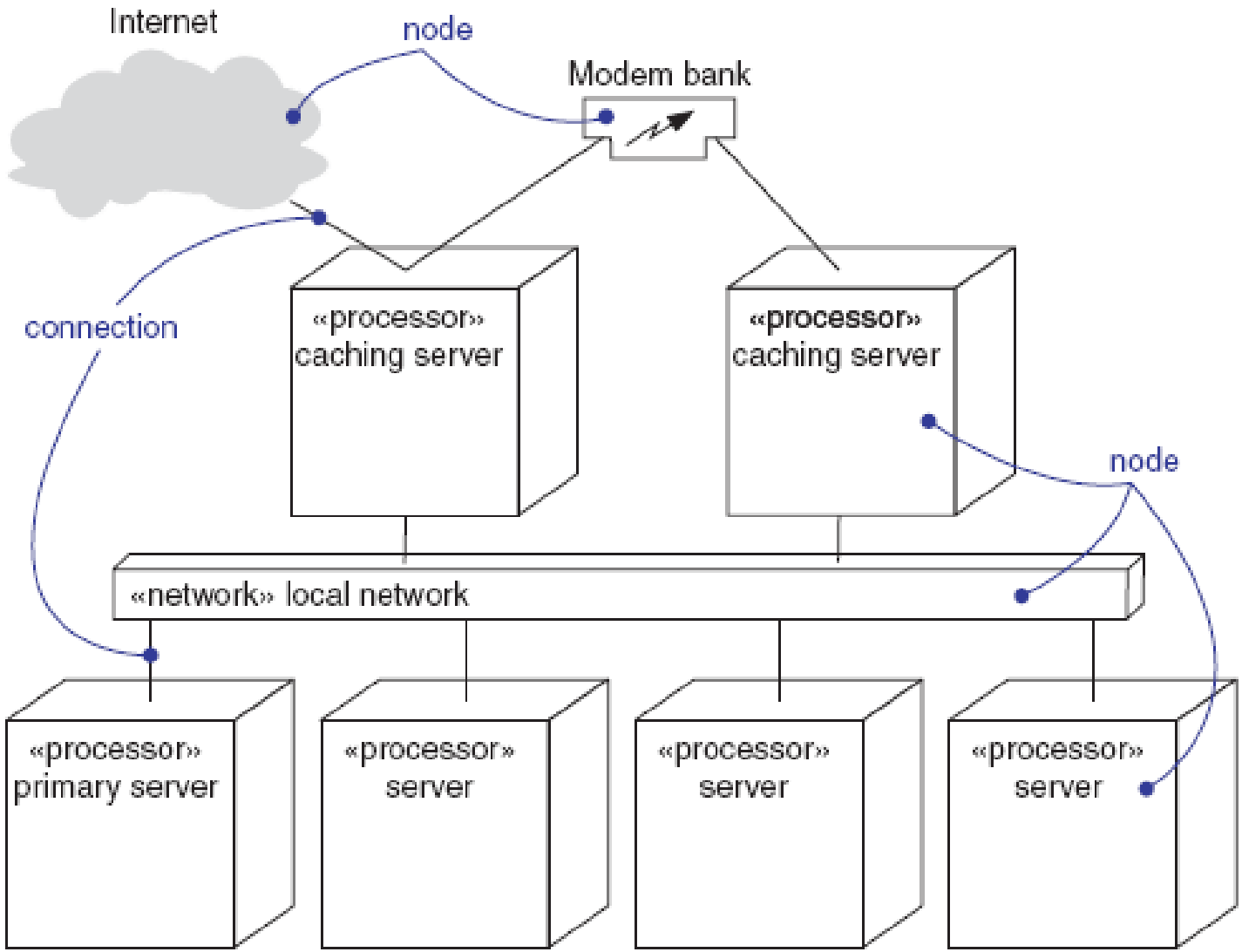


Figure 31-3: Modeling a Client/Server System

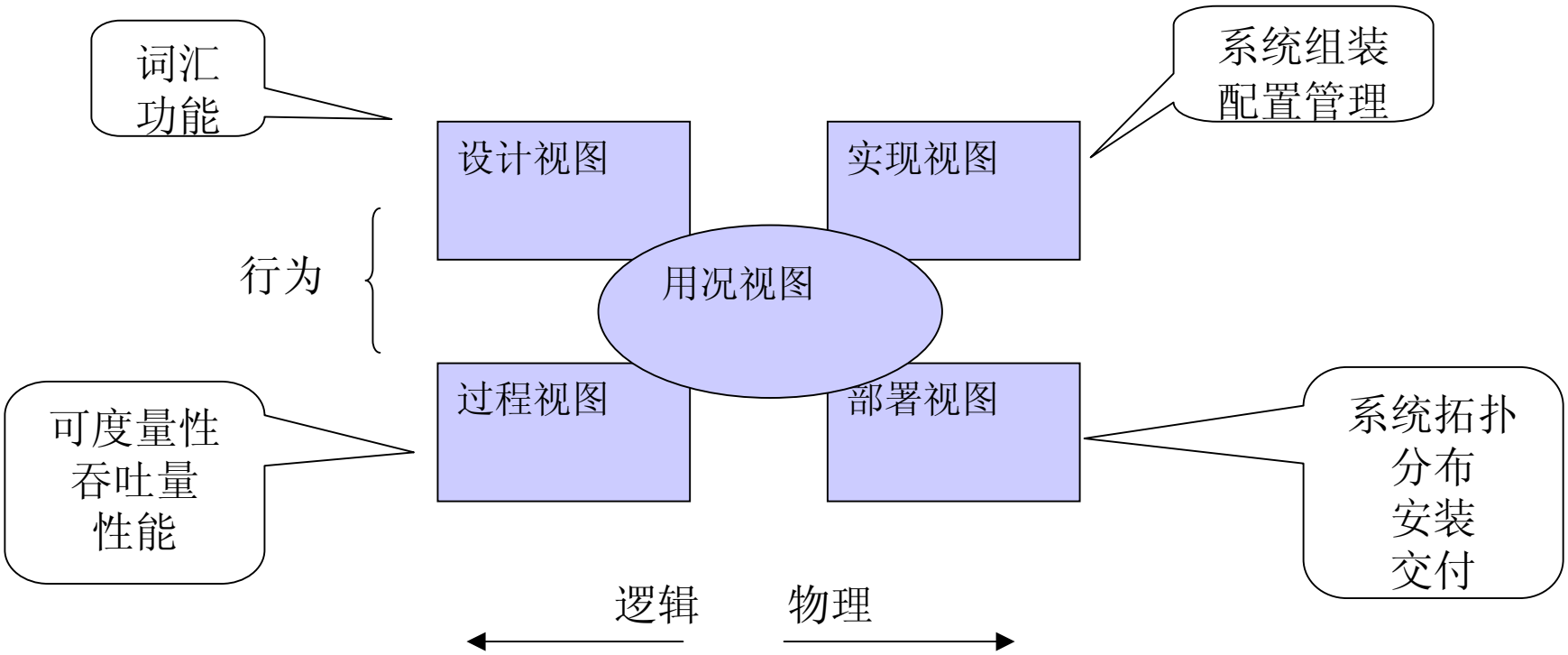
节点之间最常见的关系是关联关系，用来表示节点之间的物理连接。下图节点连接使用了以太网连接协议和串口连接协议。也可以利用关联关系表示节点间的间接连接，例如远程服务器之间的卫星通讯连接。



对单机式、嵌入式和分布式系统拓扑结构中的处理器和设备，都可以用部署图进行建模。

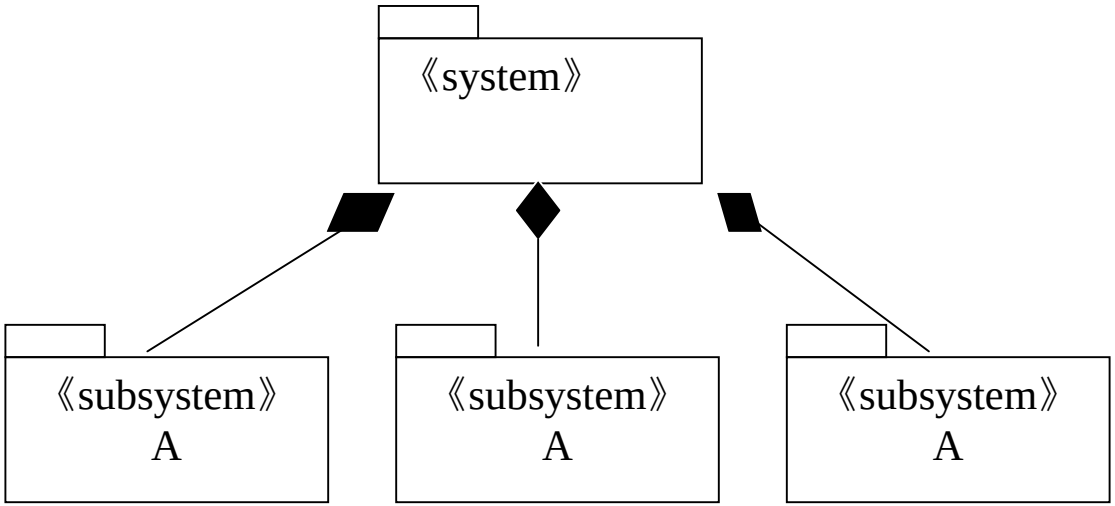

Internet
node
Modem bank
connection
«processor»
caching server
«processor»
caching server
node
«network» local network
«processor»
primary server
«processor»
server
«processor»
server
«processor»
server

4 对系统的体系结构建模

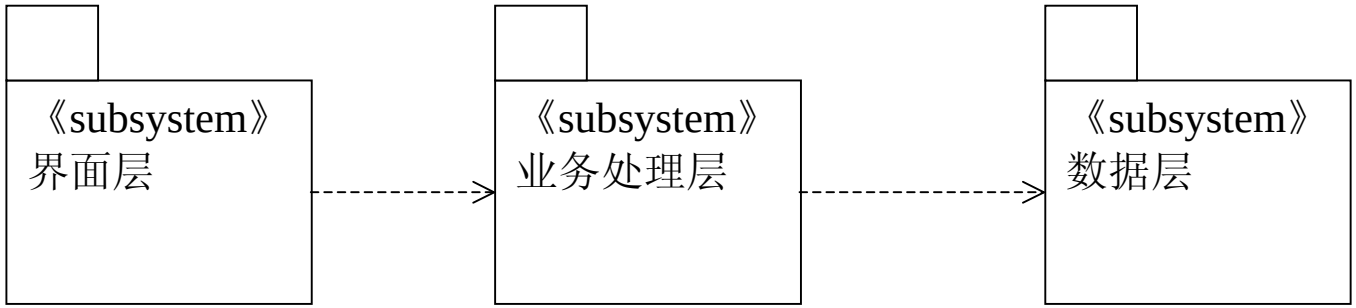


系统是由一组为了完成一定的目标而组织起来的元素构成的。如果一个系统较为复杂，可把它分解为一组子系统。子系统只是系统的一部分，它被用来将一个复杂的系统分解为几乎相互独立的部分，与系统的其它部分（如子系统和构件）有接口，并有自己应用的环境。

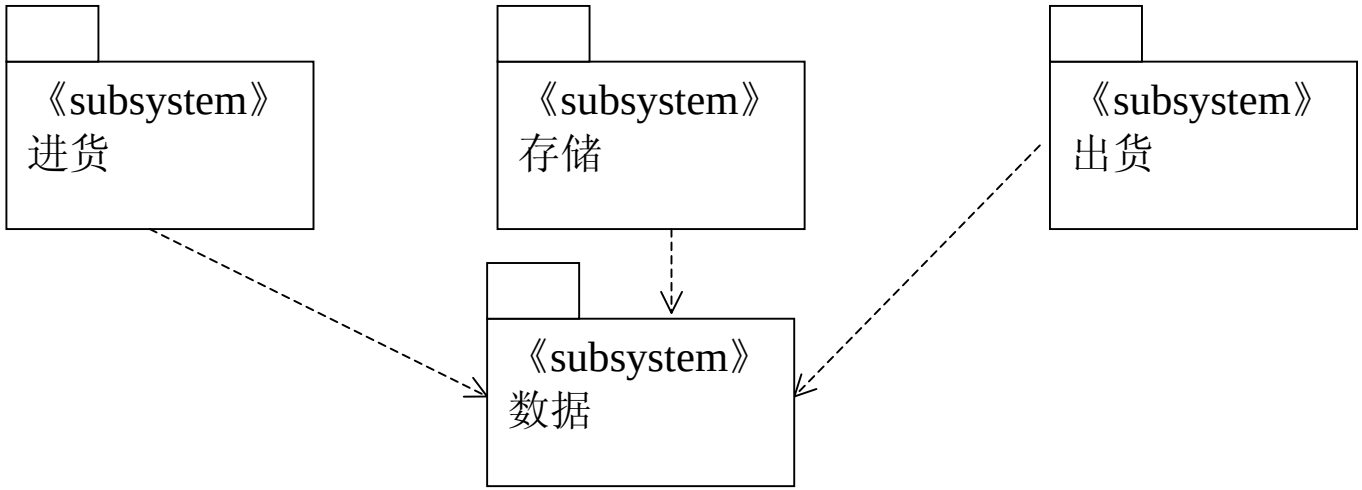
1、对设计级的系统与子系统或子系统间的关系建模



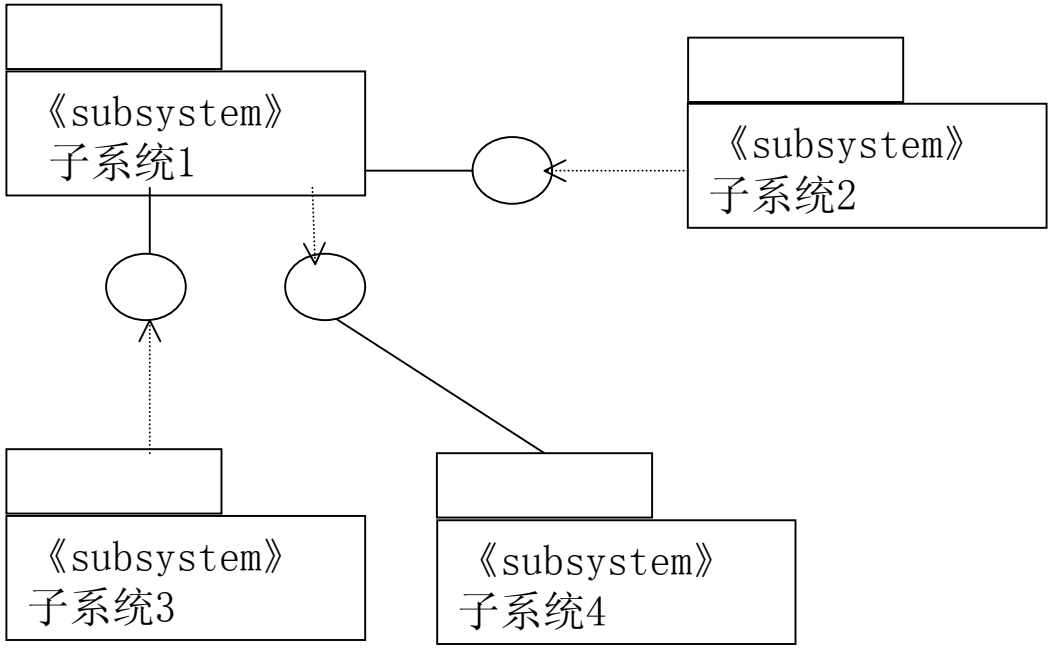
对三层体系结构模式建模



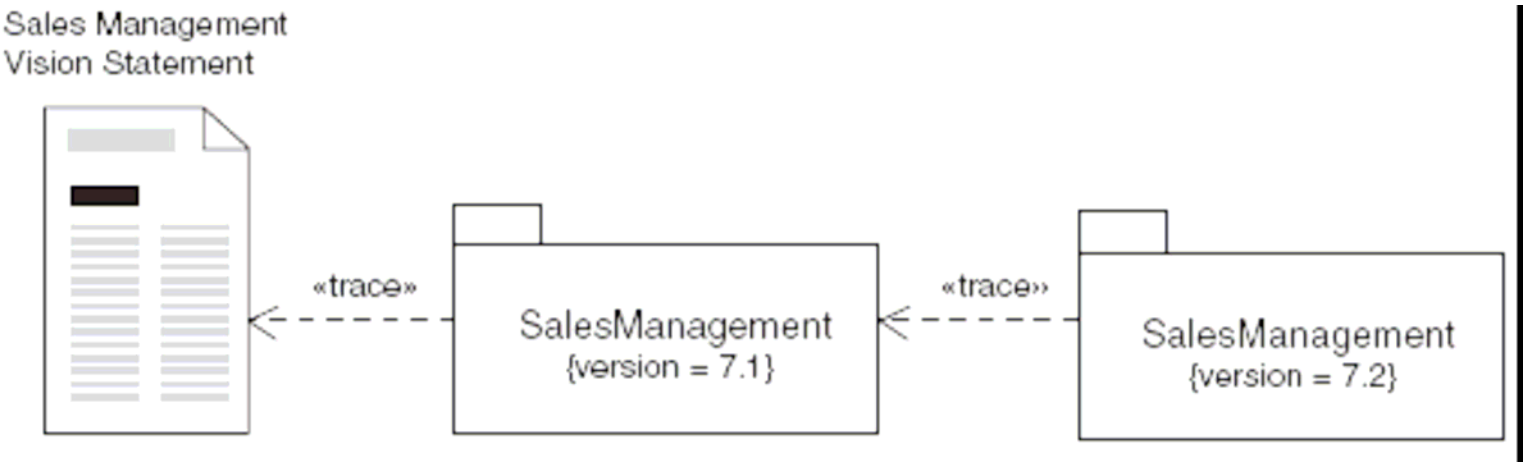
对管道过滤器型体系结构模式建模



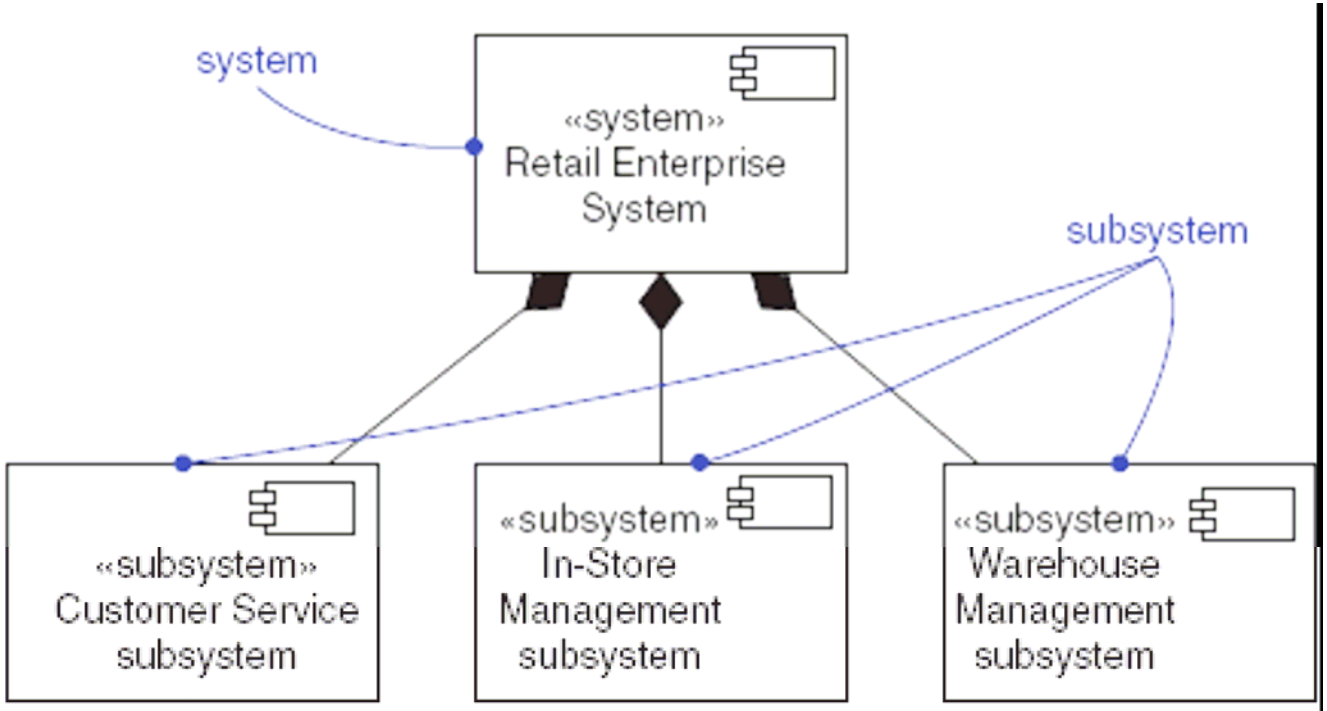
同层子系统之间通过接口相互联系

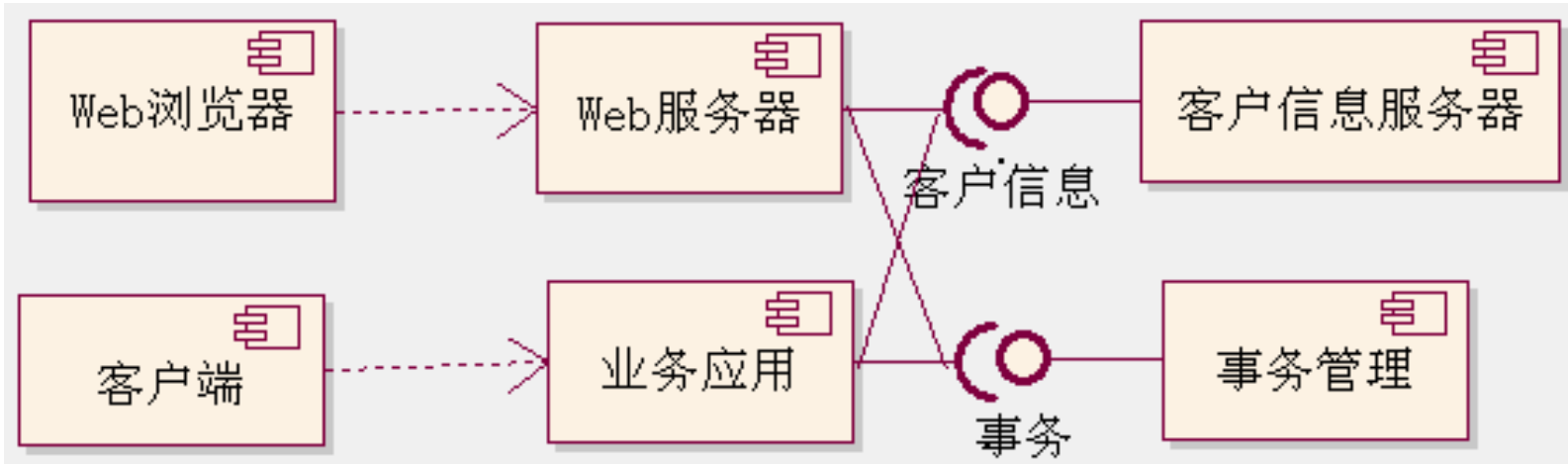


系统间的追踪关系



2、对实现后的系统与子系统或子系统间的关系建模

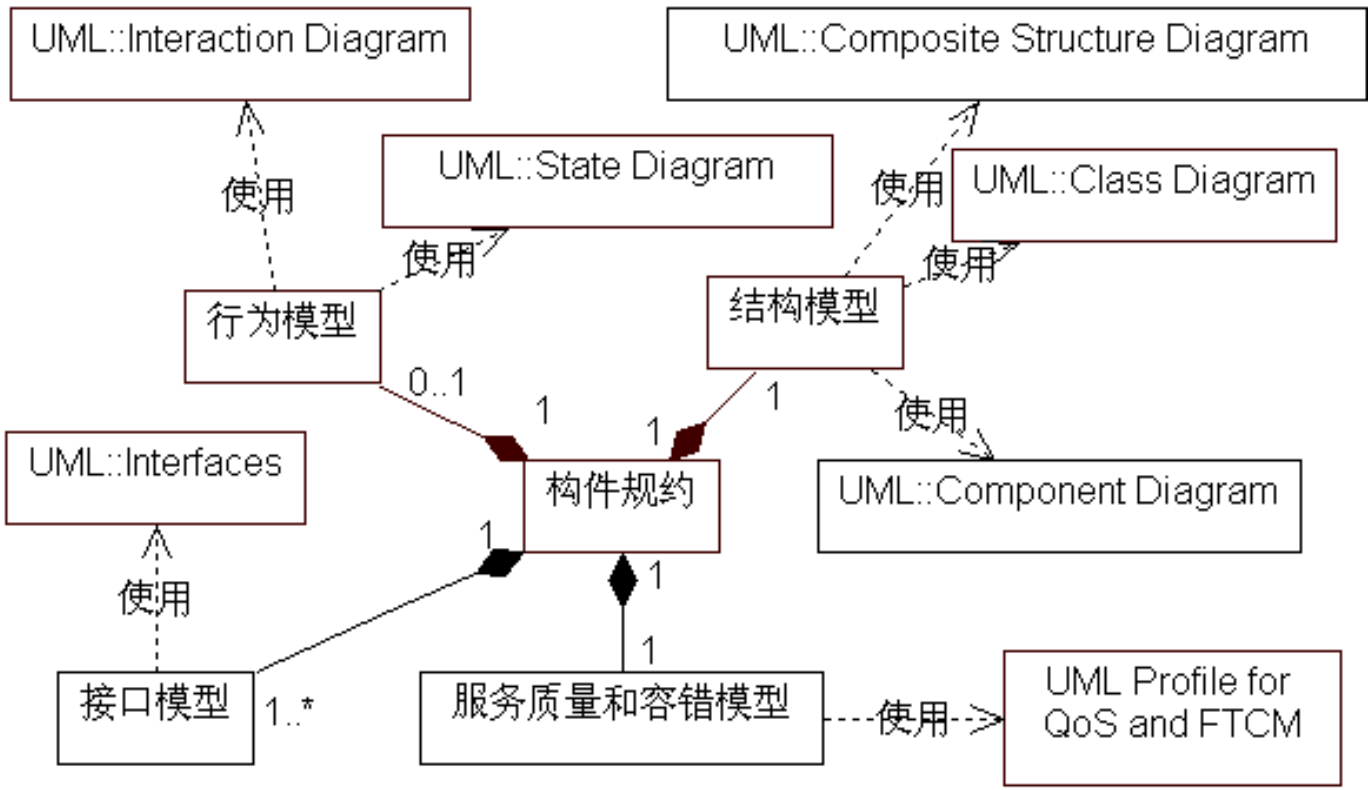




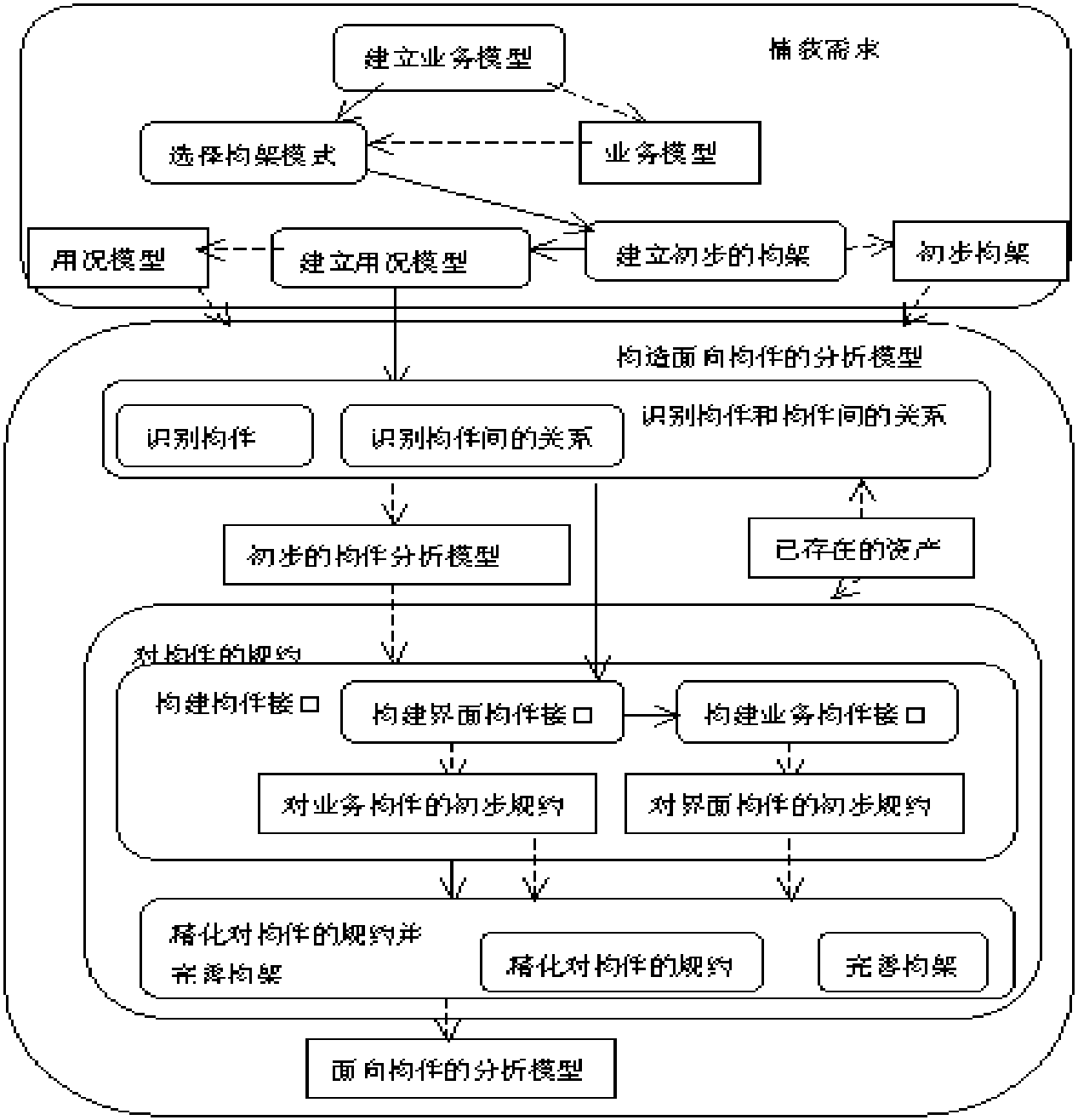
第二部分 一种建立面向构件的系统分析模型的方法

2.1 构件元模型以及构件规约

- 构件元模型采用UML 2.0
- 构件规约



2.2 建立面向构件的系统分析模型过程指南



要构造出最终的面向构件的系统分析模型，往往需要按上述过程进行多次迭代。

谢谢！