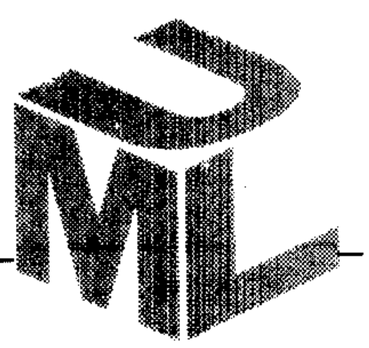
第3章 UML初览



本章使用一个简单的例子对UML中所使用的概念和视图进行初览。本章的目的是要将高层UML概念组织成一系列较小的视图和图表来可视化说明这些概念，说明如何用各种不同的概念来描述一个系统以及如何将各种视图组织在一起。概括性的说明不可能面面俱到，其中省略了许多概念。要想得到更详细的说明，可参见下一章对UML各视图的说明和本书大全部分的有关细节。

本章使用的例子是计算机管理的戏院售票系统。这是一个精心设计的例子，目的是用少量篇幅来强调说明UML的各个组件。这是一个经过有意简化的例子，忽略了有关细节。除非进行大量的反复说明，否则一个实际系统的完整模型不可能用这么少的篇幅来对UML中使用的每种组件进行介绍。

3.1 UML视图

UML中的各种组件和概念之间没有明显的划分界限，但为方便起见，我们用视图来划分这些概念和组件。视图只是表达系统某一方面特征的UML建模组件的子集。视图的划分带有一定的随意性，但我们希望这种看法仅仅是直觉上的。在每一类视图中使用一种或两种特定的图来可视化地表示视图中的各种概念。

在最上一层，视图被划分成三个视图域：结构分类、动态行为和模型管理。

结构分类描述了系统中的结构成员及其相互关系。类元包括类、用例、构件和节点。类元为研究系统动态行为奠定了基础。类元视图包括静态视图、用例视图和实现视图。

动态行为描述了系统随时间变化的行为。行为用从静态视图中抽取的系统的瞬间值的变化来描述。动态行为视图包括状态机视图、活动视图和交互视图。

模型管理说明了模型的分层组织结构。包是模型的基本组织单元。特殊的包还包括模型和子系统。模型管理视图跨越了其他视图并根据系统开发和配置组织这些视图。

UML还包括多种具有扩展能力的组件，这些扩展能力有限但很有用。这些组件包括约束、构造型和标记值，它们适用于所有的视图元素。

表3-1列出了UML的视图和视图所包括的图以及与每种图有关的主要概念。不能把这张表看成是一套死板的规则，应将其视为对UML常规使用方法的指导，因为UML允许使用混合视图。

表3-1　UML视图和图

主要的域	视　图	图	主要概念
结构	静态视图	类图	类、关联、泛化、依赖关系、实现、接口
	用例视图	用例图	用例、参与者、关联、扩展、包括、用例泛化
	实现视图	构件图	构件、接口、依赖关系、实现
	部署视图	部署图	节点、构件、依赖关系、位置
动态	状态机视图	状态机图	状态、事件、转换、动作
	活动视图	活动图	状态、活动、完成转换、分叉、结合
	交互视图	顺序图	交互、对象、消息、激活
		协作图	协作、交互、协作角色、消息
模型管理	模型管理视图	类图	包、子系统、模型
可扩展性	所有	所有	约束、构造型、标记值

3.2　静态视图

静态视图对应用领域中的概念以及与系统实现有关的内部概念建模。这种视图之所以被称之为是静态的是因为它不描述与时间有关的系统行为，此种行为在其他视图中进行描述。静态视图主要是由类及类间相互关系构成，这些相互关系包括：关联、泛化和各种依赖关系，如使用和实现关系。类是应用领域或应用解决方案中概念的描述。类图是以类为中心来组织的，类图中的其他元素或属于某个类或与类相关联。静态视图用类图来实现，正因为它以类为中心，所以称其为类图。

在类图中类用矩形框来表示，它的属性和操作分别列在分格中。如不需要表达详细信息时，分格可以省略。一个类可能出现在好几个图中。同一个类的属性和操作可只在一种图中列出，在其他图中可省略。

关系用类框之间的连线来表示，不同的关系用连线上和连线端头处的修饰符来区别。

图3-1是售票系统的类图，它只是售票系统领域模型的一部分。图中表示了几个重要的类，如Customer、Reservation、Ticket和Performance。一个顾客可多次订票，但每一次订票只能由一个顾客来执行。有两种订票方式：个人票或套票，前者只是一张票，后者包括多张票。每一张票不是个人票就是套票中的一张，但是不能又是个人票又是套票中的一张。每场演出都有多张票可供预定，每张票对应一个唯一的座位号。每次演出用剧目名、日期和时间来标识。

类可用不同的精确度和抽象级别来描述。在设计的初期阶段，所建立的模型只是问题的逻辑模型，到了设计的后期，模型中会增添许多设计结论和有关系统实现的细节。UML中的大部分视图都可不断进行类似的细化。

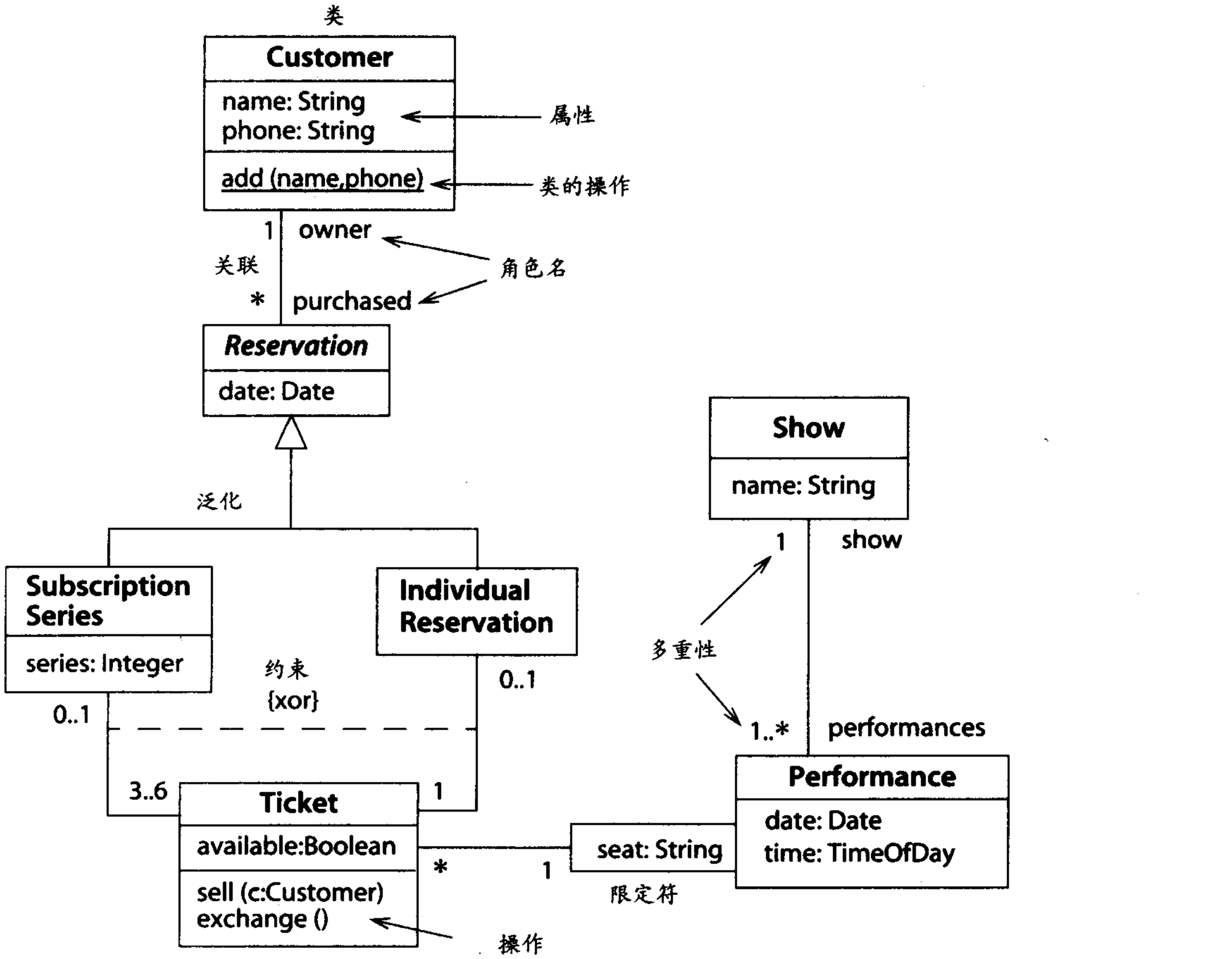


图3-1 类图

3.3 用例视图

用例视图是被称为参与者的外部用户所能观察到的系统功能的模型图。用例是系统中的一个功能单元，可以被描述为参与者与系统之间的一次交互作用。用例模型的用途是列出系统中的用例和参与者，并显示哪个参与者参与了哪个用例的执行。

图3-2是售票系统的用例图。参与者包括售票员、监督员和公用电话亭。公用电话亭是另一个系统，它接受顾客的订票请求。在售票处的应用模型中，顾客不是参与者，因为顾客不直接与售票处打交道。用例包括通过公用电话亭或售票员购票，购预约票（只能通过售票员）以及售票监督（应监督员的要求）。购票和购预约票包括一个共同的部分——即通过信用卡来付钱。（对售票系统的完整描述还要包括其他一些用例，例如换票和验票等。）

用例也可以有不同的层次。用例可以用其他更简单的用例进行说明。在交互视图中，用

例做为交互图中的一次协作来实现。

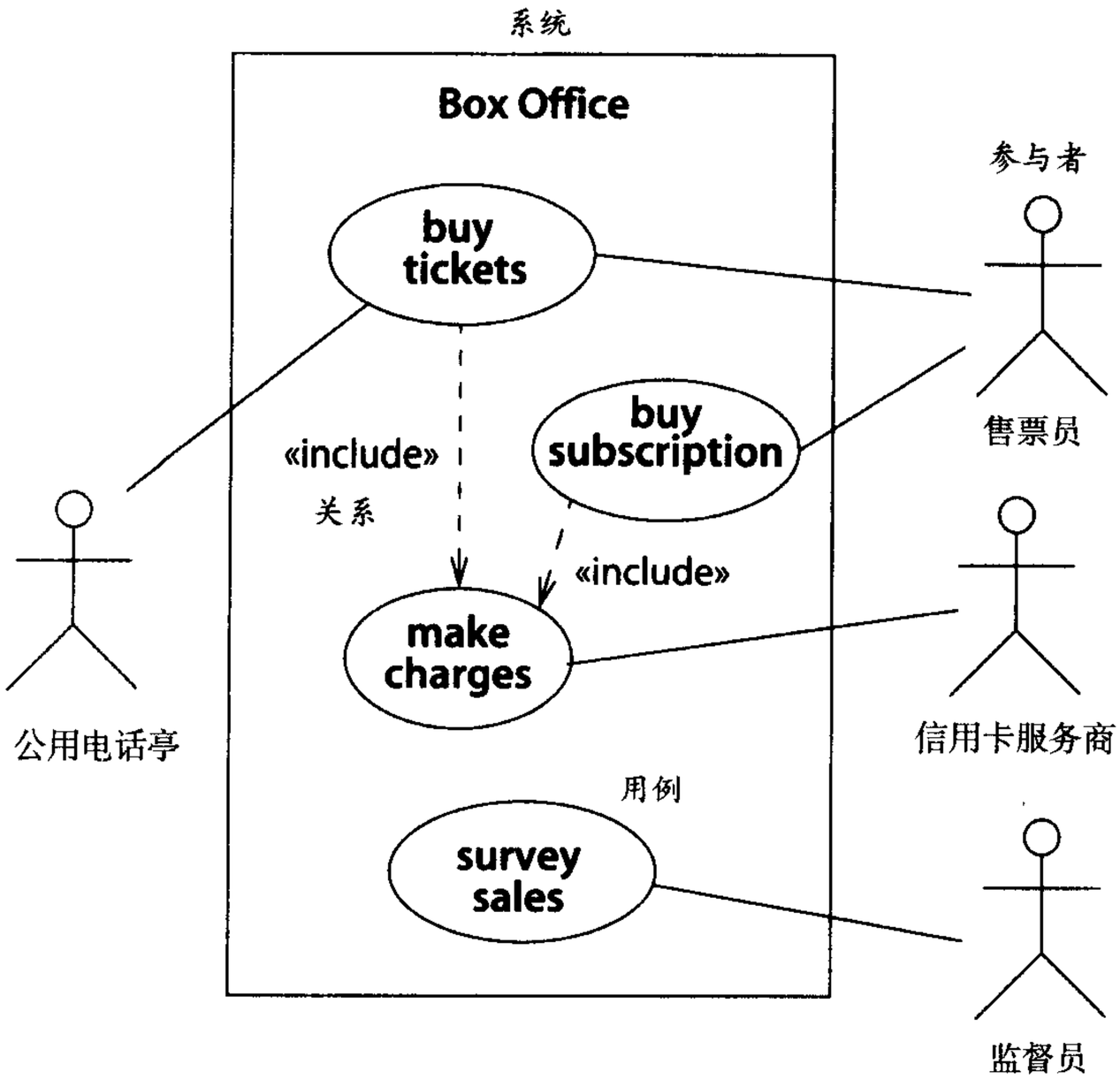


图3-2　用例图

3.4　交互视图

交互视图描述了执行系统功能的各个角色之间相互传递消息的顺序关系。类元是对在系统内交互关系中起特定作用的一个对象的描述，这使它区别于同类的其他对象。交互视图显示了跨越多个对象的系统控制流程。交互视图可用两种图来表示：顺序图和协作图，它们各有不同的侧重点。

3.4.1　顺序图

顺序图表示了对象之间传送消息的时间顺序。每一个类元角色用一条生命线来表示——即用垂直线代表整个交互过程中对象的生命期。生命线之间的箭头连线代表消息。顺序图可以用来进行一个场景说明——即一个事务的历史过程。

顺序图的一个用途是用来表示用例中的行为顺序。当执行一个用例行为时，顺序图中的每条消息对应了一个类操作或状态机中引起转换的触发事件。

图3-3是描述购票这个用例的顺序图。顾客在公用电话亭与售票处通话触发了这个用例的

执行。顺序图中付款这个用例包括售票处与公用电话亭和信用卡服务处的两个通信过程。这个顺序图用于系统开发初期，未包括完整的与用户之间的接口信息。例如，座位是怎样排列的；对各类座位的详细说明都还没有确定。尽管如此，交互过程中最基本的通信已经在这个用例的顺序图中表达出来了。

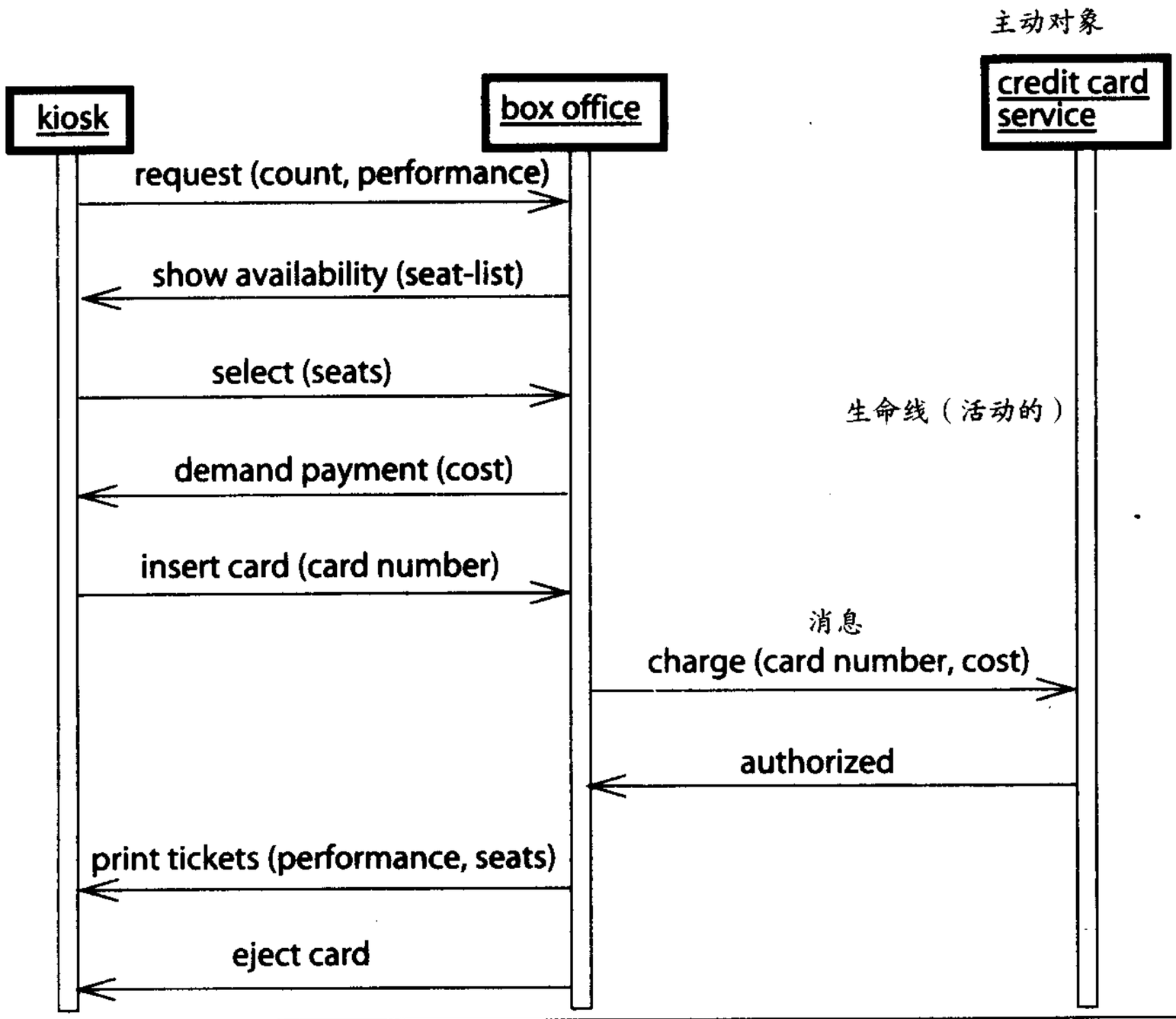


图3-3 顺序图

3.4.2 协作图

协作图对在一次交互中有意义的对象和对象间的链建模。对象和关系只有在交互的语境中才有意义。类元角色描述了一个对象，关联角色描述了协作关系中的一个链。协作图用几何排列来表示交互作用中的各角色（如图3-4）。附在类元角色上的箭头代表消息。消息的发生顺序用消息箭头处的编号来说明。

协作图的一个用途是表示一个类操作的实现。协作图可以说明类操作中用到的参数和局部变量以及操作中的永久链。当实现一个行为时，消息编号对应了程序中嵌套调用结构和信号传递过程。

图3-4是开发过程后期订票交互的协作图。这个图表示了订票涉及的各个对象间的交互关

系。请求从公用电话亭发出，要求从所有的演出中查找某次演出的资料。返回给ticketseller对象的指针db代表了与某次演出资料的局部暂时链接，这个链接在交互过程中保持，交互结束时丢弃。售票方准备了许多演出的票；顾客在各种价位做一次选择，锁定所选座位，售票员将顾客的选择返回给公用电话亭。当顾客在座位表中做出选择后，所选座位被声明，其余座位解锁。

顺序图和协作图都可以表示各对象间的交互关系，但它们的侧重点不同。顺序图用消息的几何排列关系来表达消息的时间顺序，各角色之间的相关关系是隐含的。协作图用各个角色的几何排列图形来表示角色之间的关系，并用消息来说明这些关系。在实际中可以根据需要选用这两种图。

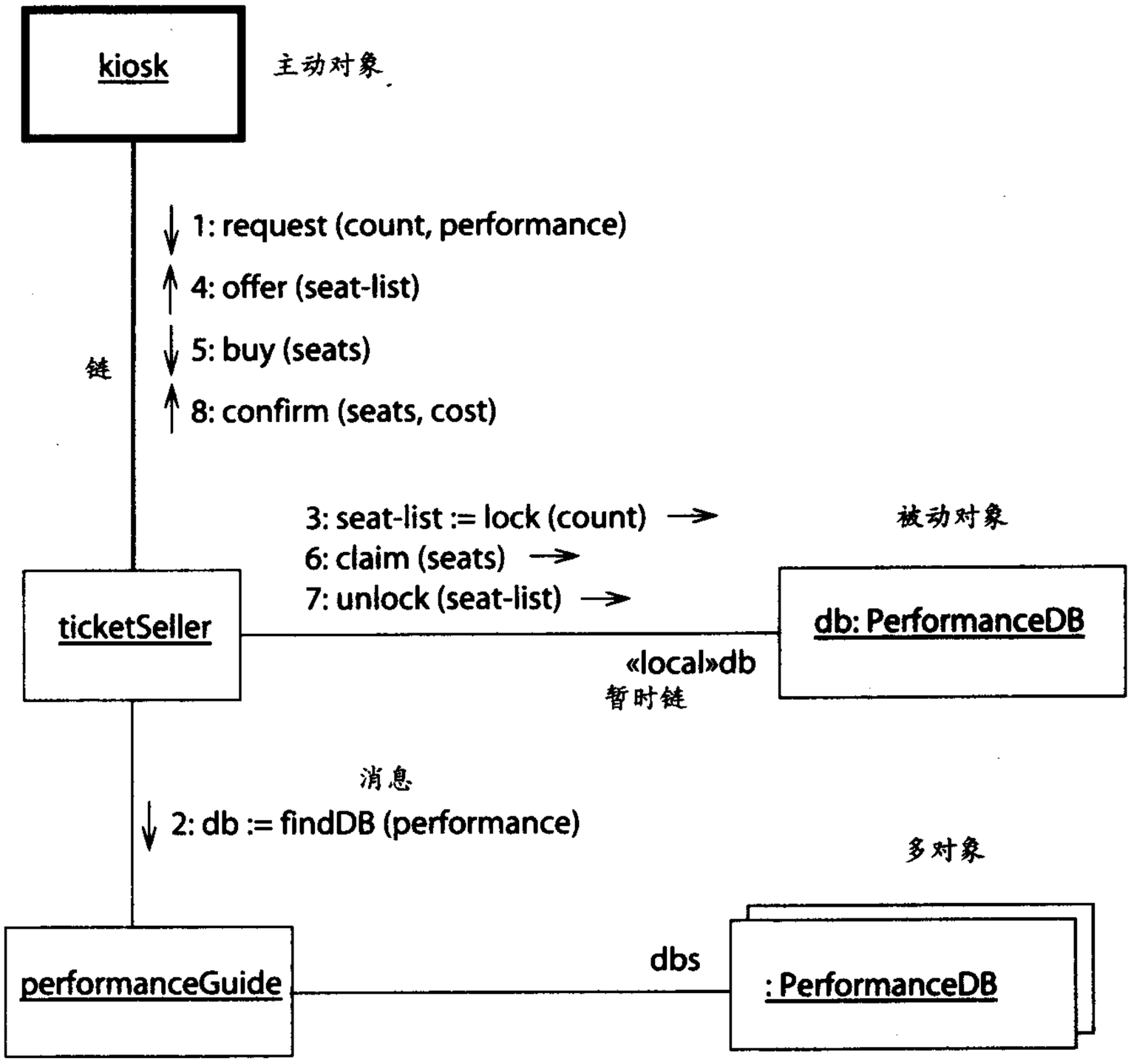


图3-4　协作图

3.5　状态机视图

状态机视图是一个类对象所可能经历的所有历程的模型图。状态机由对象的各个状态和

连接这些状态的转换组成。每个状态对一个对象在其生命期中满足某种条件的一个时间段建模。当一个事件发生时，它会触发状态间的转换，导致对象从一种状态转化到另一新的状态。与转换相关的活动执行时，转换也同时发生。状态机用状态图来表达。

图3-5是票这一对象的状态图。初始状态是Available状态。在票开始对外出售前，一部分票是给预约者预留的。当顾客预定个人票时，被预定的票首先处于锁定状态，此时顾客仍有是否确实要购买这张票的选择权，故这张票可能出售给顾客也可能因为顾客不要这张票而解除锁定状态。如果超过了指定的期限顾客仍未做出选择，此票将被自动解除锁定状态。预约者也可以换其他演出的票，如果这样的话，最初预约票也可以对外出售。

状态图可用于描述用户接口、设备控制器和其他具有反馈的子系统。它还可用于描述在生命期中跨越多个不同性质阶段的被动对象的行为，在每一阶段该对象都有自己特殊的行为。

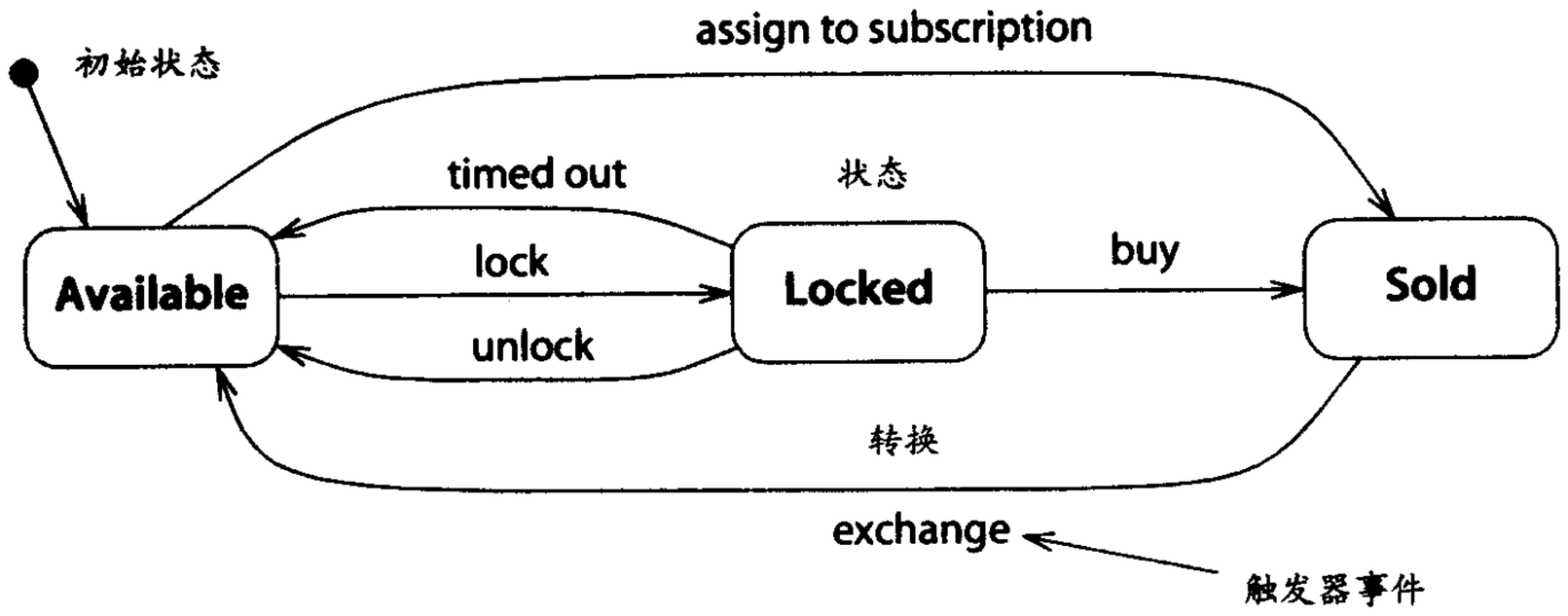


图3-5 状态图

3.6 活动视图

活动图是状态机的一个变体，用来描述执行算法的工作流程中涉及的活动。活动状态代表了一个活动：一个工作流步骤或一个操作的执行。活动图描述了一组顺序的或并发的活动。活动视图用活动图来体现。

图3-6是售票处的活动图。它表示了上演一个剧目所要进行的活动（这个例子仅供参考，不必太认真地凭着看戏的经验而把问题复杂化）。箭头说明活动间的顺序依赖关系——例如，在规划进度前，首先要选择演出的剧目。加粗的横线段表示分叉和结合控制。例如，安排好整个剧目的进度后，可以进行宣传报道、购买剧本、雇用演员、准备道具、设计照明、加工戏服等，所有这些活动都可同时进行。在进行彩排之前，剧本和演员必须已经具备。

这个例子说明活动图的用途是对人类组织的现实世界中的工作流程建模。对事物建模是活动图的主要用途，但活动图也可对软件系统中的活动建模。活动图有助于理解系统高层活动的执行行为，而不涉及建立协作图所必须的消息传送细节。

用连接活动和对象流状态的关系流表示活动所需的输入输出参数。

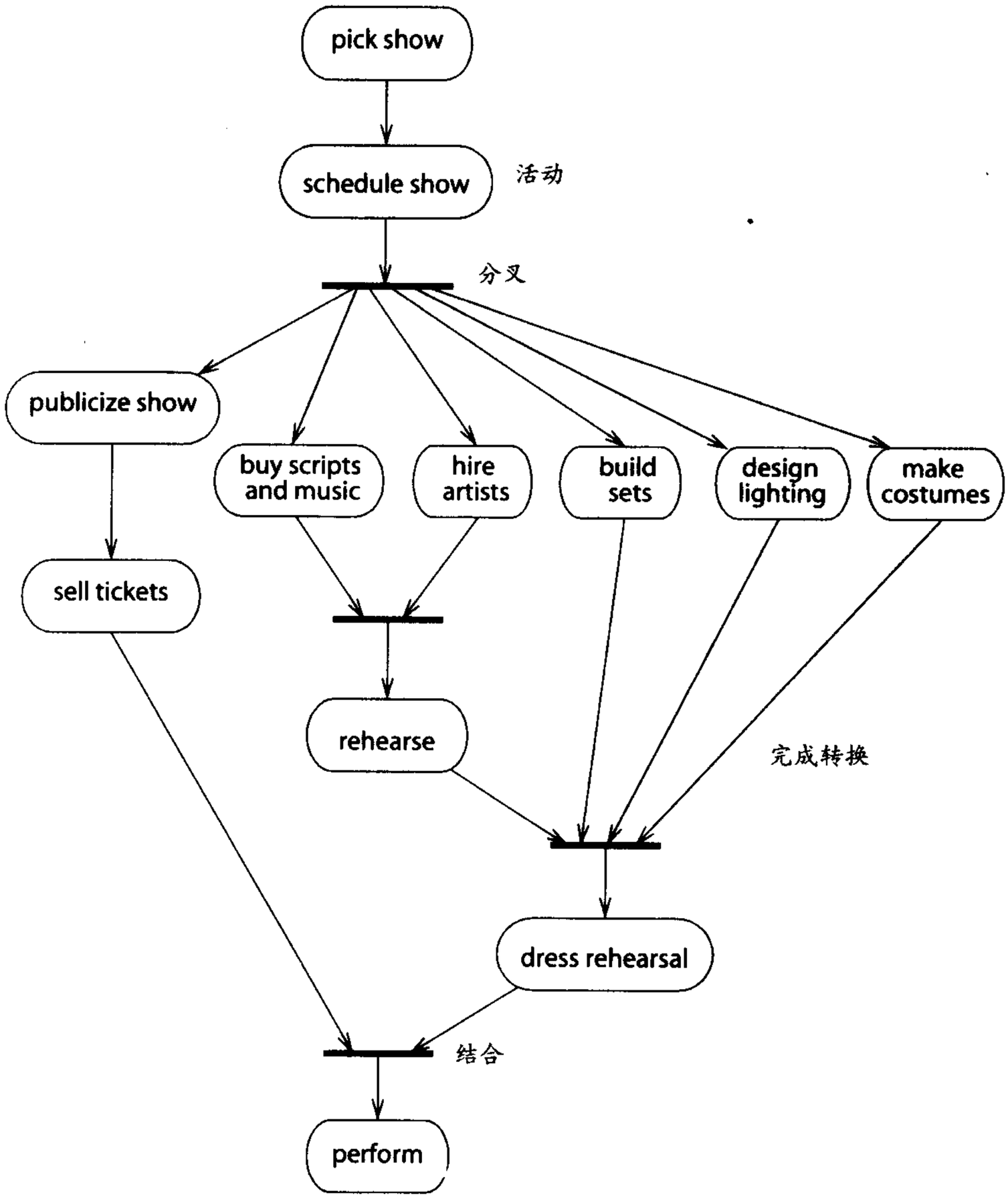


图3-6　活动图

3.7　物理视图

前面介绍的视图模型按照逻辑观点对应用领域中的概念建模。物理视图对应用自身的实现结构建模，例如系统的构件组织和建立在运行节点上的配置。这类视图提供了将系统中的类映射成物理构件和节点的机制。物理视图有两种：实现视图和部署视图。

实现视图为系统的构件建模——构件即构造应用的软件单元——还包括各构件之间的依

赖关系，以便通过这些依赖关系来估计对系统构件的修改给系统可能带来的影响。

实现视图用构件图来表现。图3-7是售票系统的构件图。图中有三个用户接口：顾客和公用电话亭之间的接口、售票员与在线订票系统之间的接口和监督员查询售票情况的接口。售票方构件顺序接受来自售票员和公用电话亭的请求；信用卡主管构件用于处理信用卡付款；还有一个存储票信息的数据库构件。构件图表示了系统中的各种构件。在个别系统的实际物理配置中，可能有某个构件的多个备份。

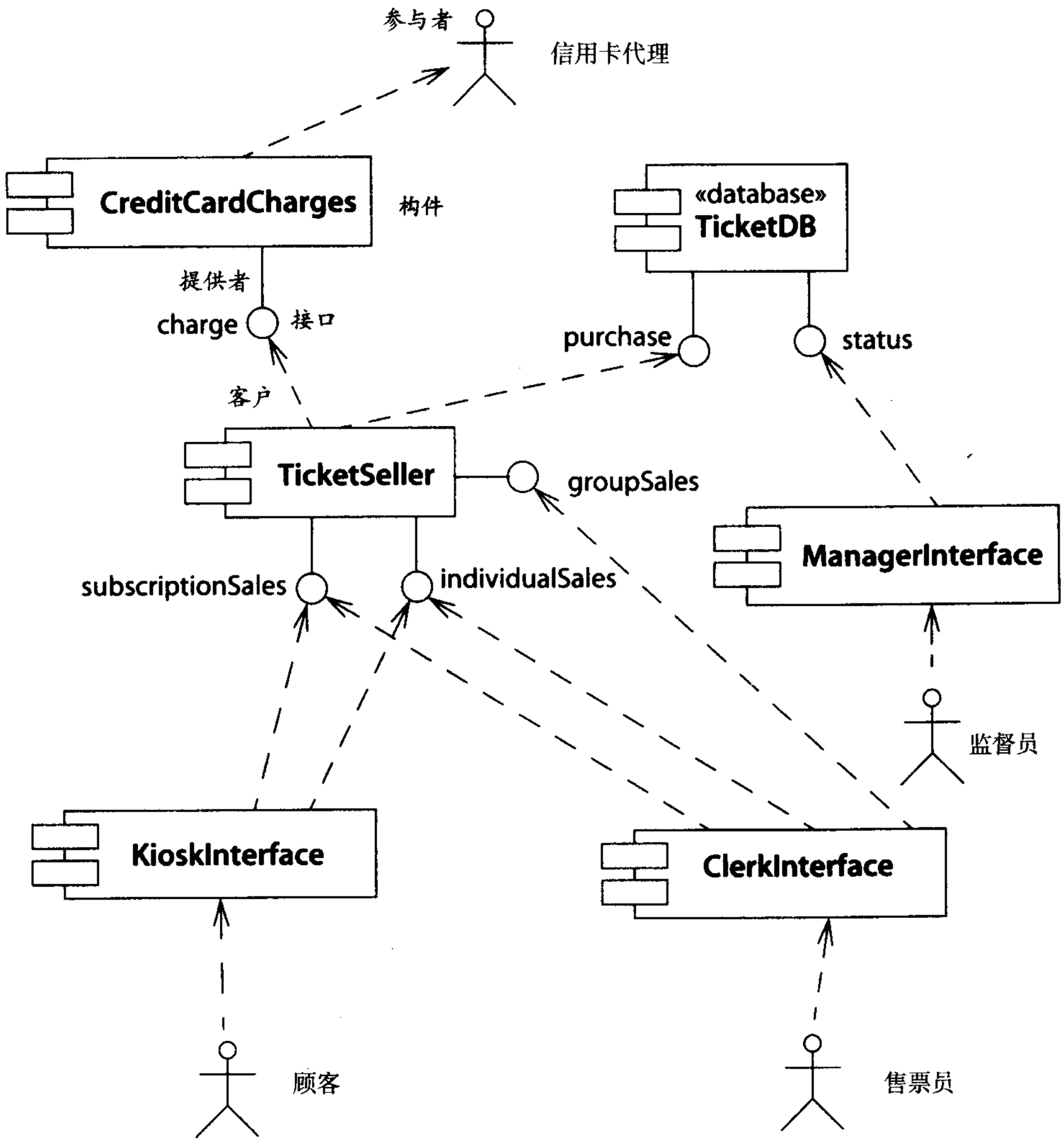


图3-7 构件图

图中的小圆圈代表接口，即服务的连贯集。从构件到接口的实线表明该构件提供的列在接口旁的服务。从构件到接口的虚线箭头说明这个构件要求接口提供的服务。例如，购买个

人票可以通过公用电话亭订购也可直接向售票员购买，但购买团体票只能通过售票员。

部署视图描述位于节点实例上的运行构件实例的安排。节点是一组运行资源，如计算机、设备或存储器。这个视图允许评估分配结果和资源分配。

部署视图用部署图来表达。图3-8是售票系统的描述层部署图。图中表示了系统中的各构件和每个节点包含的构件。节点用立方体图形表示。

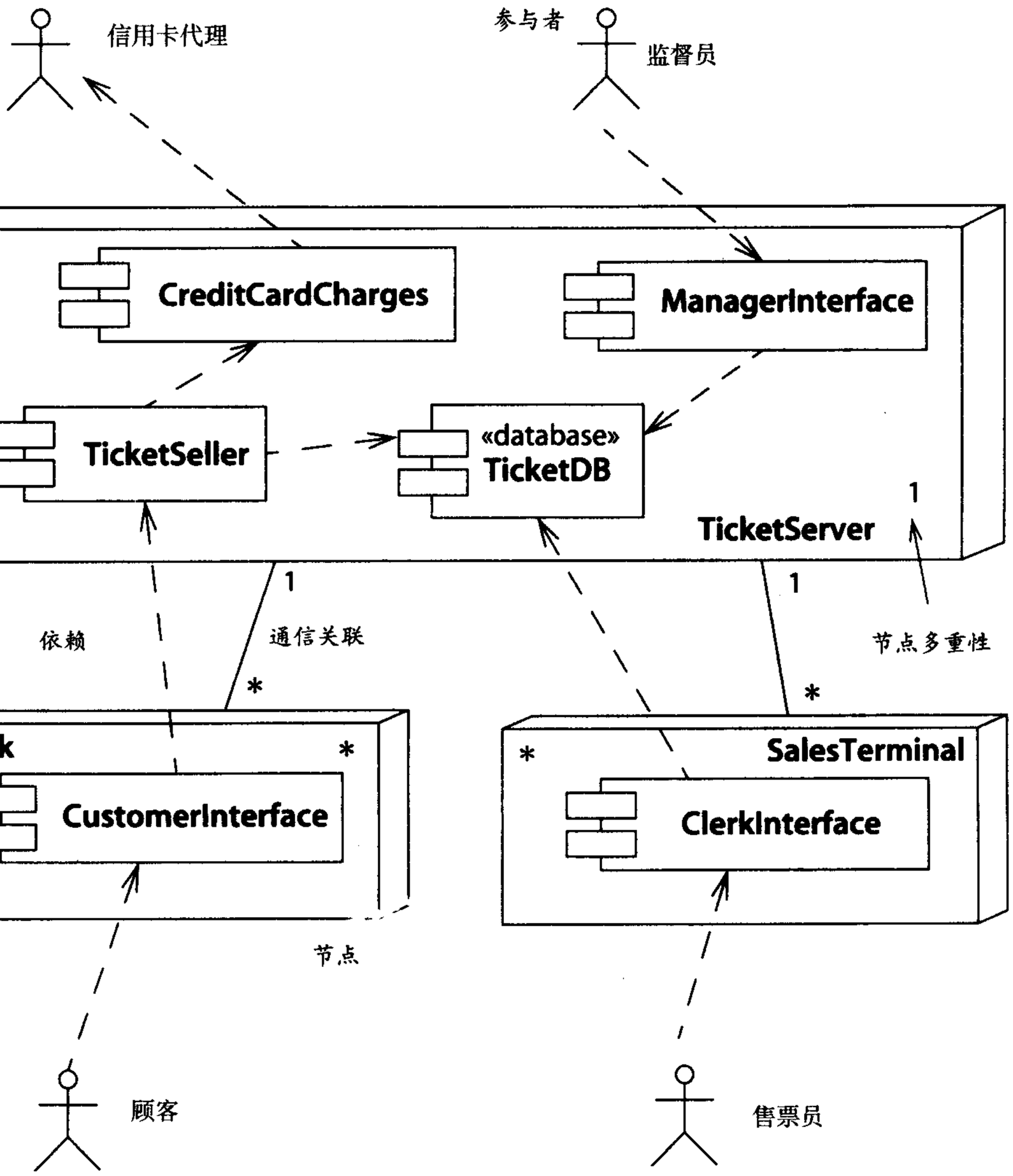


图3-8　部署图（描述层）

图3-9是售票系统的实例层部署图。图中显示了各节点和它们之间的连接。这个模型中的信息是与图3-8的描述层中的内容相互对应的。

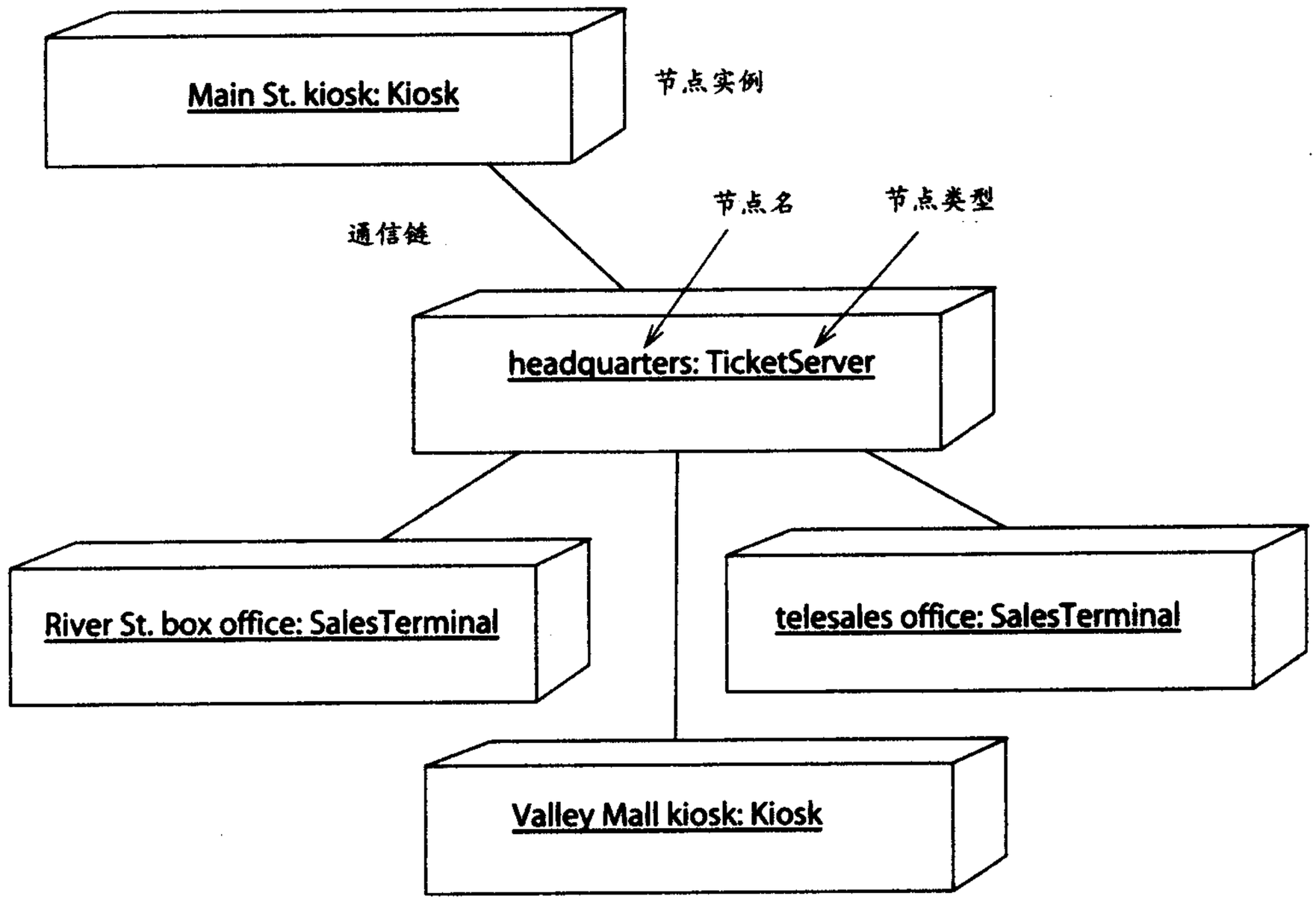


图3-9 部署图（实例层）

3.8 模型管理视图

模型管理视图对模型自身组织建模。一系列由模型元素（如类、状态机和用例）构成的包组成了模型。包可能包含其他的包，因此，整个模型实际上可看成一个根包，它间接包含了模型中的所有内容。包是操作模型内容、存取控制和配置控制的基本单元。每一个模型元素包含于包中或包含于其他模型元素中。

模型是从某一观点以一定的精确程度对系统所进行的完整描述。从不同的视角出发，对同一系统可能会建立多个模型，例如有系统分析模型和系统设计模型之分。模型是一种特殊的包。

子系统是另一种特殊的包。它代表了系统的一个部分，它有清晰的接口，这个接口可作为一个单独的构件来实现。

模型管理信息通常在类图中表达。

图3-10显示了将整个剧院系统分解所得到的包和它们之间的依赖关系。售票处子系统在前面的例子中已经讨论过了，完整的系统还包括剧院管理和计划子系统。每个子系统还包含了多个包。

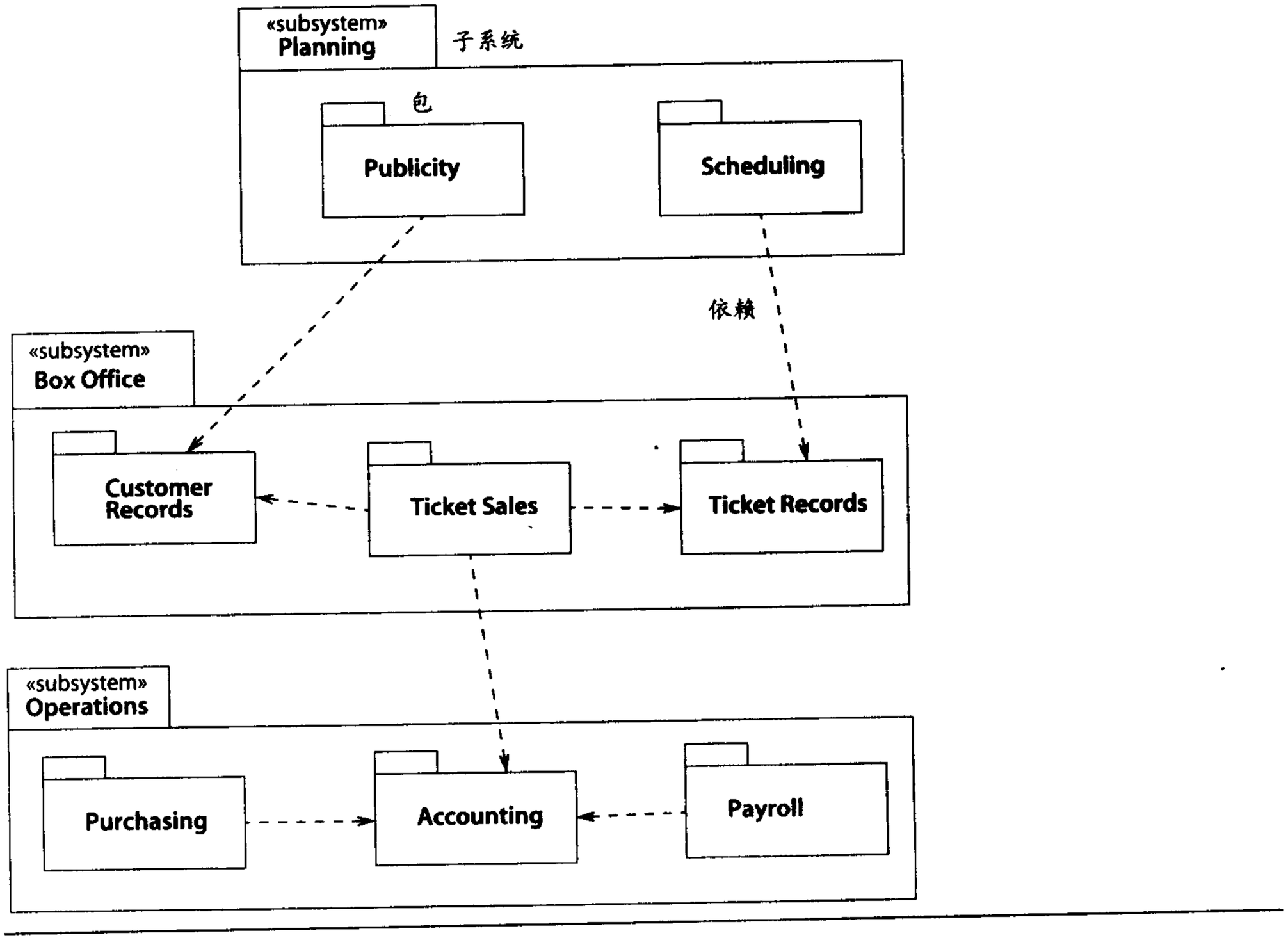


图3-10 包

3.9 扩展组件

UML包含三种主要的扩展组件：约束、构造型和标记值。约束是用某种形式化语言或自然语言表达的语义关系的文字说明。构造型是由建模者设计的新的模型元素，但是这个模型元素的设计要建立在UML已定义的模型元素基础上。标记值是附加到任何模型元素上的命名的信息块。

这些组件提供了扩展UML模型元素语义的方法，同时不改变UML定义的元模型自身的语义。使用这些扩展组件可以组建适用于某一具体应用领域的UML用户定制版本。

图3-11举例说明了约束、构造型和标记值的使用。对剧目类的约束保证了剧目具有唯一的名称。图3-1说明了两个关联的异或约束，一个对象某一时刻只能具有两个关联中的一个。用文字表达约束效果较好，但UML的概念不直接支持文字描述。

TicketdDB构件构造型表明这个是一个数据库构件，允许省略该构件的接口说明，因为这个接口是所有数据库都支持的通用接口。建模者可以增加新的构造型来表示专门的模型元素。一个构造型可以带有多个约束、标记值或者代码生成特性。如图所示，建模者可以为命名的构造型定义一个图标，作为可视化的辅助工具。尽管如此，可以使用文字形式说明。

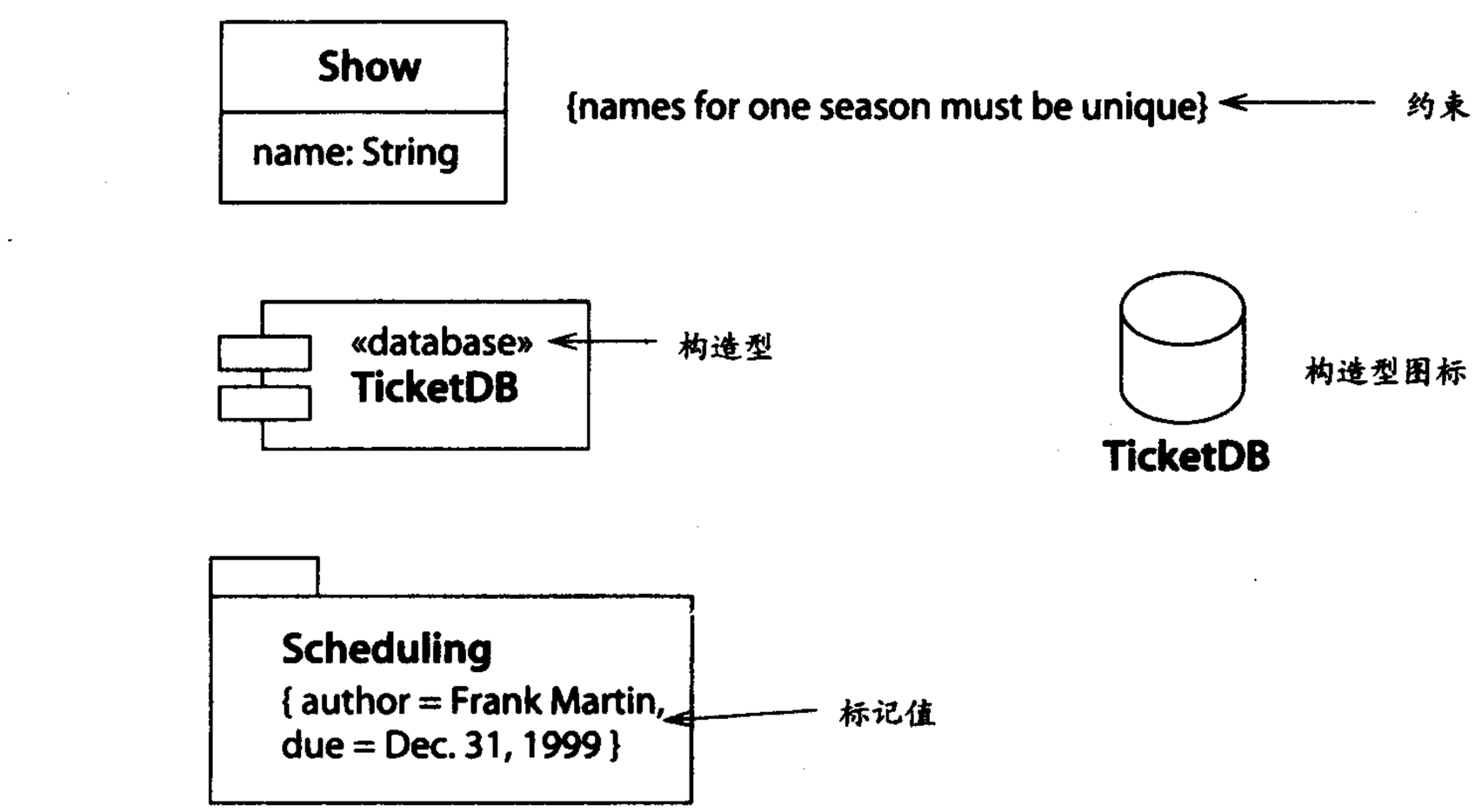


图3-11 扩展组件

Scheduling包中的标记值说明Frank Martin要在年底前完成计划的制定。可以将任意信息作为标记值写于一个模型元素中建模者选定的名字之下。使用文字有益于描述项目管理和代码生成参数。大部分标记值被保存为编辑工具中的弹出信息，在正式打印出的图表中通常没有标记值。

3.10 各种视图间的关系

多个视图共存于一个模型中，它们的元素之间有很多关系，其中一些关系列在表3-2中。表中没有将各种关系列全，但它列出了从不同视角观察得到的元素间的部分主要关系。

表3-2 不同视图元素间的部分关系

元 素	元 素	关 系
类	状态机	拥有
操作	交互	实现
用例	协作	实现
用例	交互实例	样本场景
构件实例	节点实例	位置
动作	操作	调用
动作	信号	发送
活动	操作	调用
消息	动作	激发
包	类	拥有
角色	类	分类