



软件开发环境国家重点实验室
State Key Laboratory of Software Development Environment

系统建模



- **系统建模就是建立系统抽象模型的过程，其中每一个模型表示系统的一个不同的视角或观点。**
- **系统建模通常意味着在统一建模语言（ Unified Modeling Language , UML ）中的图类型基础上，使用某种图形化的表示法，表示系统。**
- **System modelling helps the analyst to understand the functionality of the system and models are used to communicate with customers.**
- **在需求工程过程中使用模型：为了帮助得到详细的系统需求**
- **在设计过程中使用模型：为了向实现系统的工程师描述系统**
- **在实现系统之后还要使用模型：为了描述系统的结构和运行**

需求工程过程中的系统建模

- 使用现有系统的模型帮助阐明现有系统做什么，并且可以用于让利益相关者之间的讨论聚焦于当前系统的优势和弱点，进而引导出新系统的需求。
- 新系统的模型帮助解释对其他系统利益相关者所提出的需求，工程师使用这些模型来讨论设计方案并描述系统以用于实现。

- **外部视角**：对系统的上下文或环境进行建模
- **交互视角**：对系统及其环境或者系统的构件之间的交互进行建模
- **结构化视角**：对系统的组织或者系统所处理的数据结构进行建模
- **行为视角**：对系统的动态行为以及系统如何响应事件进行建模



- **As a means of facilitating discussion about an existing or proposed system**
 - ❑ Incomplete and incorrect models are OK as their role is to support discussion.
- **As a way of documenting an existing system**
 - ❑ Models should be an accurate representation of the system but need not be complete.
- **As a detailed system description that can be used to generate a system implementation**
 - ❑ Models have to be both correct and complete.

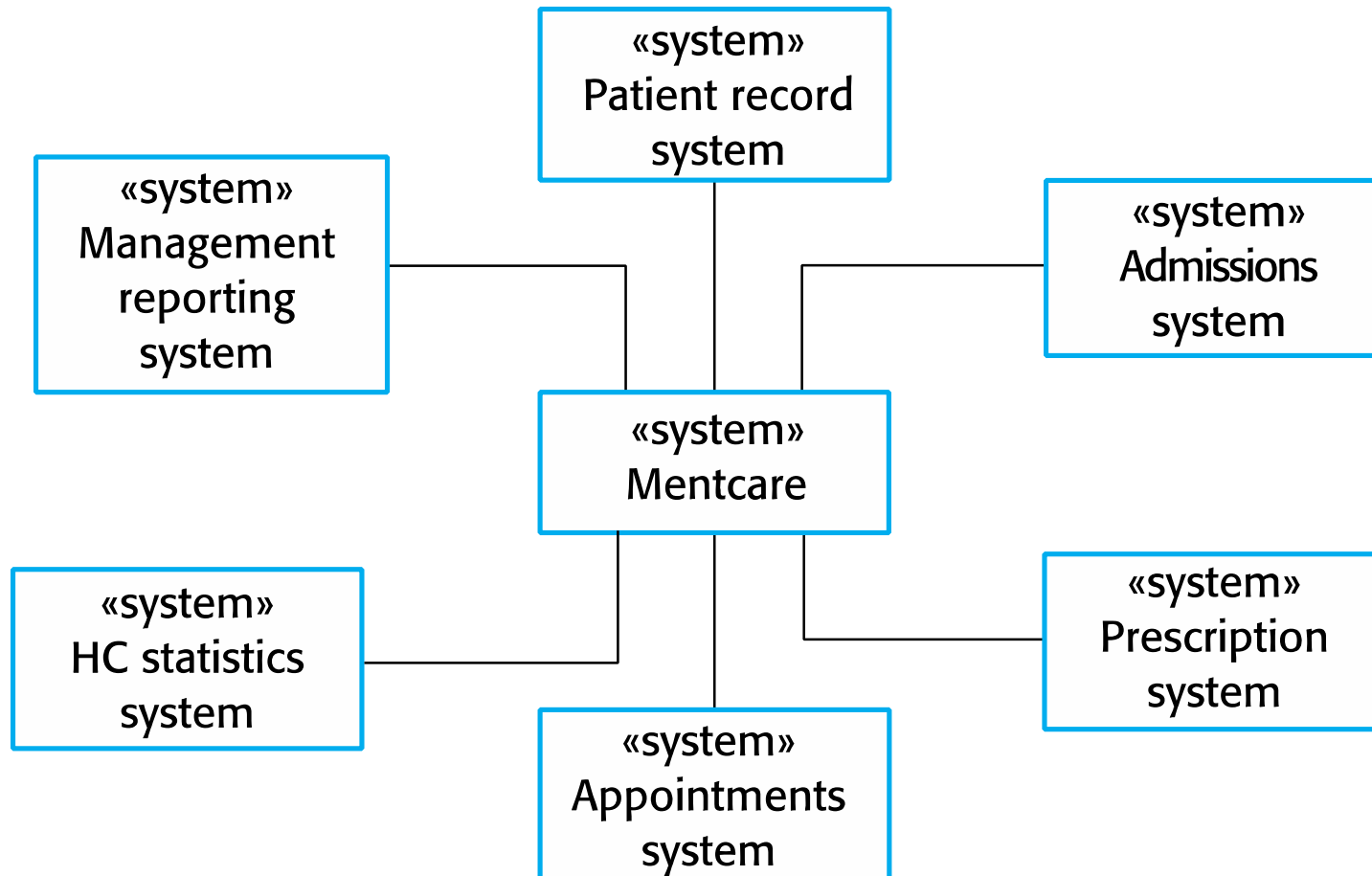


- **活动图 (activity diagram)** , 描述过程或者数据处理中所包含的活动
- **用例图 (use case diagram)** , 描述系统及其环境之间的交互
- **顺序图 (sequence diagram)** , 描述参与者与系统之间以及系统构件之间的交互
- **类图 (class diagram)** , 描述系统中的对象类以及这些类之间的联系
- **状态图 (state diagram)** , 描述系统如何对内部和外部的事件做出响应



- **在系统规格说明的早期阶段应当对系统边界进行确定，即明确哪些属于、哪些不属于所开发的系统。**
- **Context models are used to illustrate the operational context of a system - they show what lies outside the system boundaries.**
- **Social and organisational concerns may affect the decision on where to position system boundaries.**
- **Architectural models show the system and its relationship with other systems.**

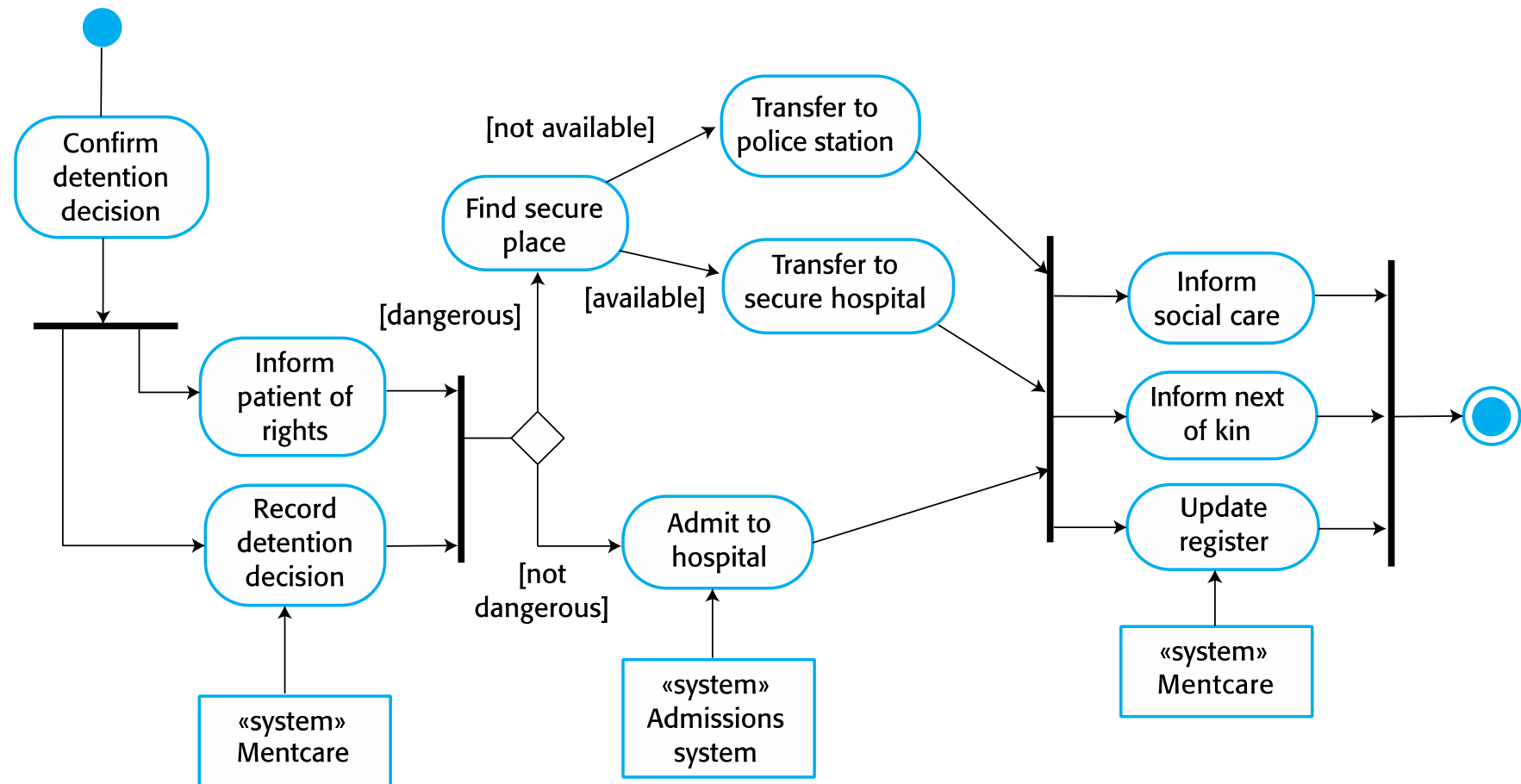
Mentcare系统上下文



上下文模型与其他模型配合使用

- **Context models simply show the other systems in the environment, not how the system being developed is used in that environment.**
- **Process models reveal how the system being developed is used in broader business processes.**
- **UML activity diagrams may be used to define business process models.**

使用UML活动图描述业务过程模型



交互模型



- 对用户交互的建模有助于识别用户需求
- 对系统间交互的建模可以凸显出可能出现的通信问题
- 对构件交互的的建模可以帮助理解，所提出的系统结构是否能实现所要求的系统性能和依赖性
- 可使用用例图和顺序图来实现交互建模
- 用例图：主要用于建模系统与外部主体（用户或者其他系统）之间的交互
- 顺序图：用于建模系统构件之间，以及与外部主体之间的交互
- 二者分别描述不同抽象层次上的交互，因此可以一起使用

用例图

- 用例图可以用于需求抽取，可以作为用户在该交互中对系统期望的简单描述
- 用例图在系统设计的早期阶段更为有用，用于表示包含系统外部交互的离散任务
- 用例图中的参与者（Actor）可以是用户（人），也可能是其他系统
- 用例图给出了一个相对简单的概览，如果要获得完整的交互描述，还需要增加更多的细节，可以配合文本描述、结构化描述或者顺序图等。



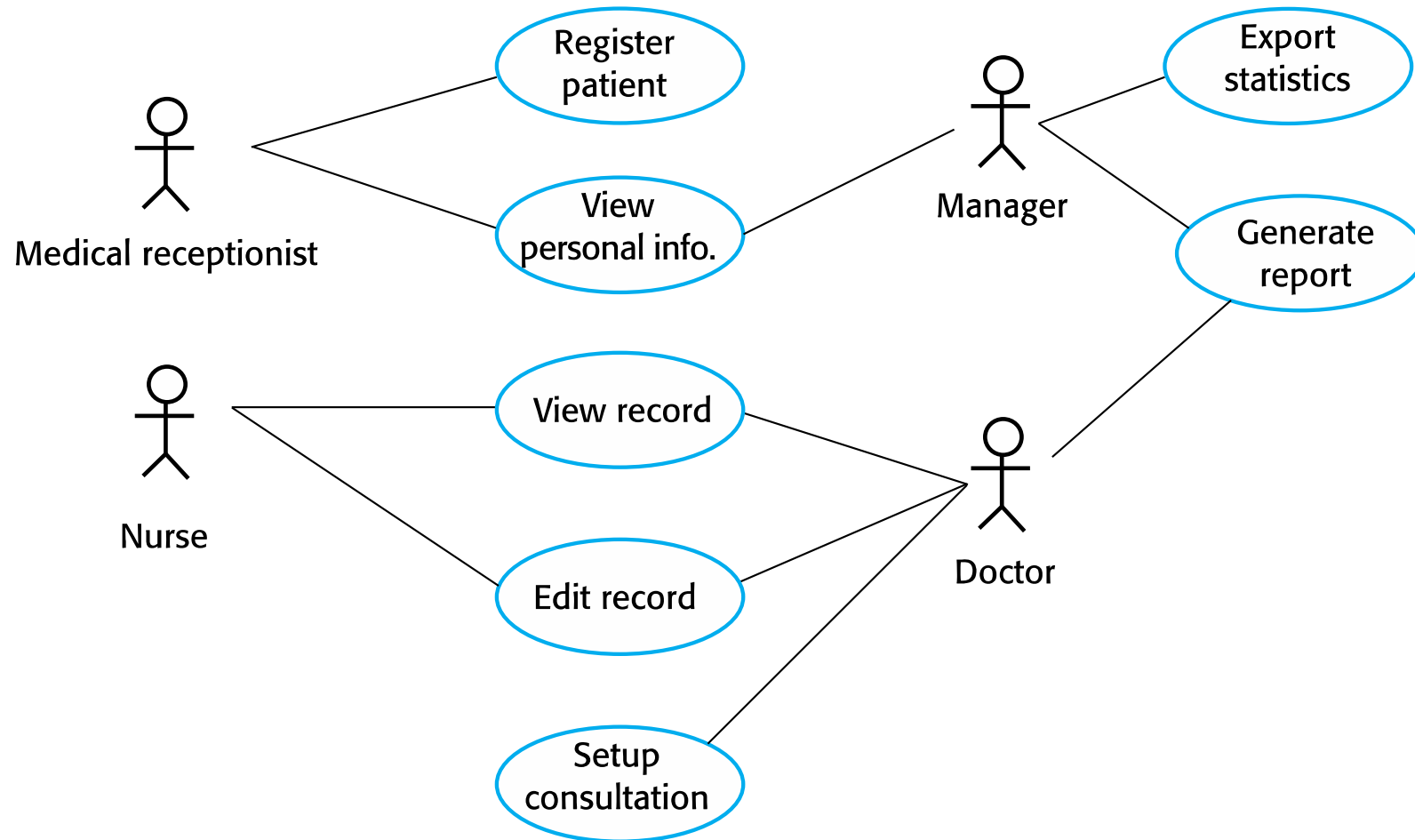
“传输数据”用例的表格化描述



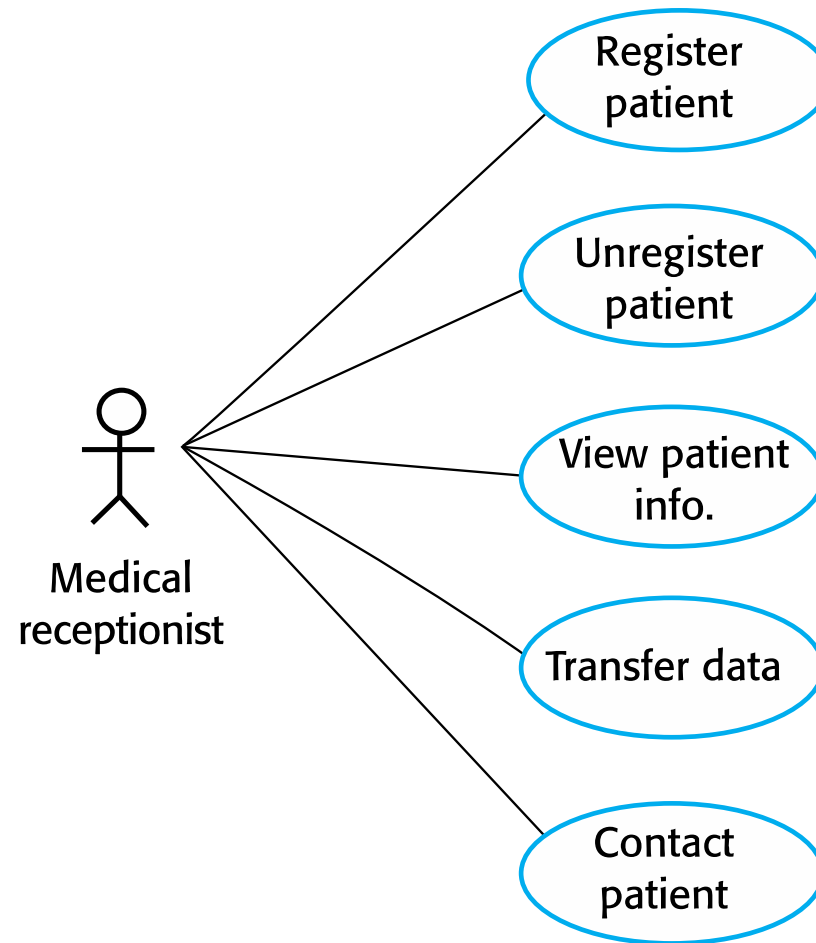
Mentcare系统：传输数据

参与者	医疗接待员、病人记录系统(PRS)
描述	医疗接待员可以通过本系统向健康管理机构维护的通用病人记录数据库传输数据，所传输的信息可以是更新的个人信息（地址、电话等）或者病人的诊断和治疗情况总结
数据	病人个人信息、治疗情况总结
触发机制	医疗接待员发出的用户命令
响应	病人记录系统已经更新的确认信息
注释	医疗接待员必须具有病人信息以及PRS的访问许可

Mentcare系统的用例



“医疗接待员” 角色用例

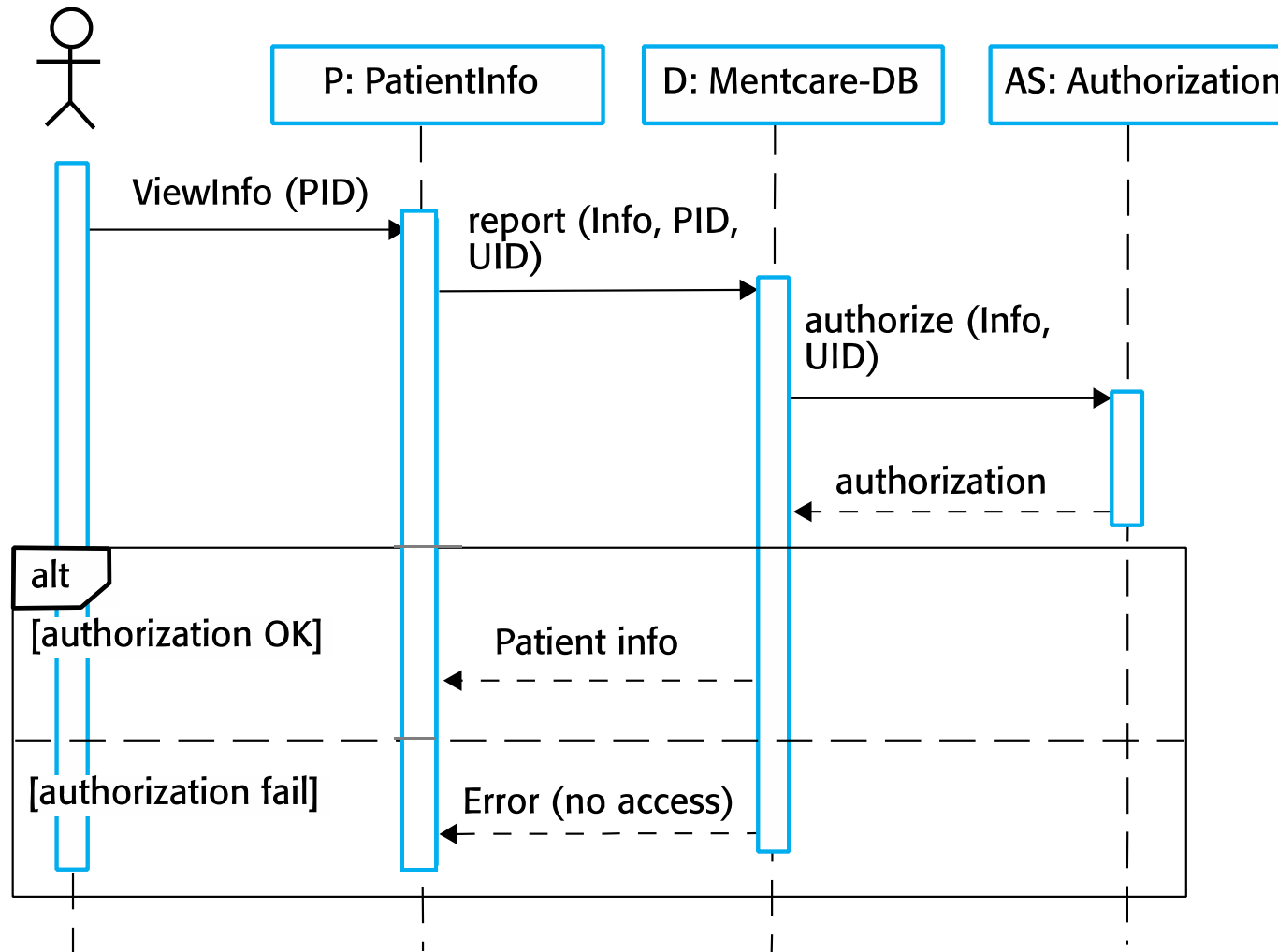


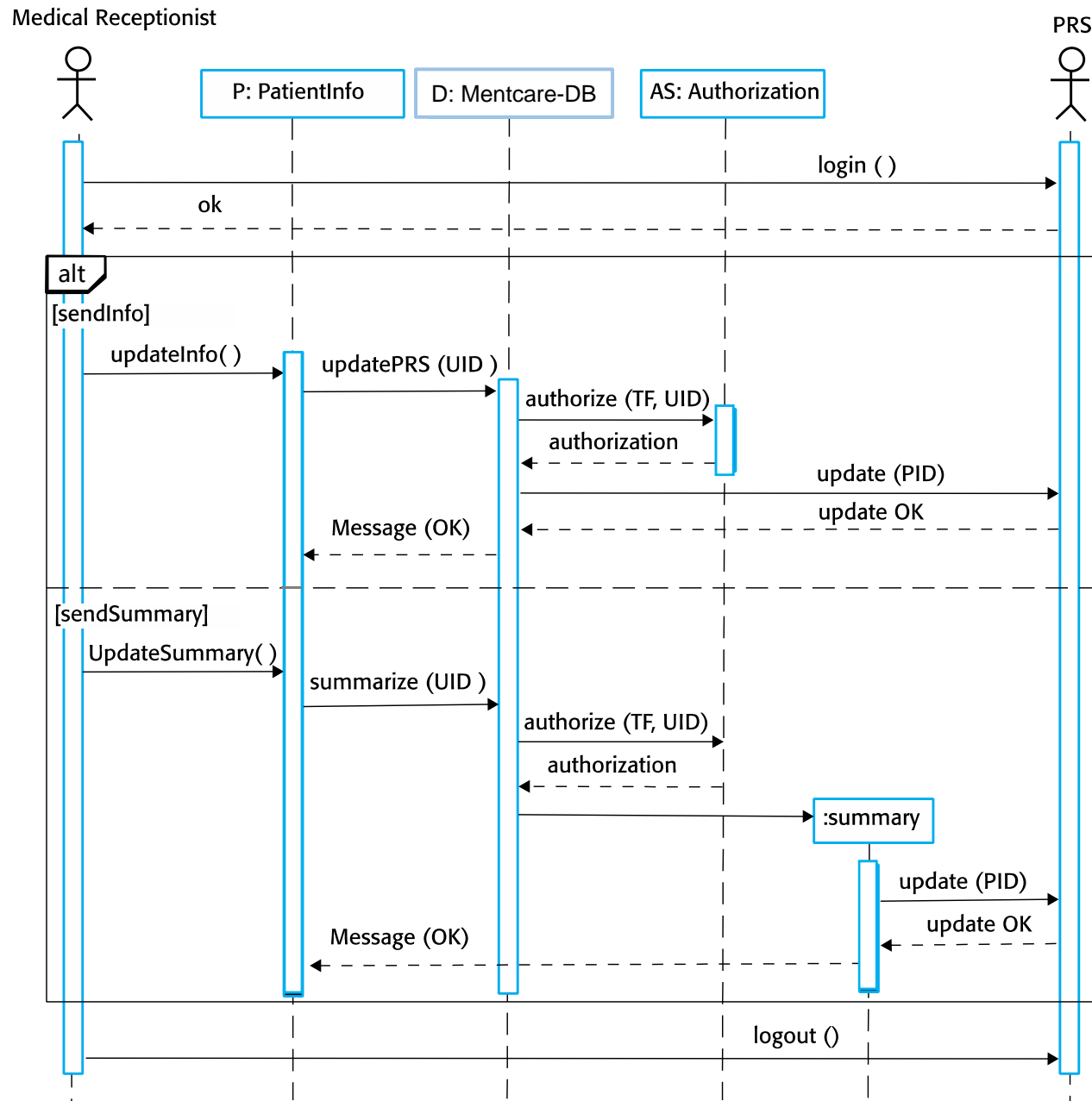


- UML中的顺序图主要用于对参与者以及系统内部对象的交互进行建模（包括对象之间的交互）
- 顺序图用于显示在一个特定的用例或者用例实例执行过程中发生的交互序列
- The objects and actors involved are listed along the top of the diagram, with a dotted line drawn vertically from these.
- Interactions between objects are indicated by annotated arrows.

查看病人信息的顺序图

Medical Receptionist



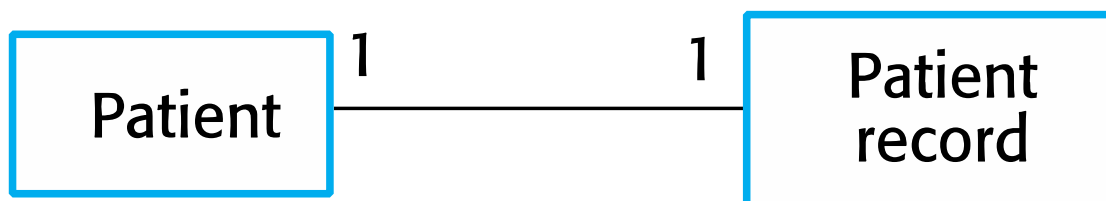


传输数据
顺序图

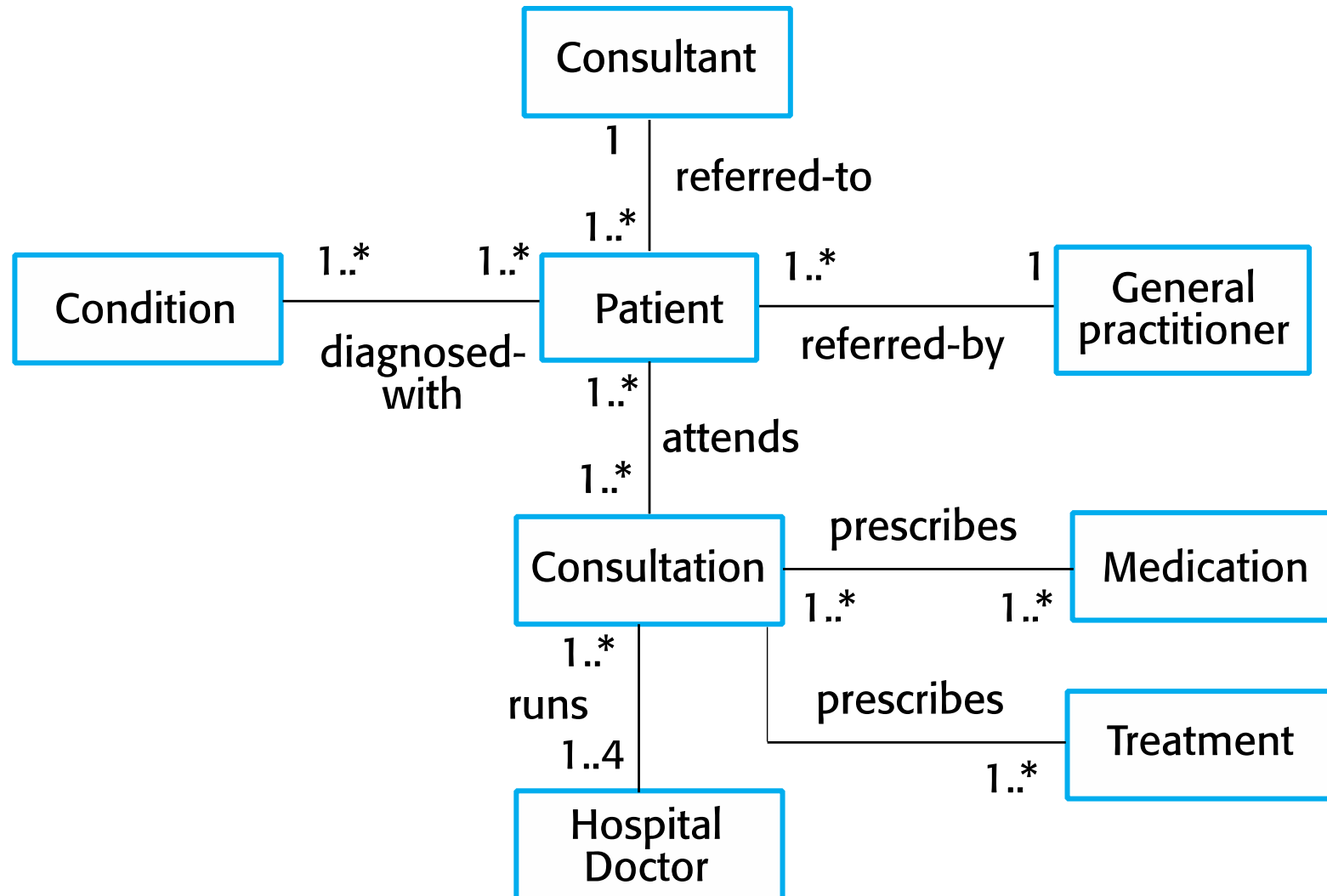


- **软件的结构模型按照构成系统的构件以及它们之间的关系显示系统的组织**
- **结构模型可以是描述系统设计组织的静态模型，也可以是描述系统执行时组织的动态模型**
- **在讨论和设计系统体系结构时创建系统的结构模型，包括总体的系统体系结构模型，或者系统中对象及其关系的模型**

- 在构建一个面向对象系统模型来显示系统中的类以及类之间的关联时，可以使用类图
- 可以将对象类理解为一类对象的泛化定义
- 关联是类之间的链接，表示这些类之间存在某种关系
- 在软件工程过程早期阶段构建模型时，对象表示显示世界中的一些事物，比如：病人、处方、医生等
 - 后期阶段还会要定义实现对象来表示系统所处理的数据



Mentcare系统中的类和关联关系



Consultation (诊疗) 类

Consultation

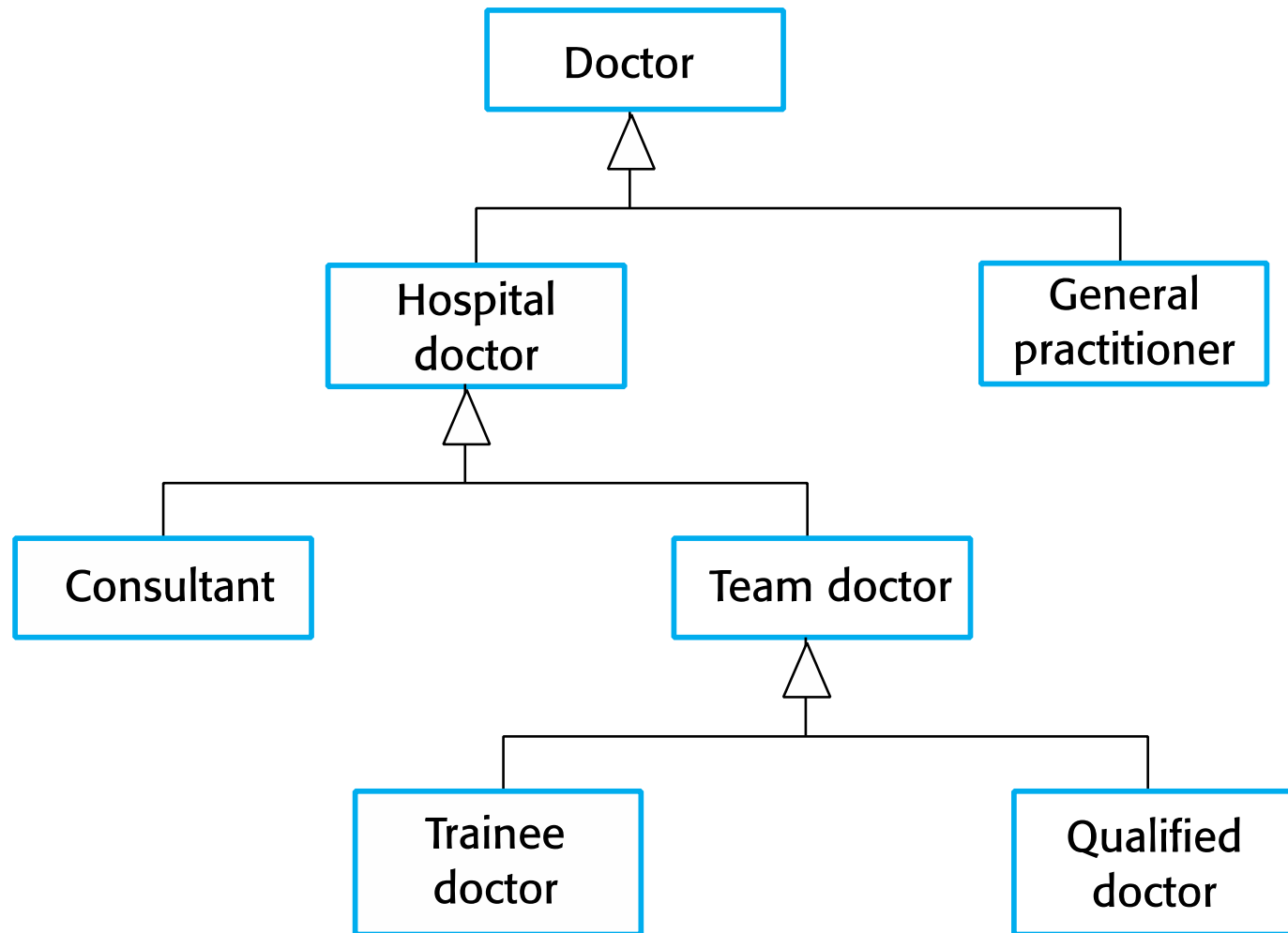
Doctors
Date
Time
Clinic
Reason
Medication prescribed
Treatment prescribed
Voice notes
Transcript
...

New ()
Prescribe ()
RecordNotes ()
Transcribe ()
...

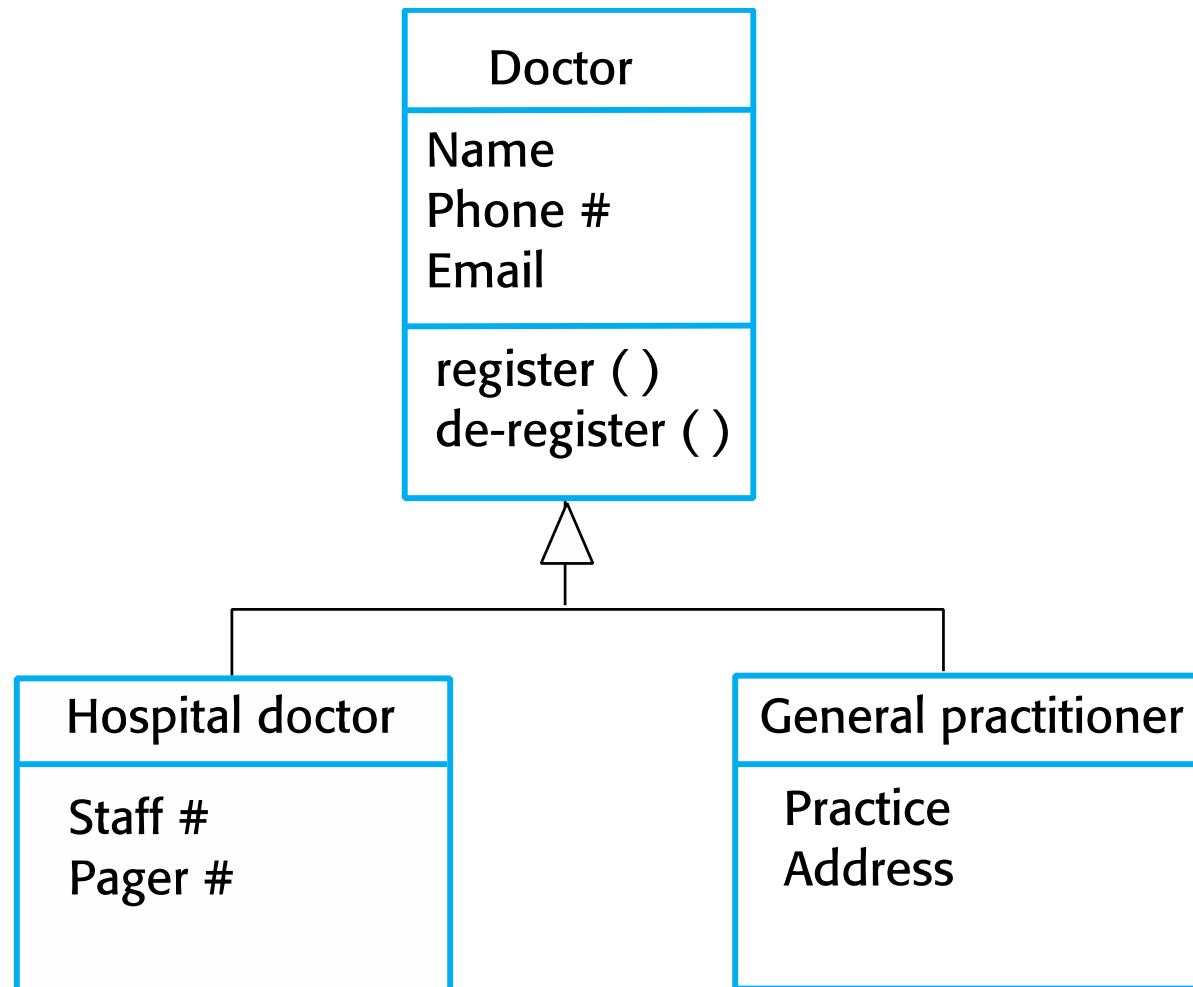
泛化 (Generalization) 关系

- **Generalization is an everyday technique that we use to manage complexity.**
- **我们在日常生活中并不是从所经历的所有事情的详细特性中进行学习，而是学习通用的类（动物、汽车、房屋等）以及这些类的特性**
- **再次基础上我们通过对事物进行分类来复用这些知识，并关注其所属的类之间的差异**

泛化层次

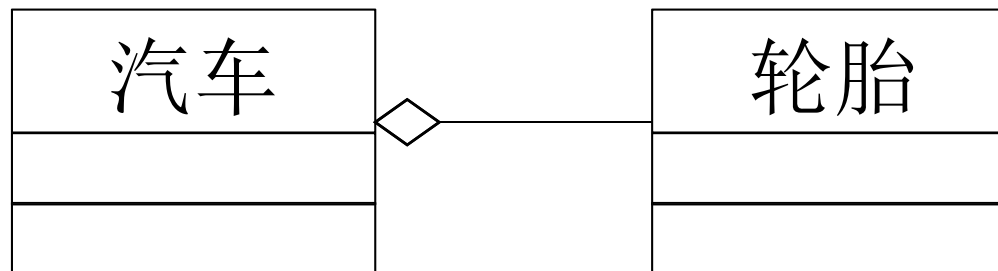


泛化层次（含细节描述）

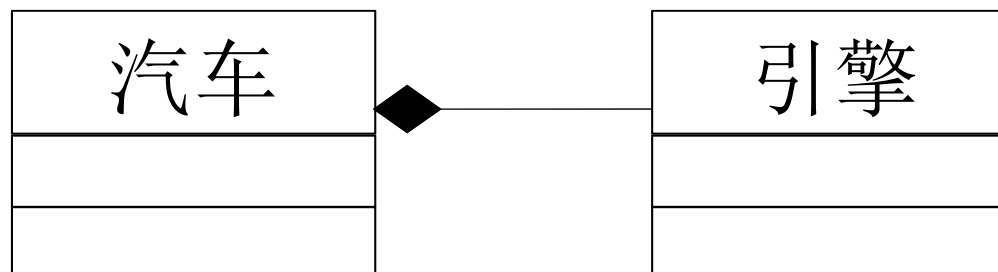


整体-部分 (Whole-Part) 关系

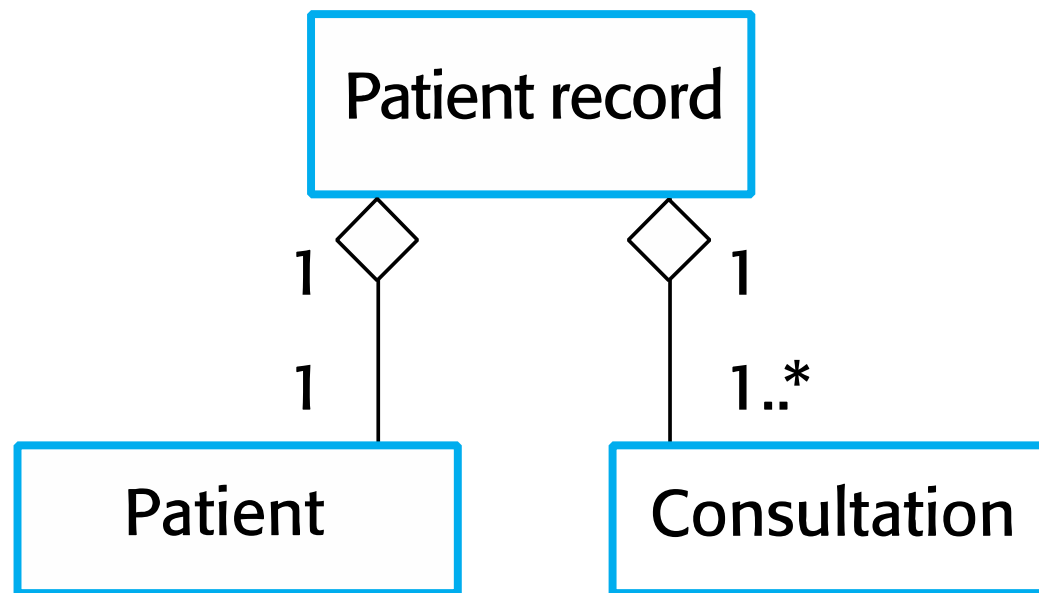
- 聚合 (Aggregation)



- 组合 (Composition)



聚合 (Aggregation) 关系



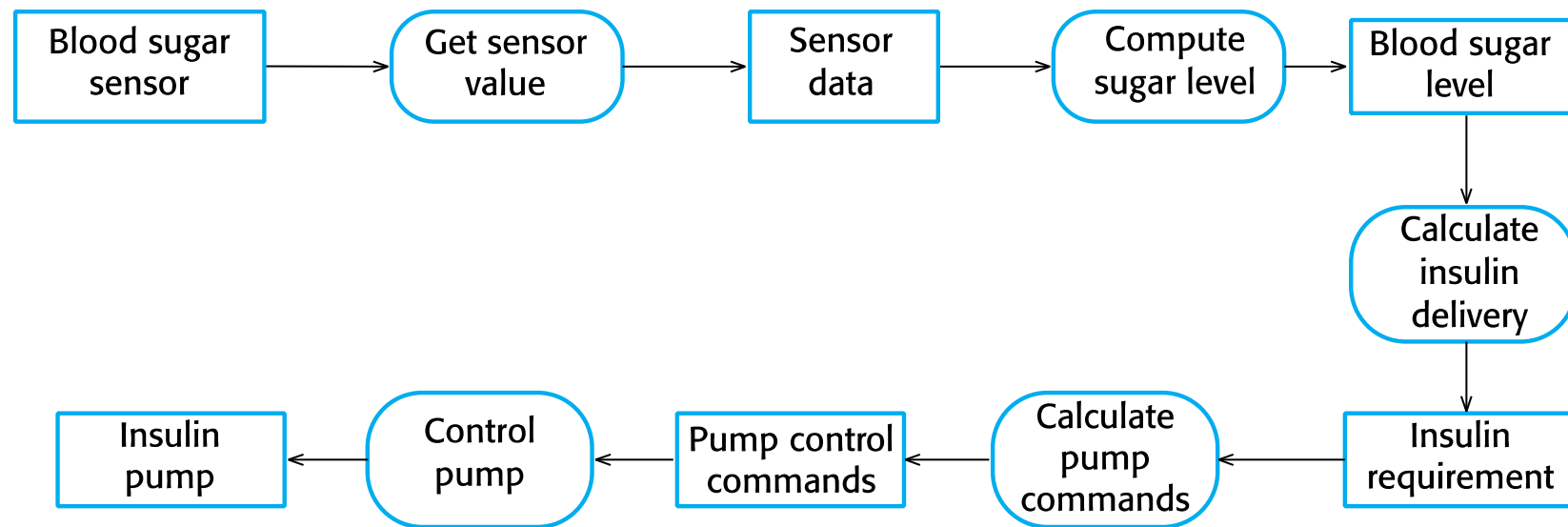


- **行为模型是对系统运行时动态行为的建模，描述了系统对环境触发的响应**
- **触发有两种类型**
 - **数据触发：由数据的产生触发相应的处理过程**
 - **事件触发：某些事件的发生触发了系统处理（相关联的数据可能同时存在）**

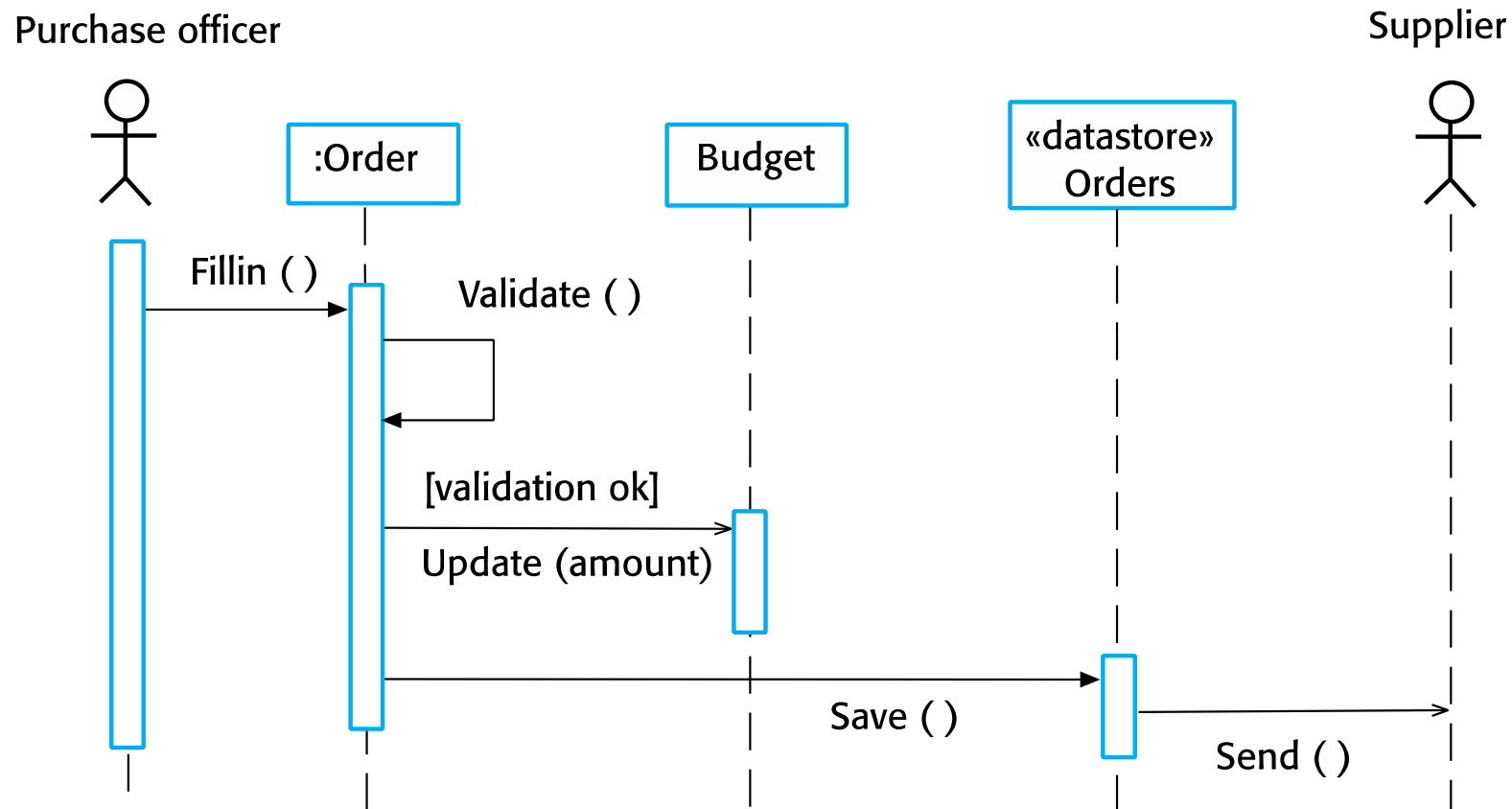


- **Many business systems are data-processing systems that are primarily driven by data. They are controlled by the data input to the system, with relatively little external event processing.**
- **数据驱动的描述处理输入数据以及生成相关的输出过程中所涉及的动作序列**
- **适用于需求分析，描述系统的端到端处理，即从处理一个初始的输入到产生相应的输出（系统的响应）之间所发生的全部动作序列**

活动模型示意图



使用顺序图描述系统中的处理序列



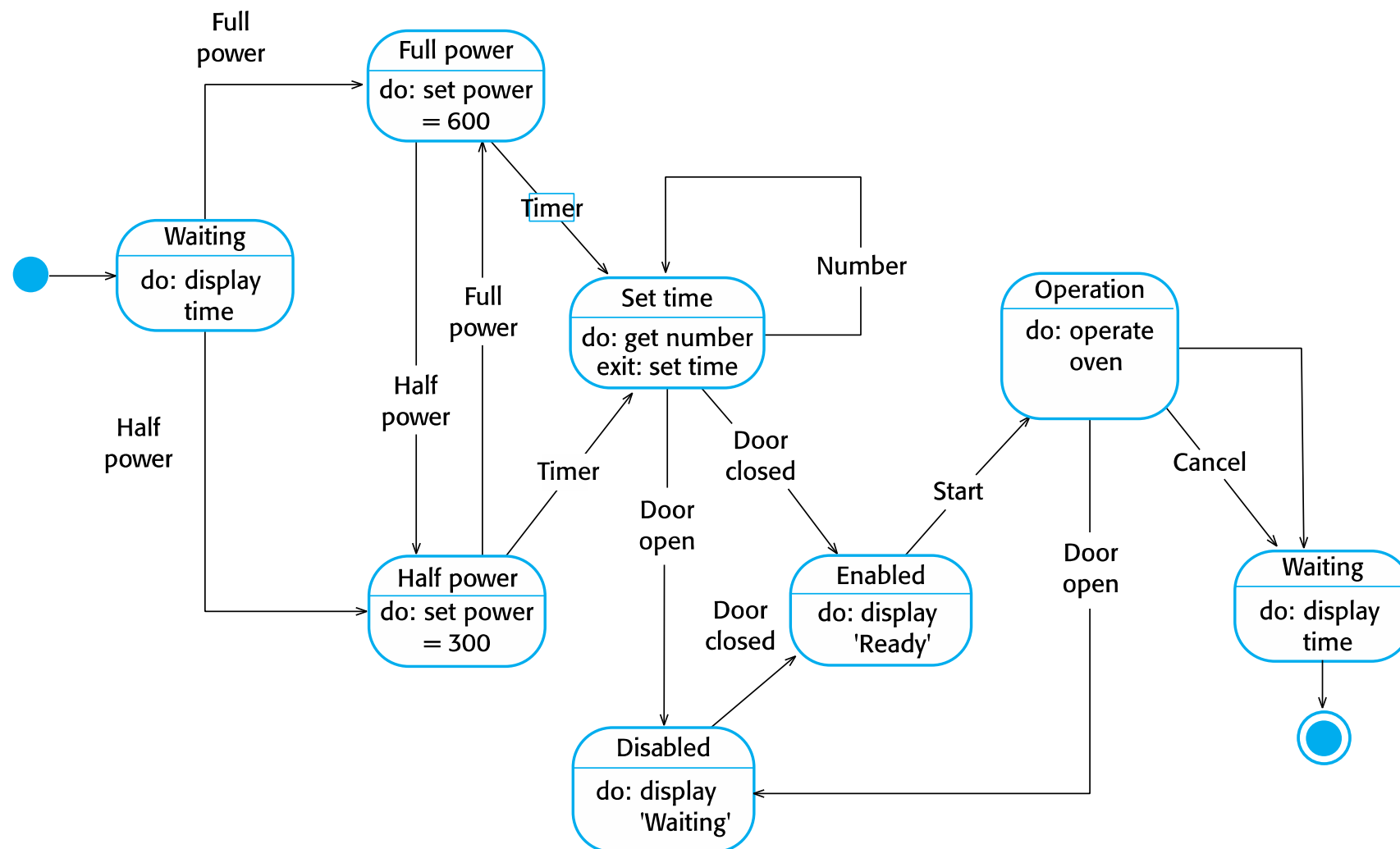


- **Real-time systems are often event-driven, with minimal data processing. For example, a landline phone switching system responds to events such as 'receiver off hook' by generating a dial tone.**
- **事件驱动的建模描述系统如何对外部和内部事件作出响应**
- **这种模型建立在系统具有有限数量的状态以及事件（触发）可以导致状态间的转换**

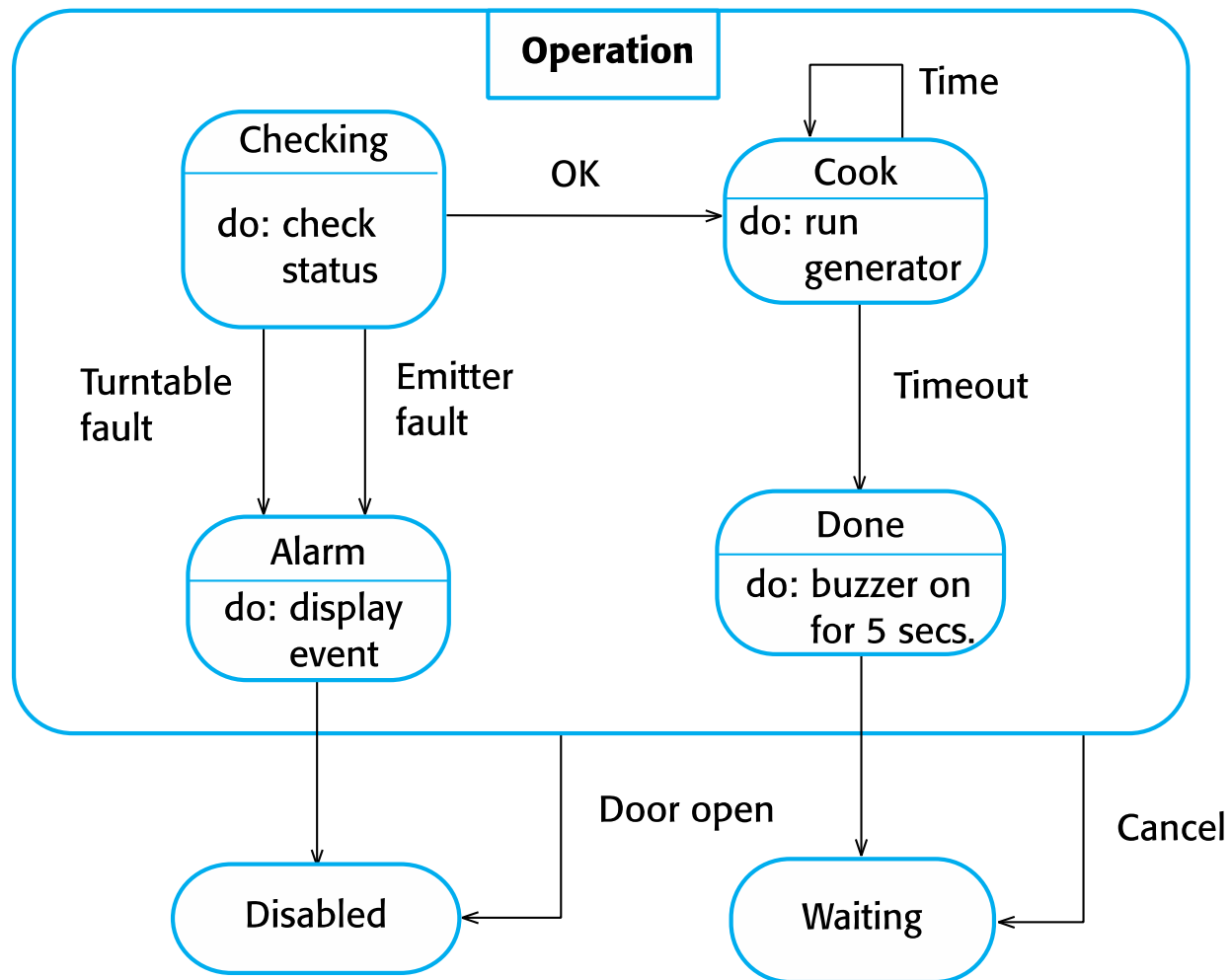


- These model the behaviour of the system in response to external and internal events.
- They show the system's responses to stimuli so are often used for modelling real-time systems.
- 状态机模型将系统状态表示为节点，将事件表示为节点之间的弧，当事件发生时，系统从一个状态转移到另一个状态
- Statecharts are an integral part of the UML and are used to represent state machine models.

微波烤箱状态图



微波烤箱的操作状态



微波烤箱的状态



State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows 'Half power'.
Full power	The oven power is set to 600 watts. The display shows 'Full power'.
Set time	The cooking time is set to the user's input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows 'Not ready'.
Enabled	Oven operation is enabled. Interior oven light is off. Display shows 'Ready to cook'.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for five seconds. Oven light is on. Display shows 'Cooking complete' while buzzer is sounding.

微波烤箱的事件触发机制

Stimulus	Description
Half power	The user has pressed the half-power button.
Full power	The user has pressed the full-power button.
Timer	The user has pressed one of the timer buttons.
Number	The user has pressed a numeric key.
Door open	The oven door switch is not closed.
Door closed	The oven door switch is closed.
Start	The user has pressed the Start button.
Cancel	The user has pressed the Cancel button.



软件开发环境国家重点实验室
State Key Laboratory of Software Development Environment

系统建模案例

面向对象的建模过程



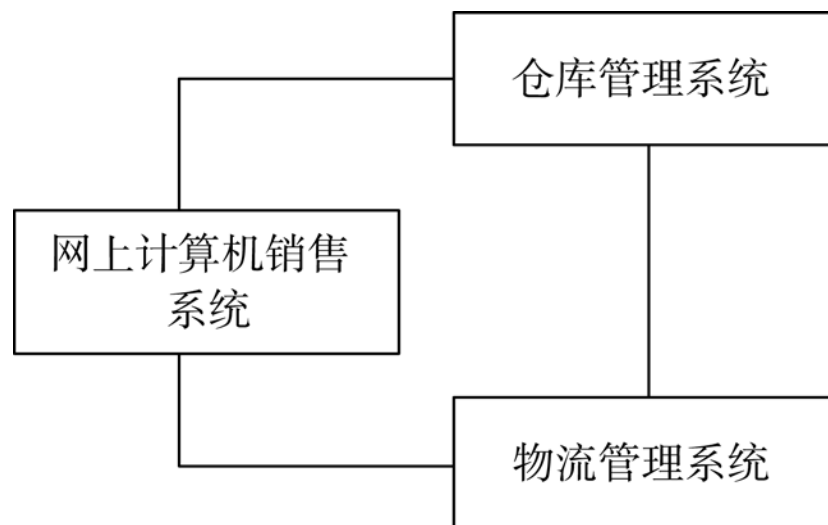
- (1) 通过与用户沟通了解用户的基本需求。
- (2) 确定系统的边界，定义系统做什么和不做什么，以及目标系统和其他外部系统的交互关系，建立上下文模型。
- (3) 了解系统的业务流程，建立**活动图模型**。
- (4) 从用户与系统交互的角度，确定目标系统功能，建立**用例模型**。
- (5) 通过识别问题域内的全部实体对象和类，定义其属性和方法，类之间层次关系，建立系统静态结构**类图模型**。
- (6) 基于用例，通过**顺序（时序）图**描述系统内各对象之间的交互关系。
- (7) 识别对象的行为和系统的工作过程，**状态图**从事件驱动角度分析对象状态的变化，完善类图。
- (8) 迭代地执行步骤1~7，直到完成模型的建立。

案例说明：网上计算机销售系统



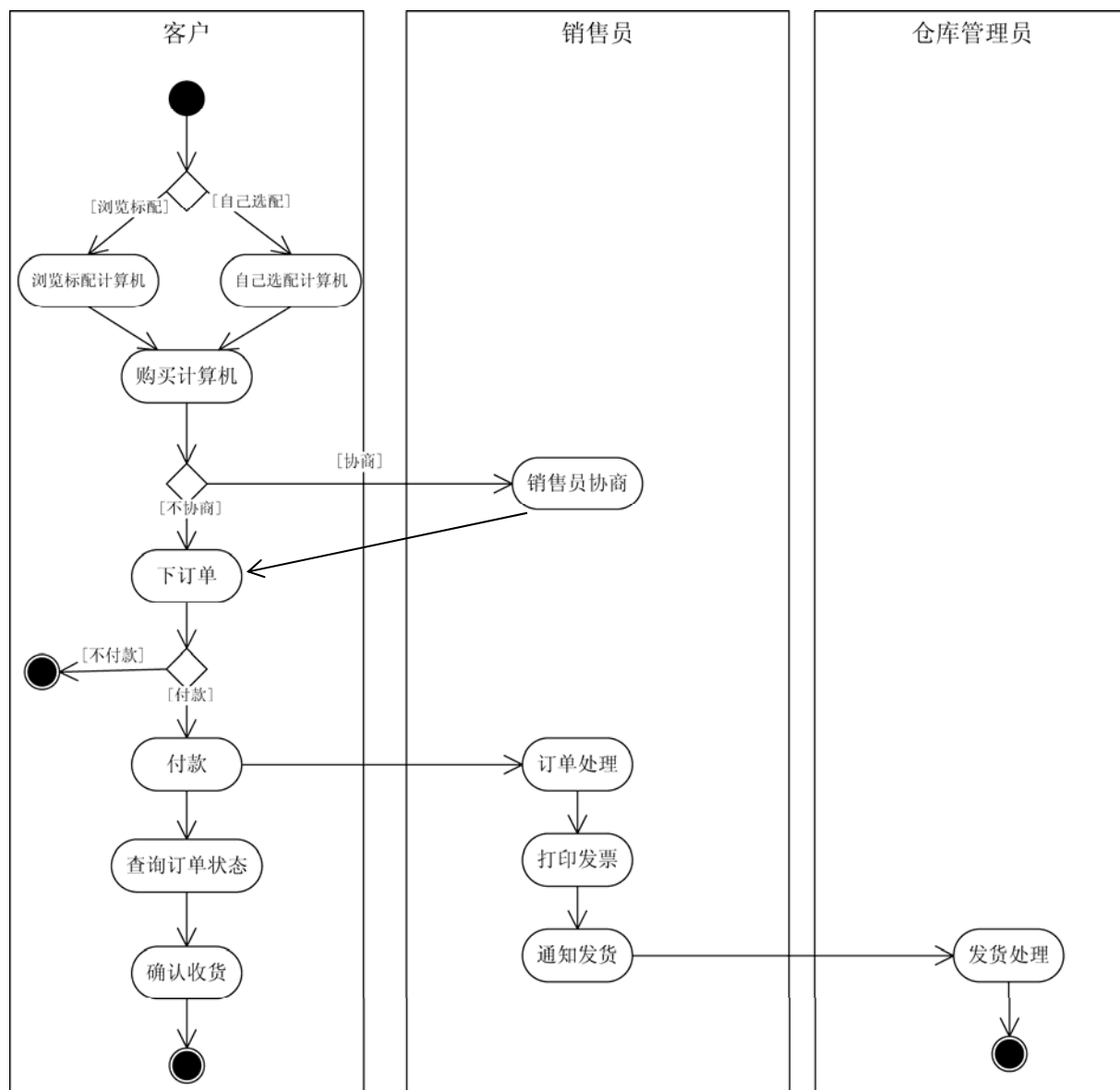
- 计算机厂商准备开发一个“网上计算机销售系统”，以方便客户通过网络购买计算机。客户可以通过Web页面登录进入系统、查看、选择、购买标准配置计算机，也可以选择计算机的配置，购买自己配置的计算机，可配置的选配件（如内存）显示在一个可供选择的表中。根据用户选择的每个配件，系统可以计算出计算机的价格。客户可以选择在线购买计算机，也可要求销售员在发出订单之前与自己联系，解释订单细节，协商价格等。
- 客户在准备发出订单时，必须在线填写运送和发票地址以及付款细节（微信、支付宝、银行卡），一旦订单被输入，系统会向客户发送一份确认邮件，并附上订单细节。在等待计算机到达过程中，客户可以在线查询订单的状态。
- 后端订单处理：销售员验证客户的信用和付款方式，向仓库请求所购的计算机，打印发票并请求仓库将计算机运送给客户。

上下文模型



所要开发的网上计算机销售系统仅负责计算机的销售部分
销售商品的发货由仓库管理系统负责
商品的运输由物流管理系统负责
三个系统共享数据

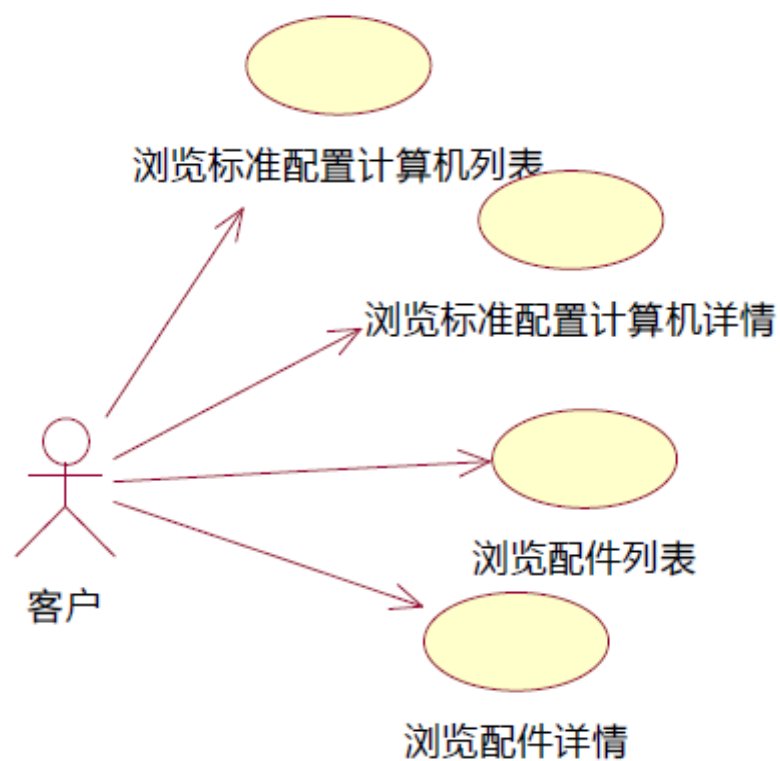
活动图（业务流程）



1. 起始点
2. 结束点
3. 活动
4. 对象
5. 迁移
6. 分支
7. 分叉与会合
8. 分区

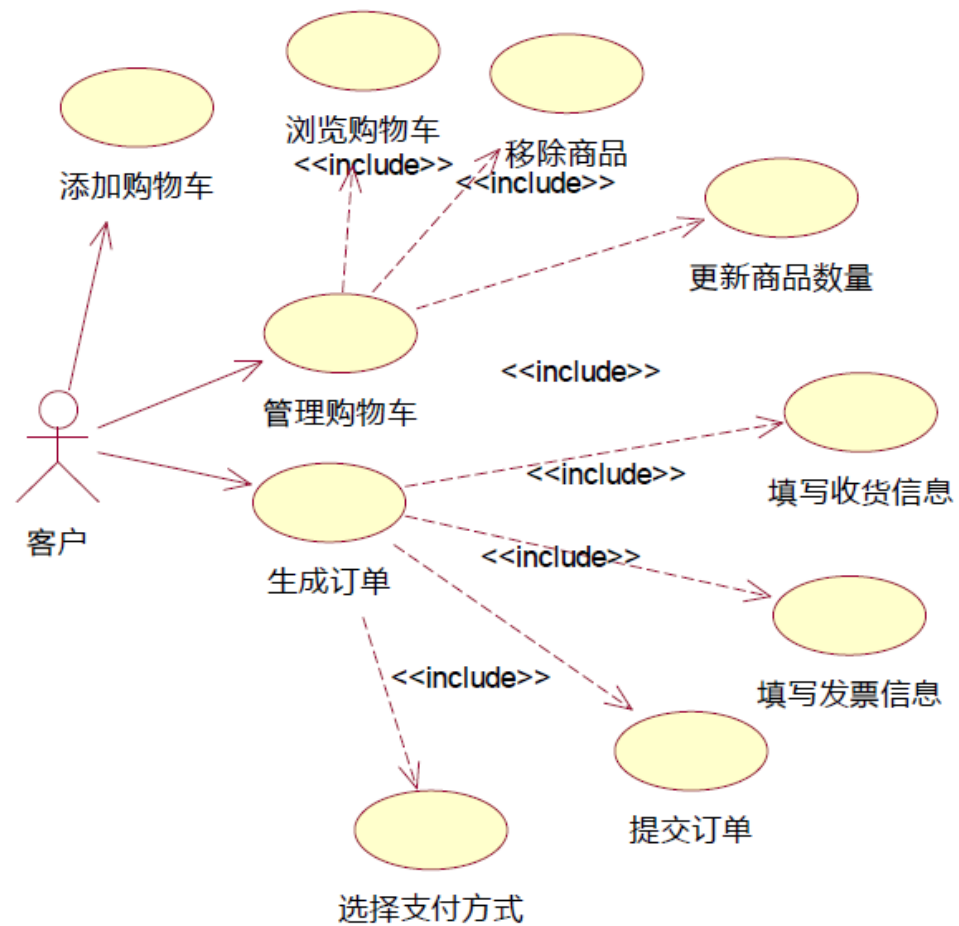
用例图 (Use Case Diagram)

(1) 浏览计算机信息



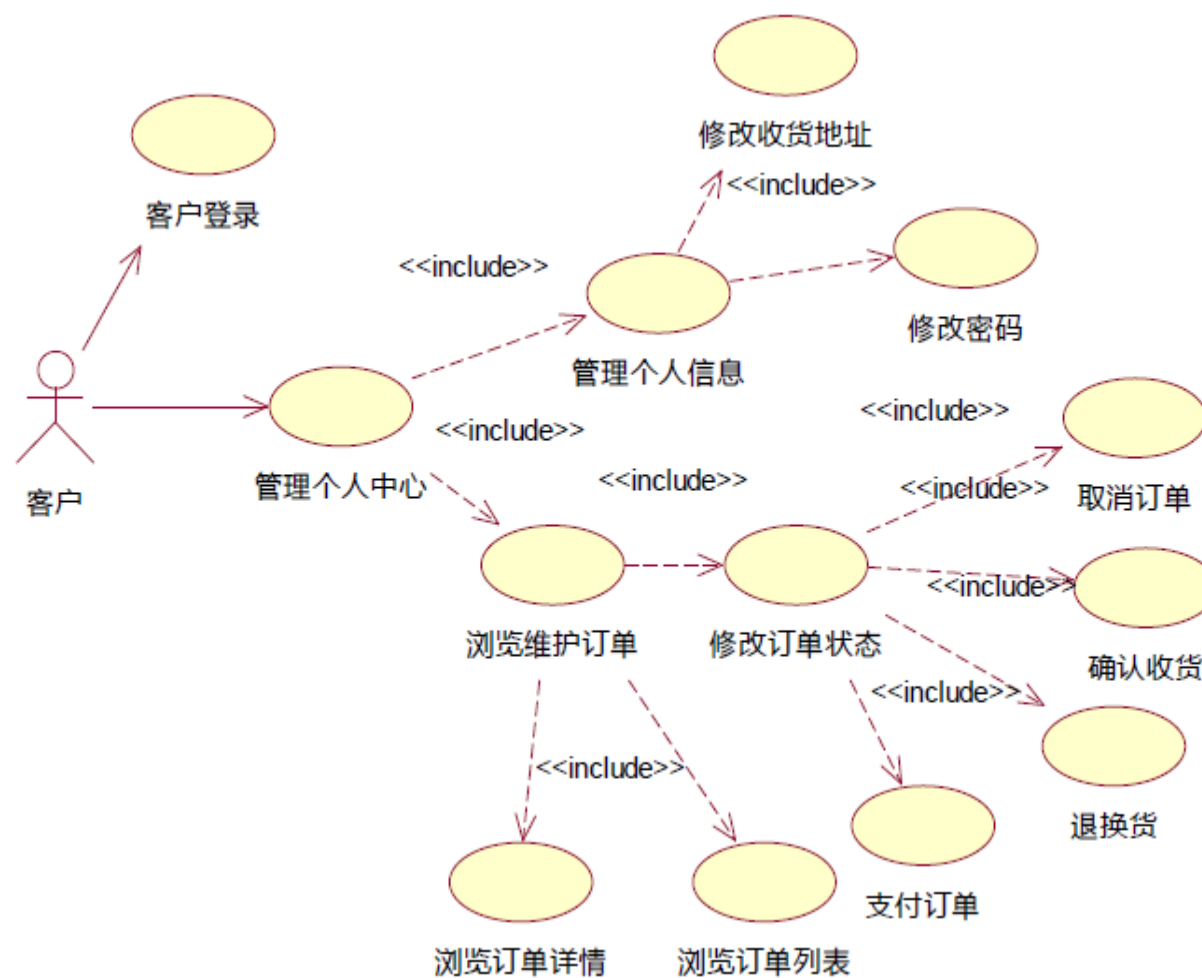
用例图 (Use Case Diagram)

• (2) 购买计算机



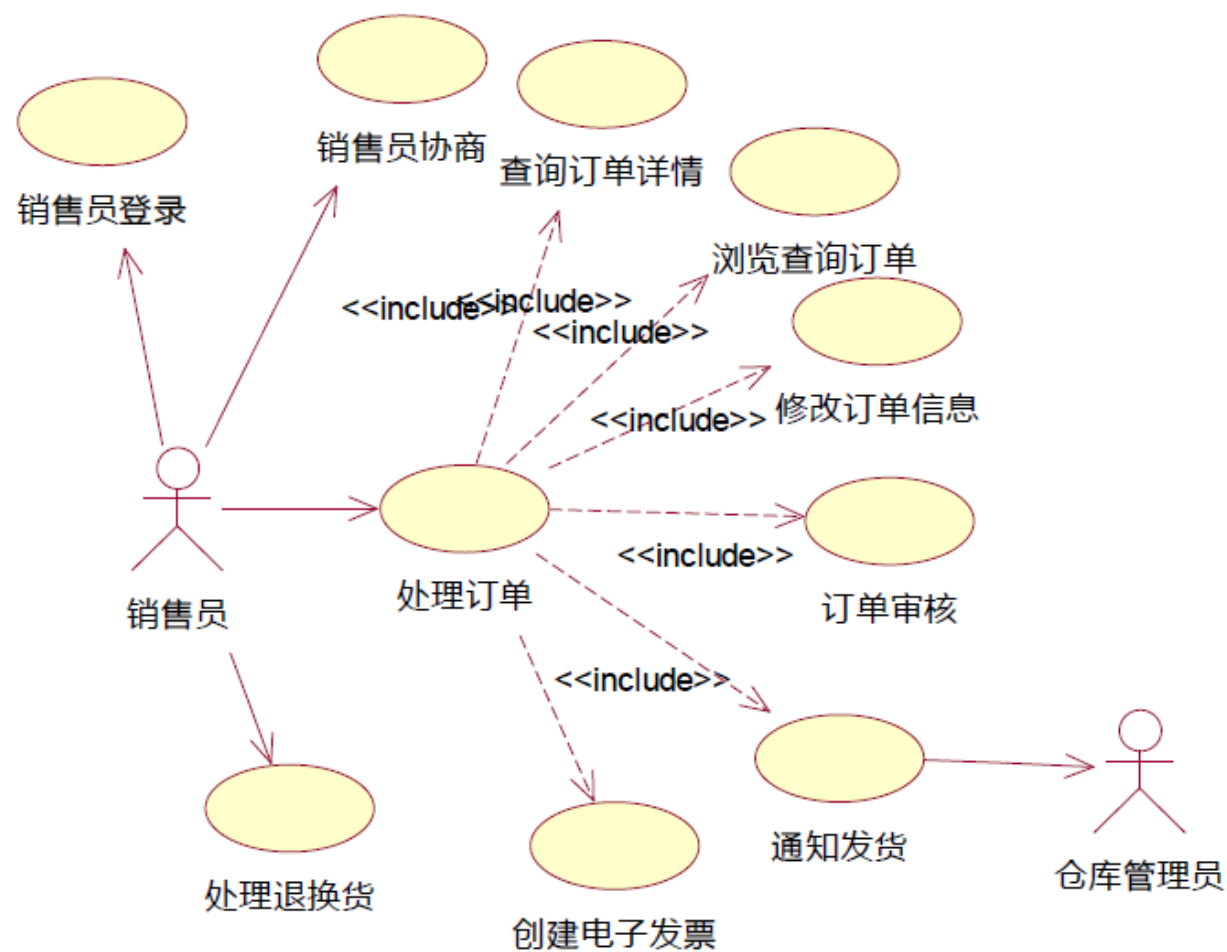
用例图 (Use Case Diagram)

• (3) 客户个人中心



用例图 (Use Case Diagram)

• (4) 销售订单管理



用例之间的关系

(1)包含关系：一个基本用例可以包含其他用例具有的行为。执行基本用例时，每次都必须执行被包含用例，被包含用例也可单独执行。例如，填写电子表格功能，在网上预订的过程中使用，不管如何处理网上预订用例，总要运行填写电子表格用例，因此具有包含关系。



用例之间的关系

(2) 扩展关系：当某个基本用例需要附加一个用例来扩展其原有的功能时，附加的扩展用例与原有的基本用例之间的关系就体现为扩展关系。

- 扩展关系可以有控制条件，当用例实例执行到达一个控制点时，控制条件决定是否执行扩展。
- 例如，汽车租赁系统中，顺利归还汽车，那么执行基本用例“还车”即可；但是如果超过了还车时间或汽车受损，即特定条件为超时或损坏，则执行扩展用例“缴纳罚款”。



基本用例 <<extend>> 扩展用例



还车 <<extend>> 缴纳罚款

用例之间的关系

(3) 泛化关系：其代表一般与特殊的关系，类似于继承，子用例表示父用例的特殊形式，子用例继承父用例的行为，也可以增加新的行为或覆盖父用例行为。例如，“网上预订”和“电话预订”都具有“预订”这个基本的行为，但是又增加了属于自己特有的行为特征。



类图 (Class Diagram)



- (1) 类名：类名是访问类的索引
- (2) 属性：属性用来描述该类的对象所具有的特征。
- (3) 操作：操作描述对数据的具体处理方法。

类图 (Class Diagram)

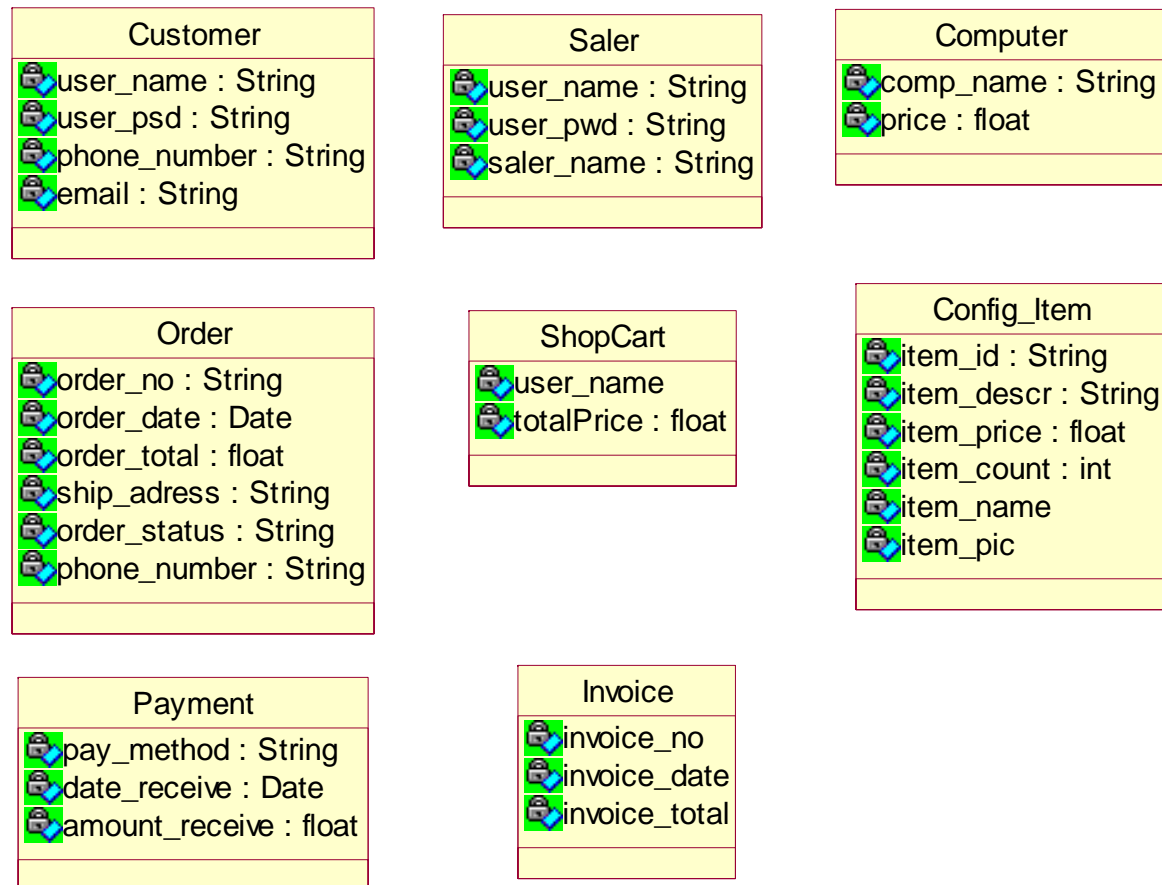


对象模型中的类包括以下三种。

- (1) 实体类**：其是问题域中的核心类，一般是从客观世界中的实体对象归纳和抽象出来的，用于长期保存系统中的信息，以及提供针对这些信息的相关处理行为。
- (2) 边界类**：其是从系统和外界进行交互的对象中归纳、抽象出来的，它是系统类的对象和系统外的执行者的连接媒介，外界的消息只有通过边界类的对象实例才能发送给系统。
- (3) 控制类**：是实体类和边界类之间的“润滑剂”，用于协调系统边界类和实体类之间的交互。

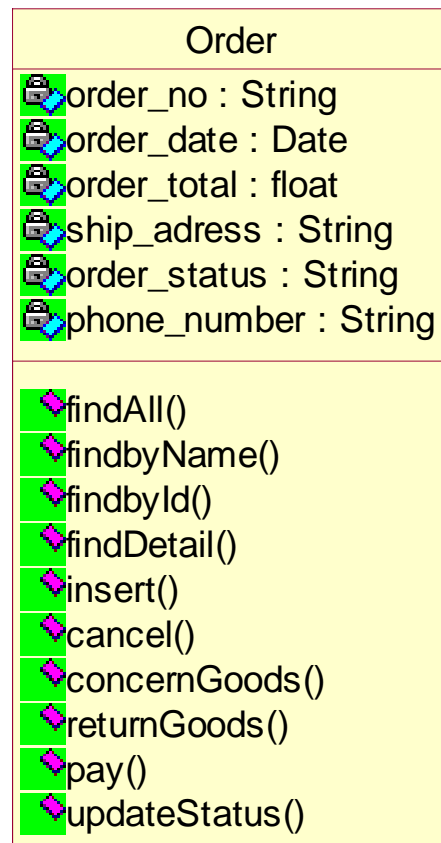
类图 (Class Diagram)

• 计算机销售系统中的类属性



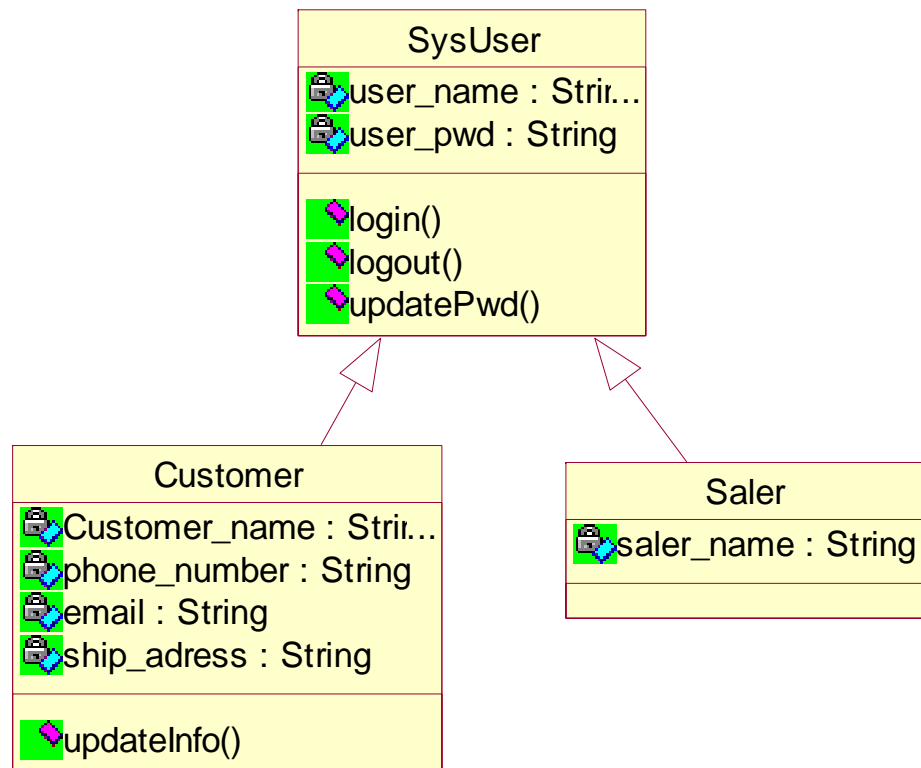
类图 (Class Diagram)

• 计算机销售系统中**Order (订单)**类的类图



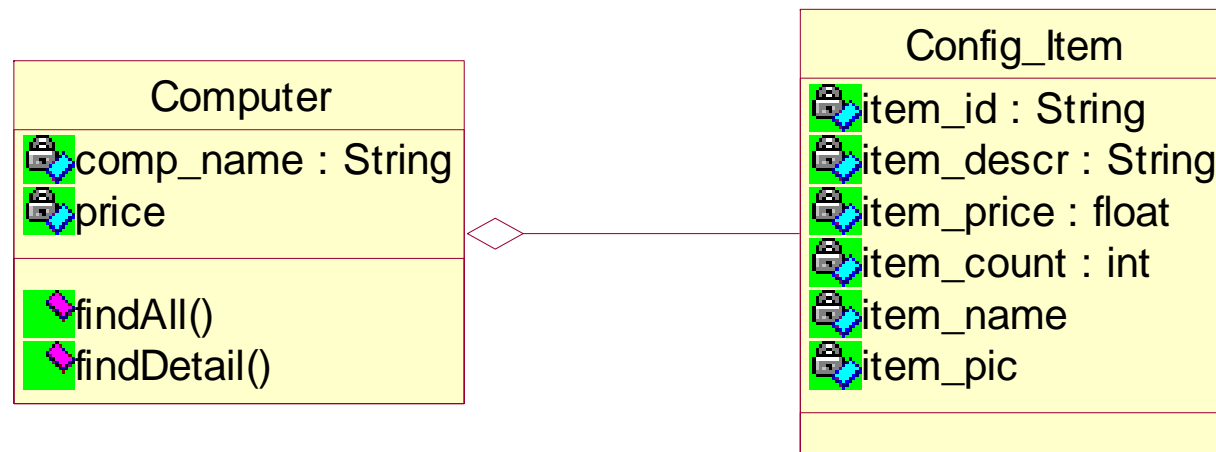
类图 (Class Diagram)

- 识别泛化关系



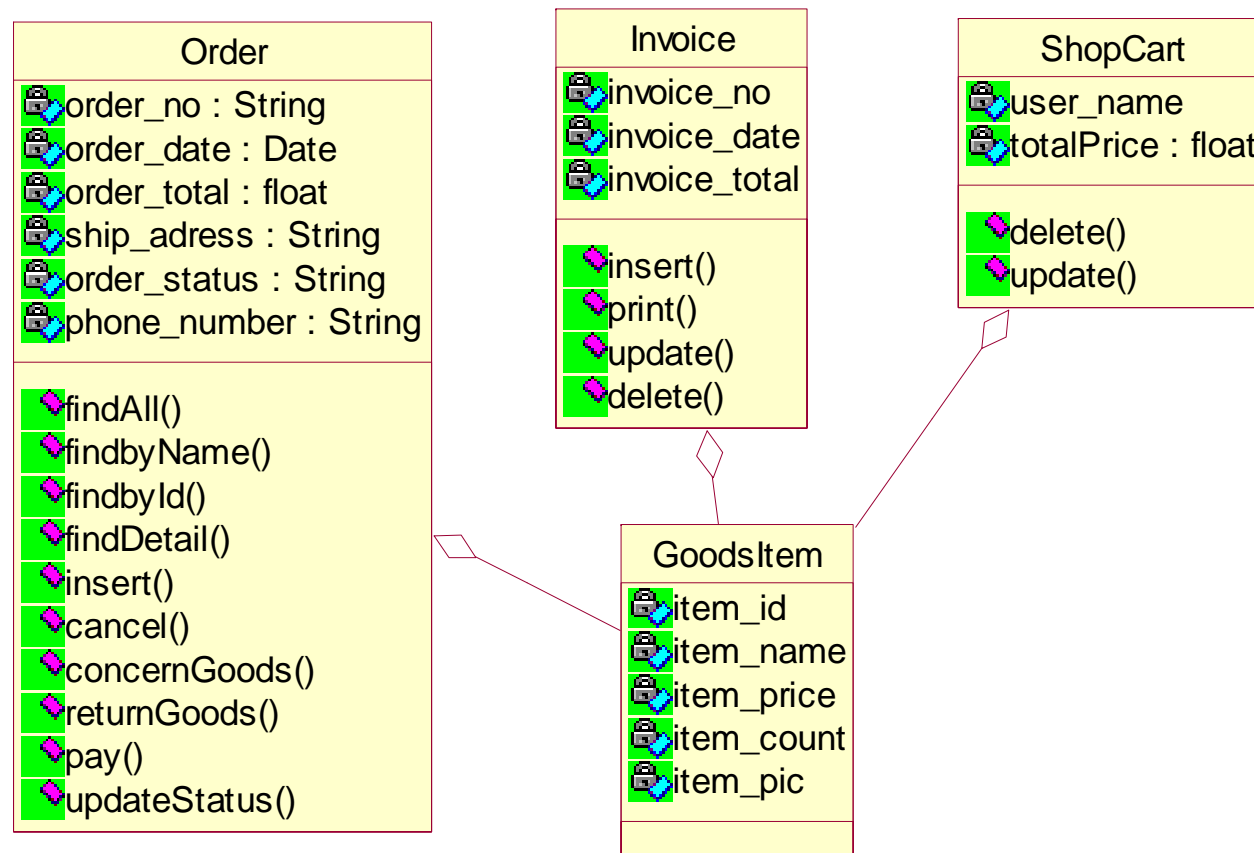
类图 (Class Diagram)

- 识别整体-部分关系

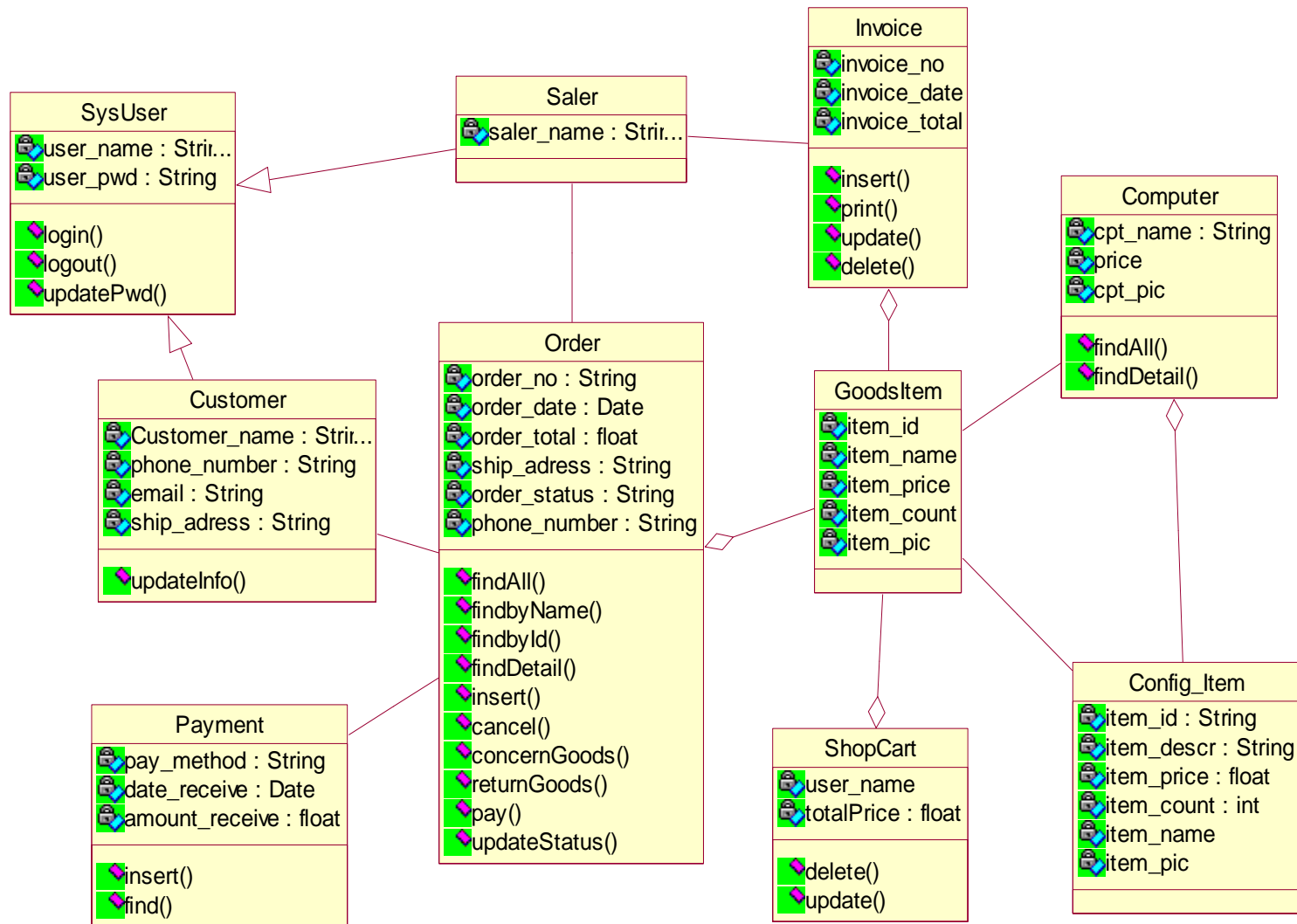


类图 (Class Diagram)

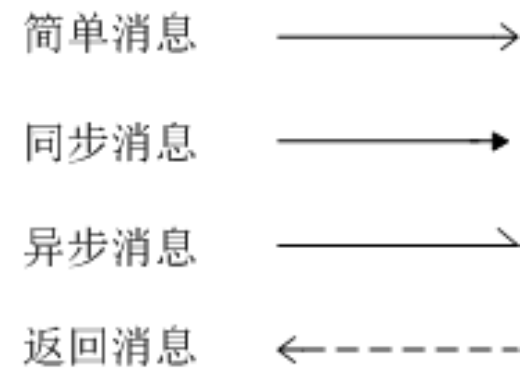
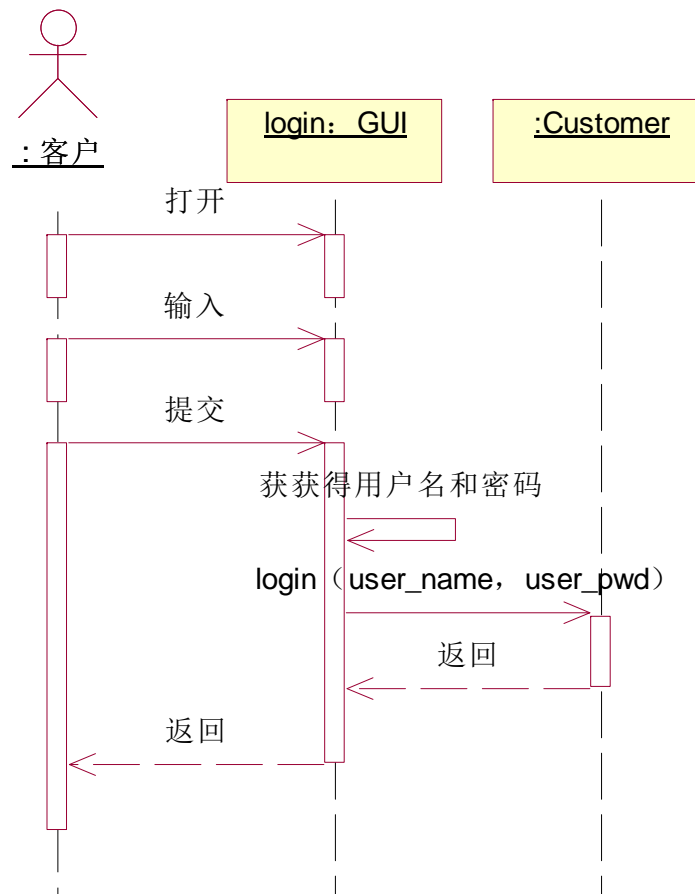
• 识别整体-部分关系



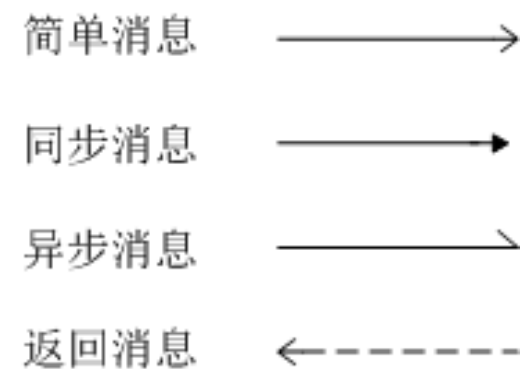
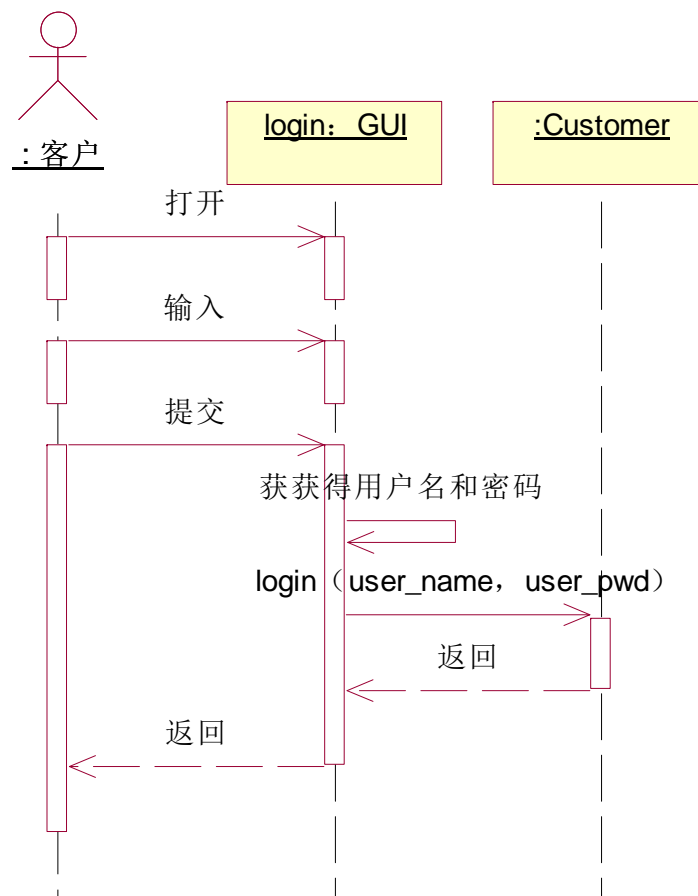
类图 (Class Diagram)



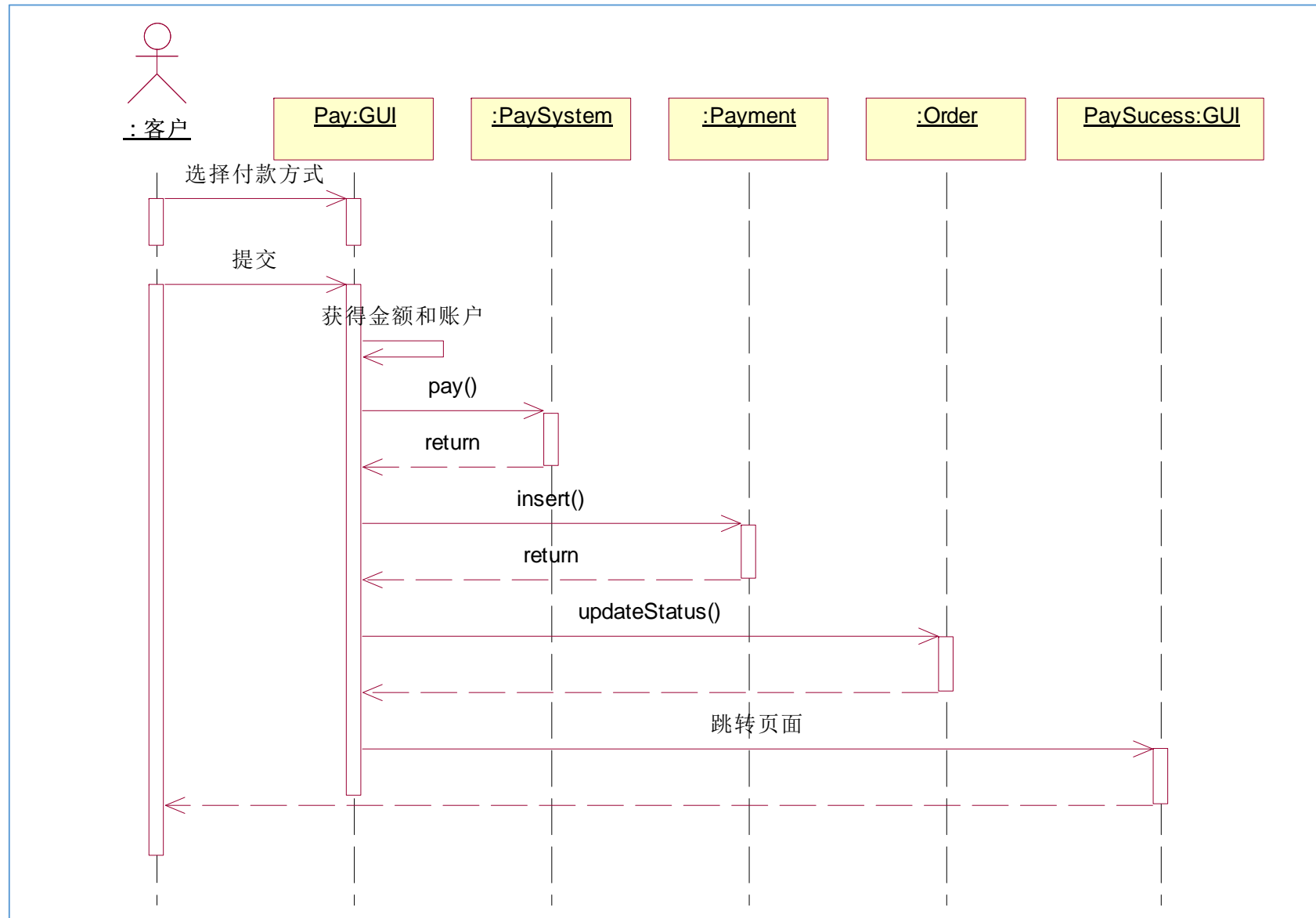
顺序图 (Sequence Diagram)



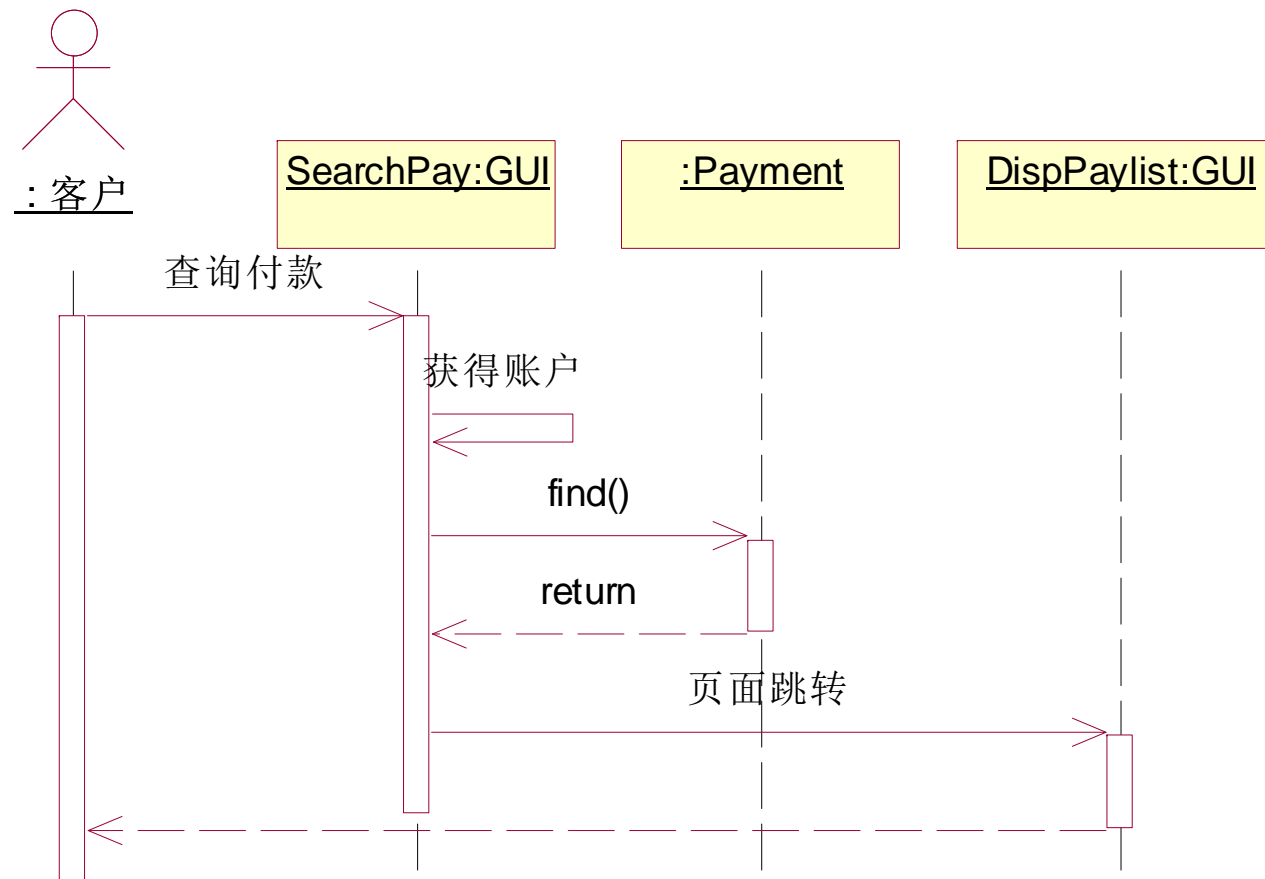
顺序图 (Sequence Diagram)



顺序图 (Sequence Diagram)

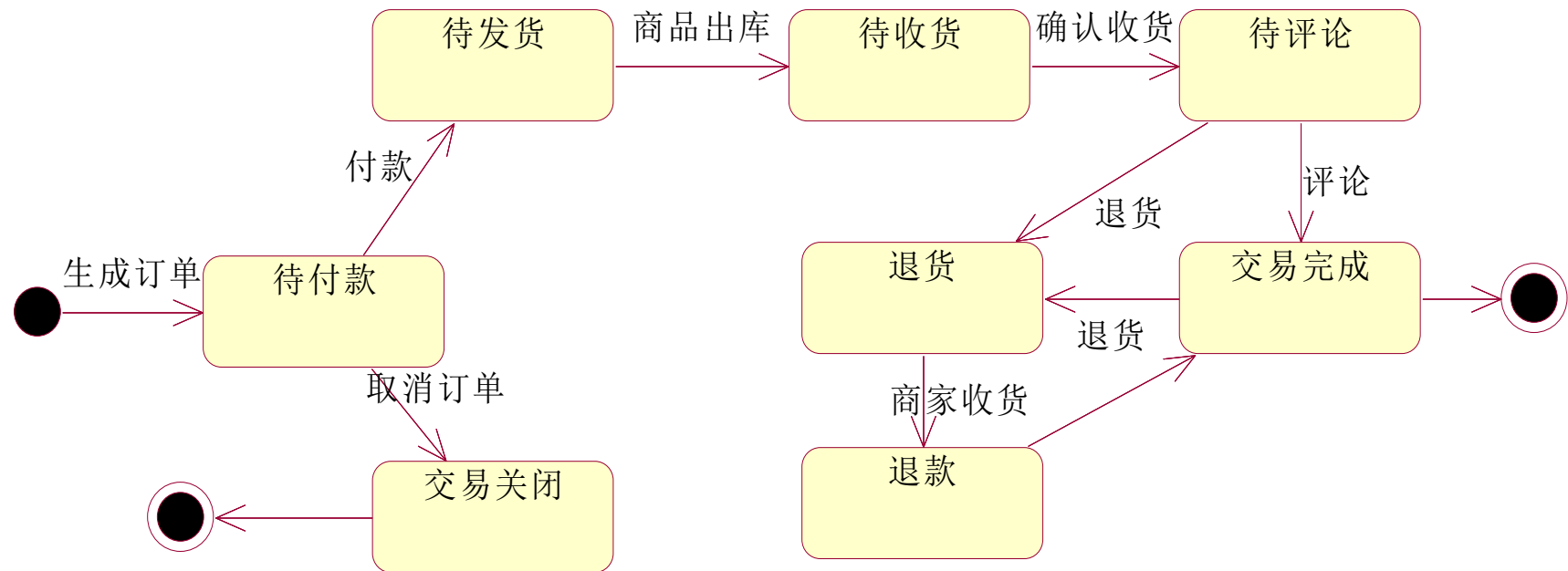


顺序图 (Sequence Diagram)



检查计算机销售系统的类图中是否有Payment类，同时该类是否具有find()方法，检查中发现在计算机销售系统的用例图中不存在“查询付款”用例，因此需要在客户个人中心的用例图中补充该用例

状态图 (Sequence Diagram)



在状态图中触发状态转移的事件大部分都应该存在于用例中，在上图中“商品出库”、“商家收货”因超出边界而不做处理，“评论”事件引起订单状态从“未评论”到“交易完成”的迁移，而该用例却未在客户个人中心用例图中出现，所以应对该用例予以补充。



谢谢！