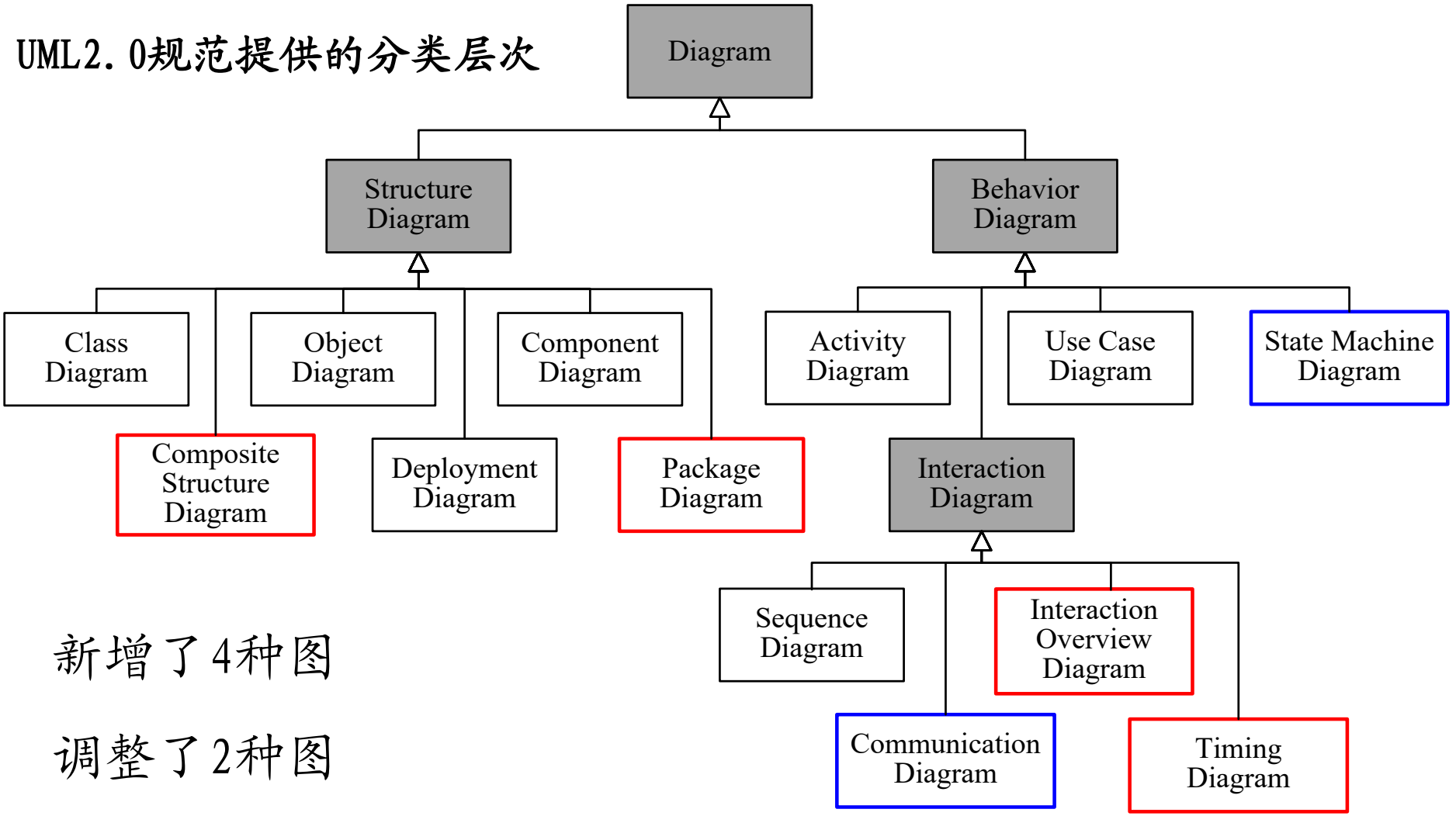
南京大学
NANJING UNIVERSITY

UML模型介绍

张天
软件工程组
ztluck@nju.edu.cn
2022年秋季

UML 2.0支持13种图

UML2.0规范提供的分类层次



新增了4种图

调整了2种图

常用的中译名

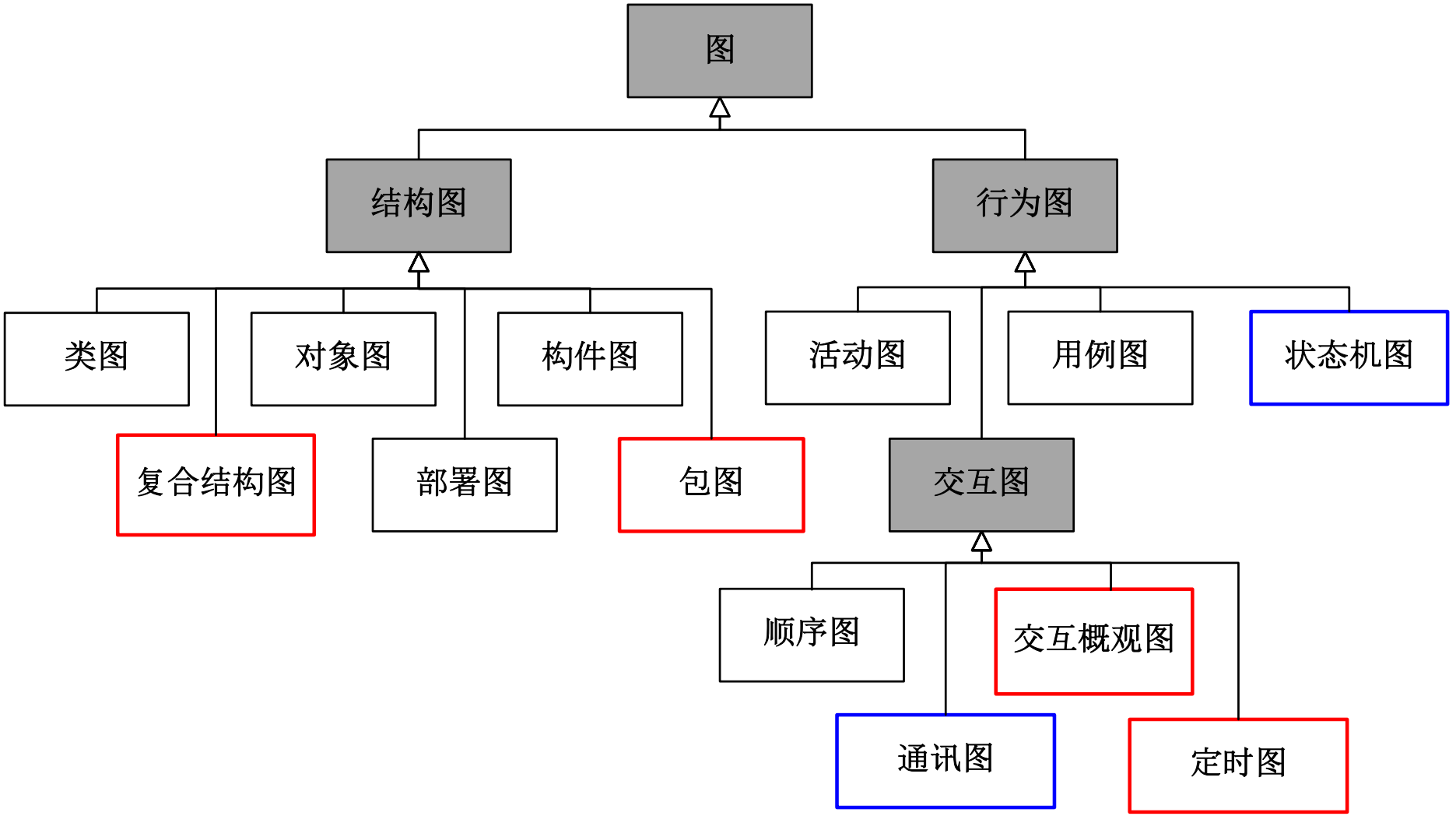


图
结构图
行为图
类图
对象图
构件图
活动图
用例图
状态机图
复合结构图
部署图
包图
交互图
顺序图
交互概观图
通讯图
定时图

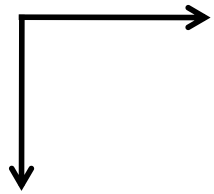
UML2.0增改的图

- 相对于UML1.5增加了4种图
  - 复合结构图（Composite Structure Diagram）
  - 包图（Package Diagram）
  - 交互概观图（Interaction Overview Diagram）
  - 定时图（Timing Diagram）
    注：包图在UML1.x中并不是一种正式的图
- 对2种图重新命名
  - 原来的协作图(Collaboration Diagrams)改名为通讯图(Communication Diagrams)
  - 原来的状态图(Statechart Diagrams)改名为状态机图(State Machine Diagrams)

从实用的角度再看13种图

- 功能
  - 用例图
- 结构
  - 类图
  - 对象图
  - 包图
  - 复合结构图
- 行为
  - 状态机图
  - 顺序图
  - 活动图
  - 通讯图
  - 交互概观图
- 实现
  - 构件图
  - 部署图
- 时间
  - 定时图

介绍顺序：



用例图（Use Case Diagram）

用例图描述系统外部的执行者与系统的用例之间的某种联系。

- **用例**：是指对系统提供的功能（或称系统的用途）的一种描述；
- **执行者**：是那些可能使用这些用例的人或外部系统；
- **联系**：用例和执行者之间的联系描述了“谁使用哪个用例”。

用例（Use Case）

从本质上将，一个用例是用户与计算机之间为达到某个目的的一次典型交互作用：

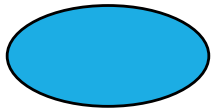
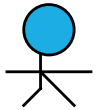
- 用例描述了用户提出的一些可见的需求；
- 用例可大可小；
- 用例对应一个具体的用户目标

用例图（Use Case Diagram）

- 用例图着重于从系统外部执行者的角度来描述系统需要提供哪些功能，并且指明了这些功能的执行者是谁；
- 用例图在UML方法中占有十分重要的地位，人们甚至称UML是一种用例图驱动的开发方法。

用例图的符号

用例图中的图符:

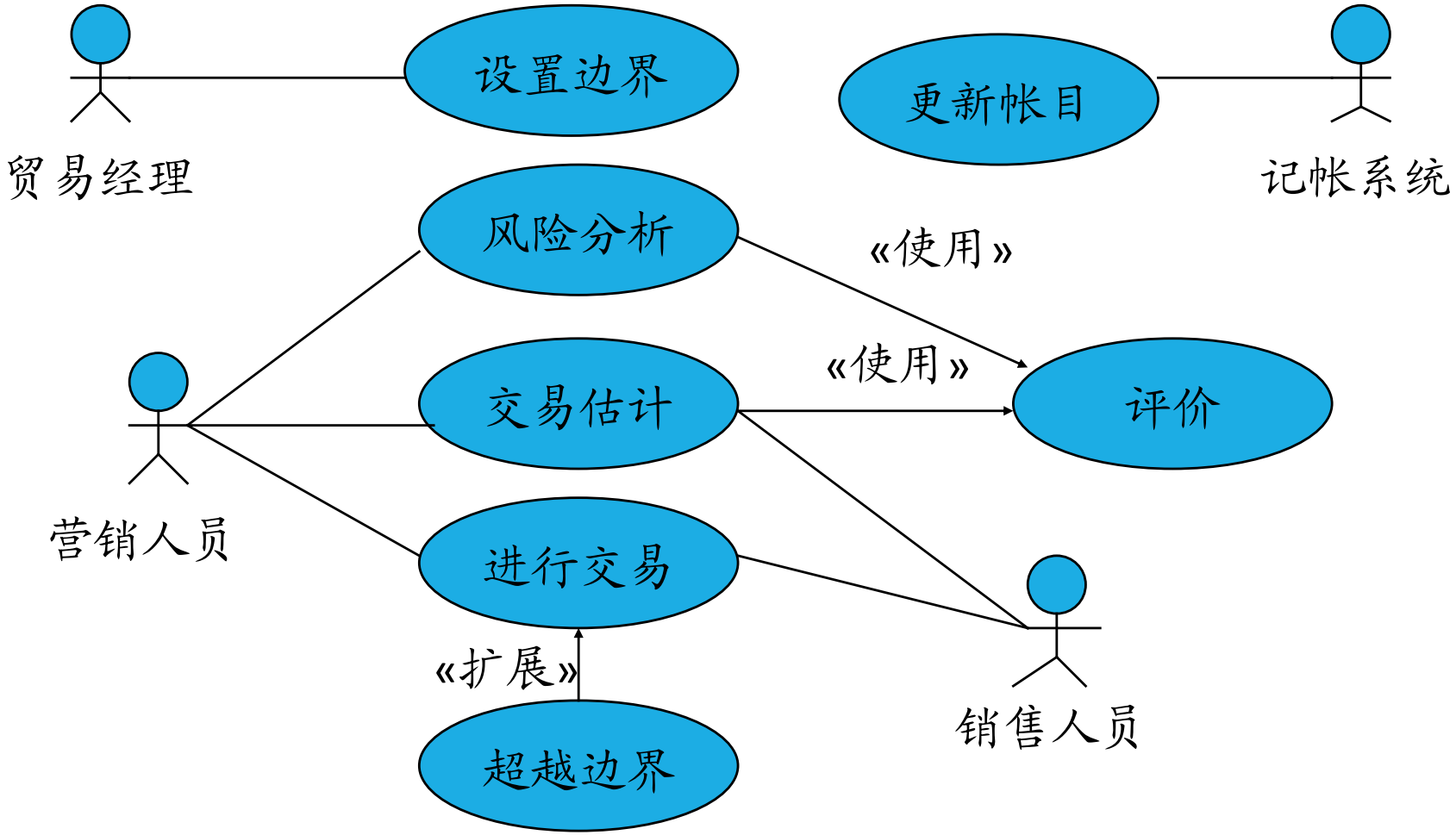
-  用例
-  执行者
-  系统：用于界定系统功能范围，描述该系统功能的用例都置于其中，而描述外部实体的执行者都置于其外。
- 关联：连接执行者和用例，表示执行者所代表的系统外部实体与该用例所描述的系统需求有关。

用例图的符号

用例图中的图符:

- «使用» 使用：由用例A连向用例B，表示用例A中使用了用例B中的行为或功能。
- «扩展» 扩展：由用例A连向用例B，表示用例B描述了一项基本需求，而用例A则描述了该基本需求的特殊情况。
-  注释体：对UML实体进行文字描述
- ------- 注释连接：将注释体与要描述的实体连接，说明该注释体是针对该实体所进行的描述。

用例图（Use Case Diagram）



基于用例的需求分析过程

- 1. 获取原始需求
- 2. 开发一个可以理解的需求
  - 2.1 识别参与者
  - 2.2 识别用例
  - 2.3 构建用例图
- 3 详细、完整地描述需求
  - 进行用例阐述
- 4 重构用例模型
  - 4.1 识别用例间的关系
  - 4.2 对用例进行组织和分包

获取原始需求：考勤卡应用程序

初次访谈记录

开发者：谁将使用这个应用程序？

客　　户：所有用它来记录可记帐以及不可记帐的工时的雇员

……

开发者：现在考勤卡应用程序是什么样的？

客　　户：每半个月就用一个Excel表格来记录。每个雇员都将通过他的表格填好，然后用电子邮件发给我。这个表格相当标准：纵向是收费项目代码，横向是日期。雇员可以在每个条目上填写说明。

开发者：这个收费项目代码可以从什么地方得到？

……

开发者：谁来管理收费项目代码？

客　　户：嗯，必要的时候由我来添加这个代码。而每个经理总会告诉他的下属应该填写什么。

……

用例图（Use Case Diagram）

用例模型的获取:

- 获取执行者
- 获取用例

获取执行者的启发规则

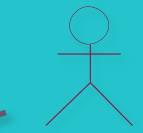
获取执行者:

- 谁使用系统的主要功能（主要使用者）？
- 谁需要系统支持他们的日常工作?
- 谁来维护、管理系统使其能正常工作（辅助使用者）？
- 系统需要控制哪些硬件？
- 系统需要与其他哪些系统交互？
- 对系统产生的结果感兴趣的是哪些人?

识别参与者：考勤卡系统

初次访谈记录

开发者：谁将使用这个应用程序？
客　　户：所有用它来记录可记帐以及不可记帐的工时的雇员



……
开发者：现在考勤卡应用程序是什么样的？
客　　户：每半个月就用一个Excel表格来记录。每个雇员都将通过他的表格填好，然后用电子邮件发给我。这个表格相当标准：纵向是收费项目代码，横向是日期。雇员可以在每个条目上填写说明。
开发者：这个收费项目代码可以从什么地方得到？

……
开发者：谁来管理收费项目代码？
客　　户：嗯，必要的时候由我（业务经理）来添加这个代码。而每个经理总会告诉他的下属应该填写什么。
……

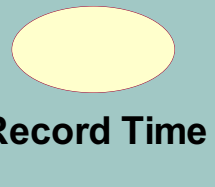
获取用例的启发规则

获取用例:

- 执行者要求系统提供哪些功能?
- 执行者需要读、产生、删除、修改或存储系统中的信息有哪些类型?
- 必须提醒执行者的系统事件有哪些?
- 执行者必须提醒系统事件有哪些? 怎样把这些事件表示成用例中的功能?

识别用例：考勤卡系统

初次访谈记录



开发者：谁将使用这个应用程序？

客　　户：所有用它来**记录可记帐以及不可记帐的工时**的**雇员**

……

开发者：现在考勤卡应用程序是什么样的？

客　　户：每半个月就用一个Excel表格来记录。每个雇员都将通过他的表格填好，然后用电子邮件发给我。这个表格相当标准：纵向是收费项目代码，横向是日期。雇员可以在每个条目上填写说明。

开发者：这个收费项目代码可以从什么地方得到？

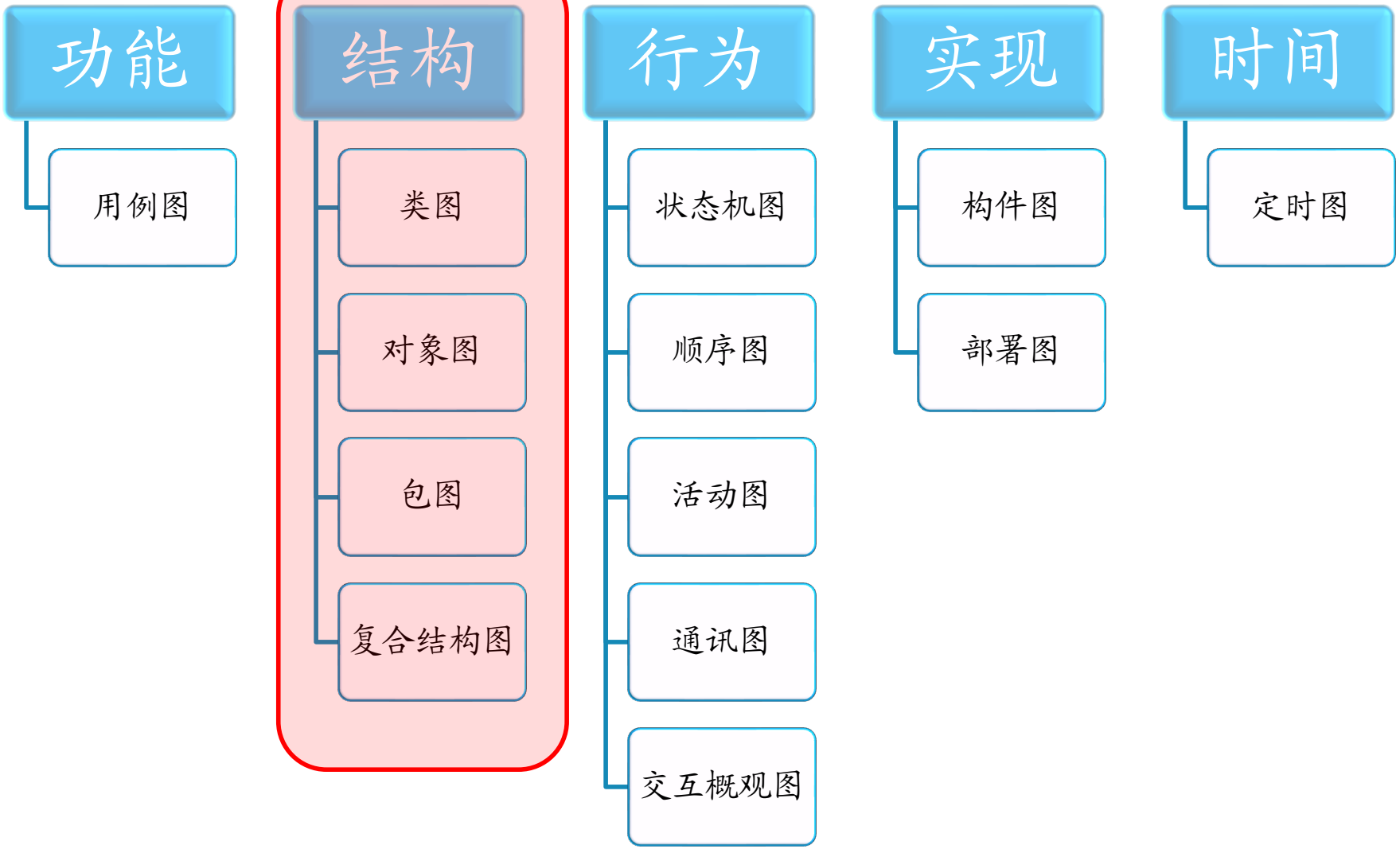
……

开发者：谁来**管理收费项目代码**？

客　　户：嗯，必要的时候由我**（业务经理）**来添加这个代码。而每个经理总会告诉他的下属应该填写什么。

……

UML2.0的结构图


功能
用例图
结构
类图
对象图
包图
复合结构图
行为
状态机图
顺序图
活动图
通讯图
交互概观图
实现
构件图
部署图
时间
定时图

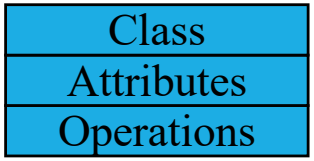
类图（Class Diagram）

- 在面向对象的建模技术中，类、对象和它们之间的关系是最基本的建模元素。对于一个想要描述的系统，其类模型、对象模型以及它们之间的关系揭示了系统的结构。
- 类图描述了系统中的类及其相互之间的各种关系，其本质反映了系统中包含的各种对象的类型以及对象间的各种静态关系（关联，子类型）。

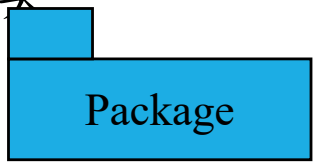
类图（Class Diagram）

类图中的图符:

- 类：表示一个类，其中第一栏是类名，第二栏是类的属性，第三栏是类的操作。



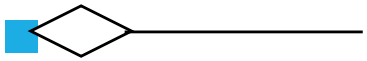
- 包：包是一种分组机制，表示一个类图集合。



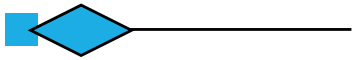
- 关联：用于表示类的对象之间的关系。其特殊形式有组成关联和聚集关联。

类图（Class Diagram）

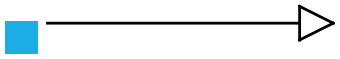
类图中的图符:



聚集关联：用于表示类的对象之间的关系是整体与部分的关系。



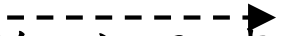
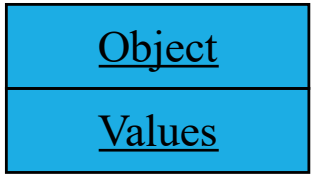
组成关联：用于表示类的对象之间的关系：整体拥有各部分，部分与整体共存，如整体不存在了，部分也会随之消失。



泛化关联：泛化关系（继承关系）定义了类和包间的一般元素和特殊元素之间的分类关系。

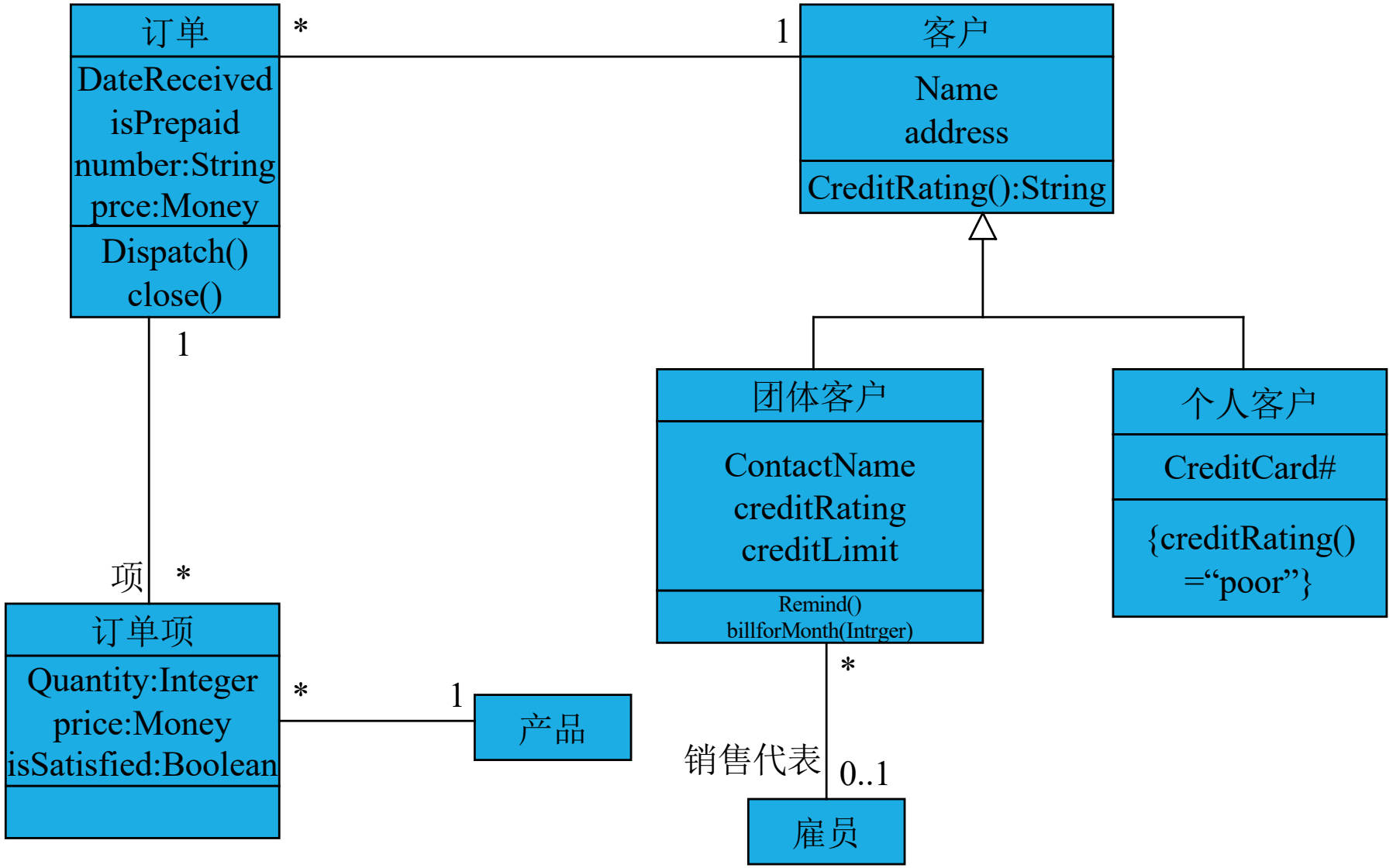
类图（Class Diagram）

类图中的图符：

- 
  依赖关系：有两个类或包元素X、Y，修改元素X的定义可能会引起对另一个元素Y的定义的修改，则称元素Y依赖于元素X。
- 

  对象：类的一个实例。
- 
  链接：用于表示对象间的关联关系的一个实例。

类图（Class Diagram）



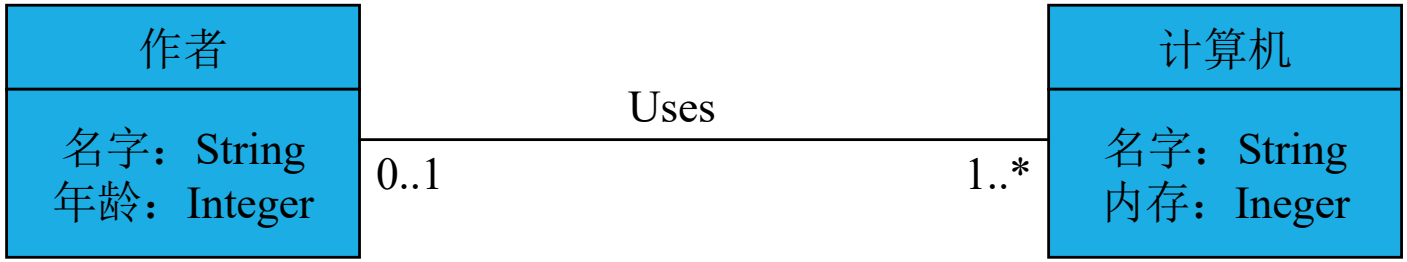
对象图（Object Diagram）

对象图

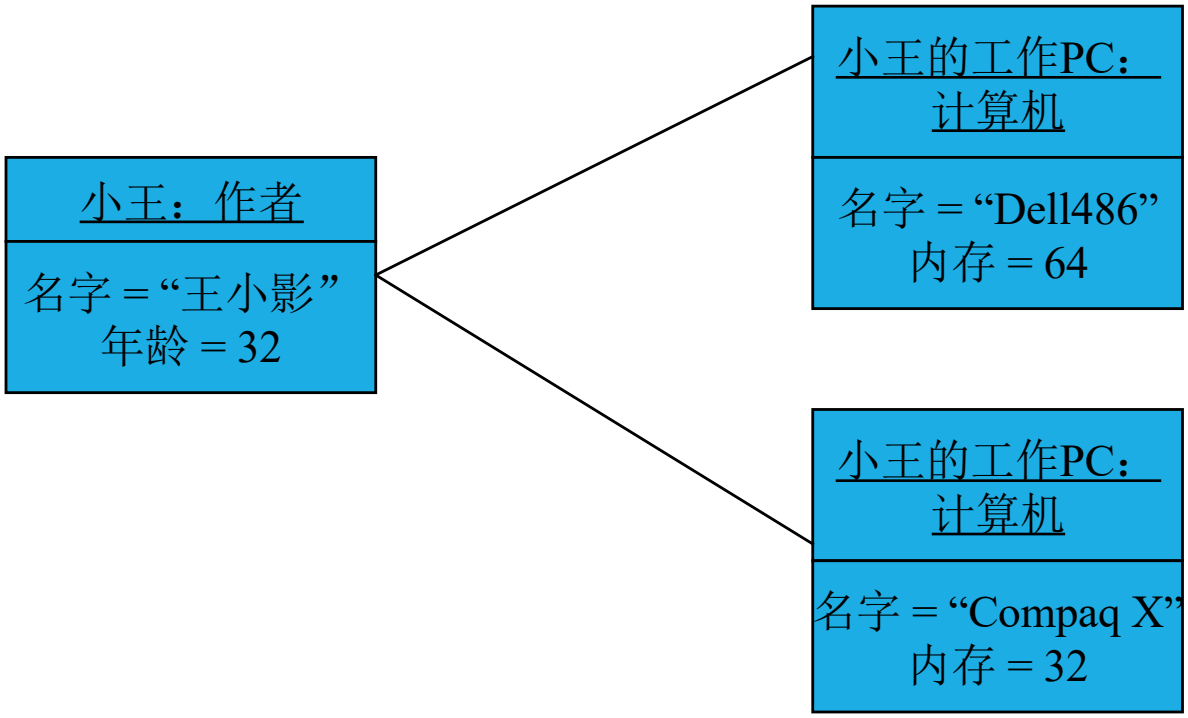
- 对象图是类图的一种变形。除了在对象名下面要加下划线以外，对象图中所使用的符号与类图基本相同。
- 对象图是类图的一种实例化。一张对象图表示的是与其对应的类图的一个具体实例，即系统在某一时期或者某一特定时刻可能存在的具体对象实例以及它们相互之间的具体关系。

对象图（Object Diagram）

类图



对象图



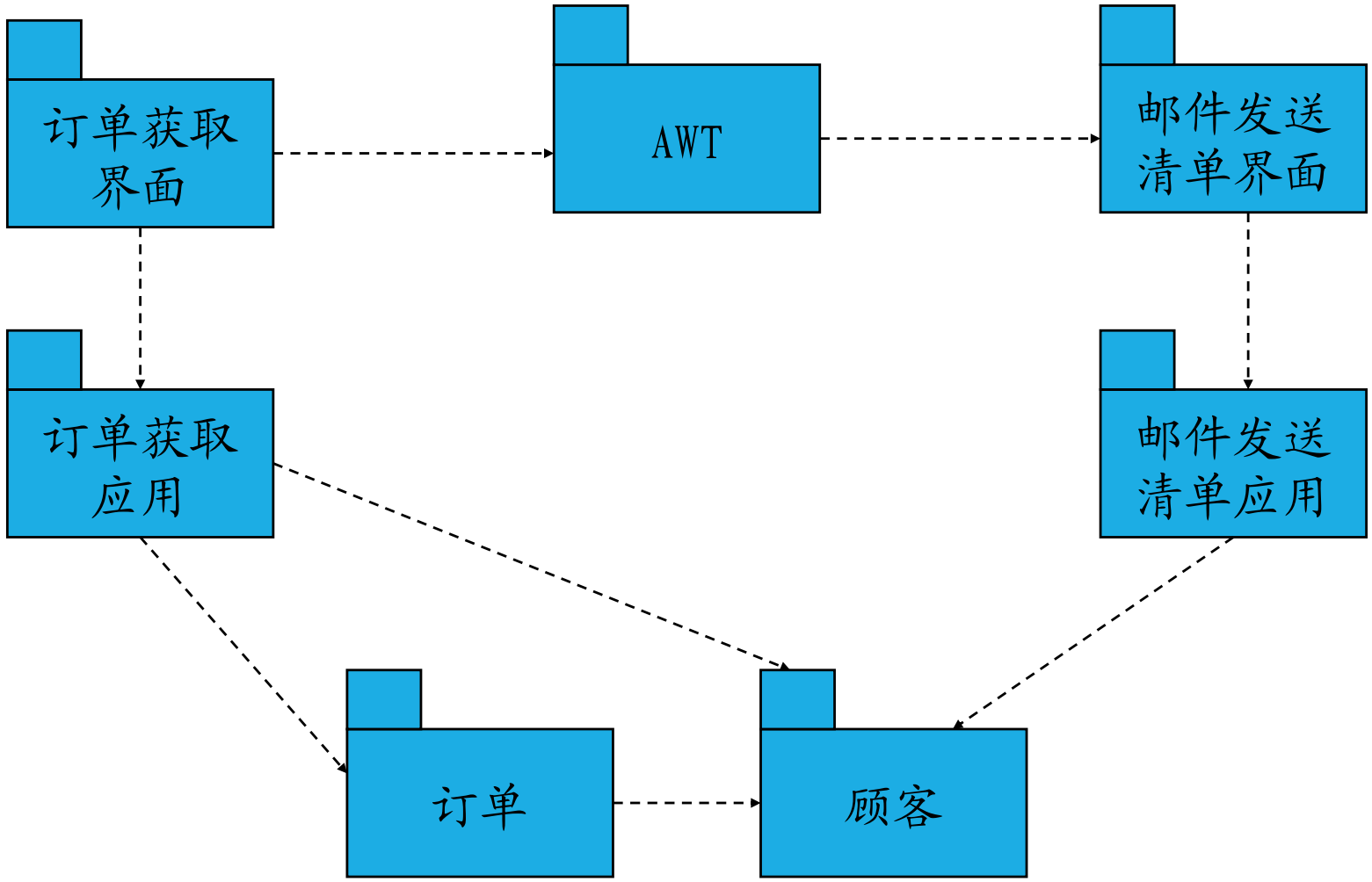
对象图（Object Diagram）

- 对象图并不象类图那样具有重要的地位，但是利用它可以帮助我们通过具体的实例分析，更具体直观地了解复杂系统类图的丰富内涵。
- 对象图还常常被用作通讯图的一部分，用以展示一组对象实例之间的动态协作关系。

包图（Package Diagram）

- 包是类的集合。
- 包图所显示的是类的包以及这些包之间的依赖关系。
- 如果两个包中的任意两个类之间存在依赖关系，则这两个包之间存在依赖关系。
- 包的依赖是不传递的。

包图（Package Diagram）



包图（Package Diagram）

何时使用包图:

- 在大项目中，包图是一种重要工具（有专家建议，只要你不能将整个系统的类图压缩到一张A4纸上，你就应该使用包图）；
- 依赖产生耦合，应该尽量将依赖性减少到最低程度；
- 包的概念对测试也是特别有用的。

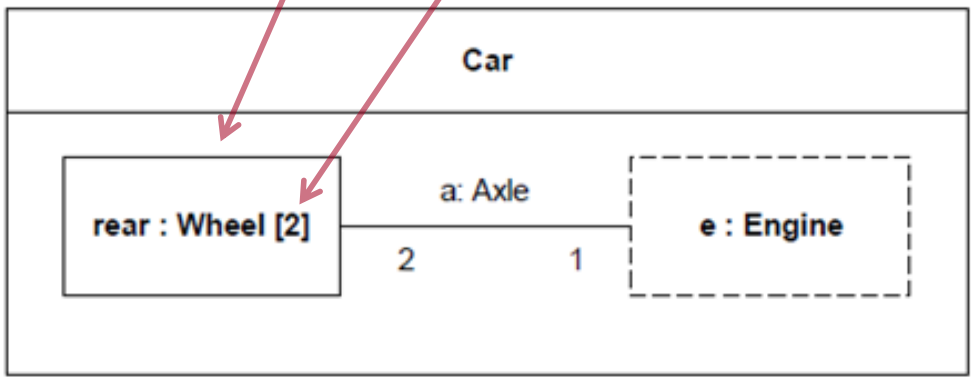
复合结构图（Composite Structure Diagram）

- UML2.0中一个重要的新特征是可以通过复合结构图把一个类层次地分解成一个内部结构
- 复合结构和包的区别
  - 包主要反映了编译期的分组关系
  - 复合结构则体现了运行期的分组关系

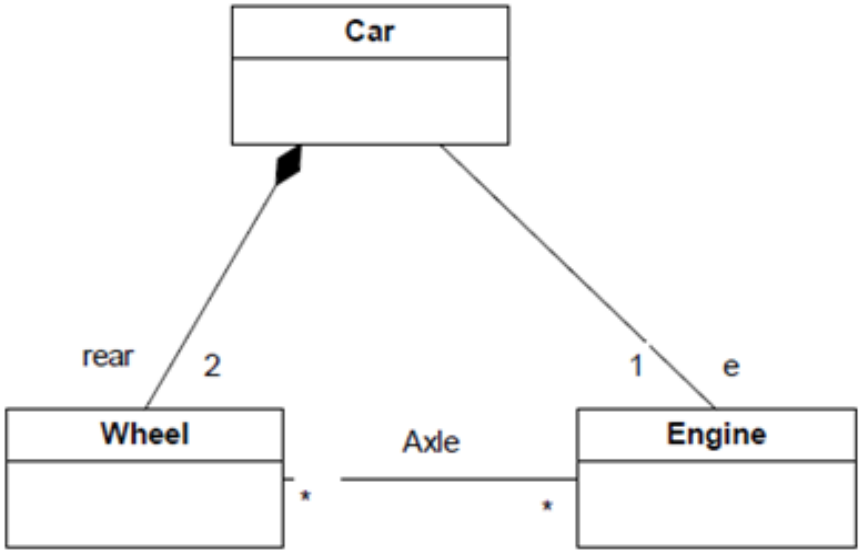
复合结构及语义等价体

实线框表示拥有关系

重数



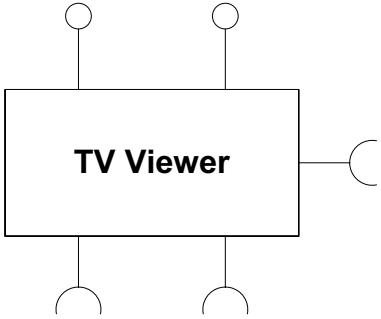
Car的复合结构

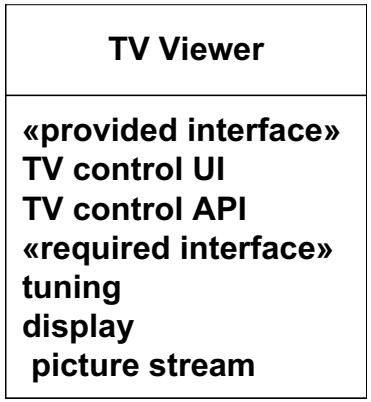


语义等价类图

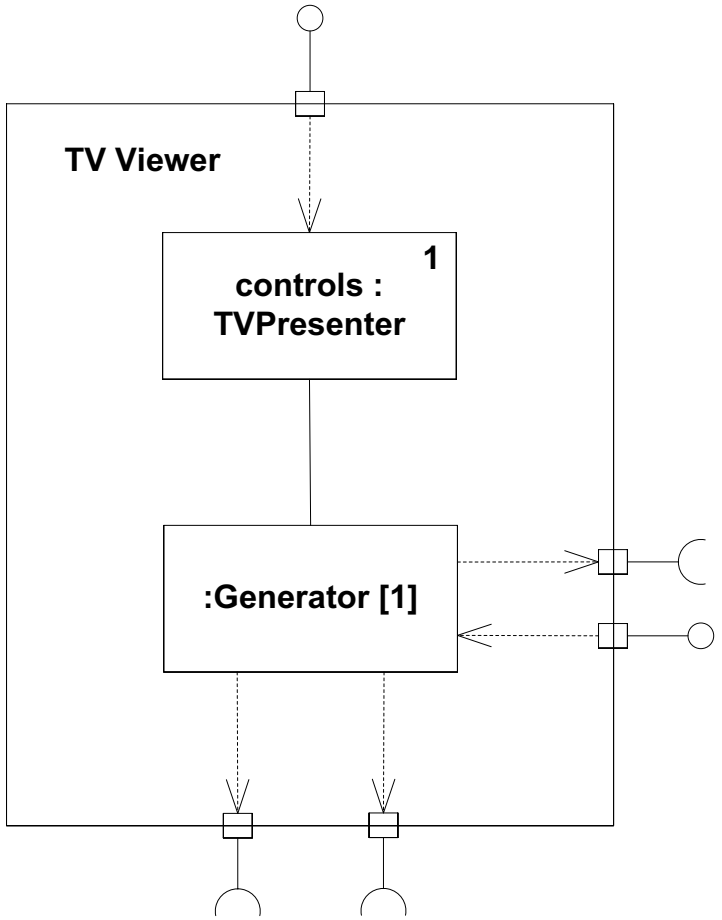
《UML Distilled》上的示例

从接口的角度描述

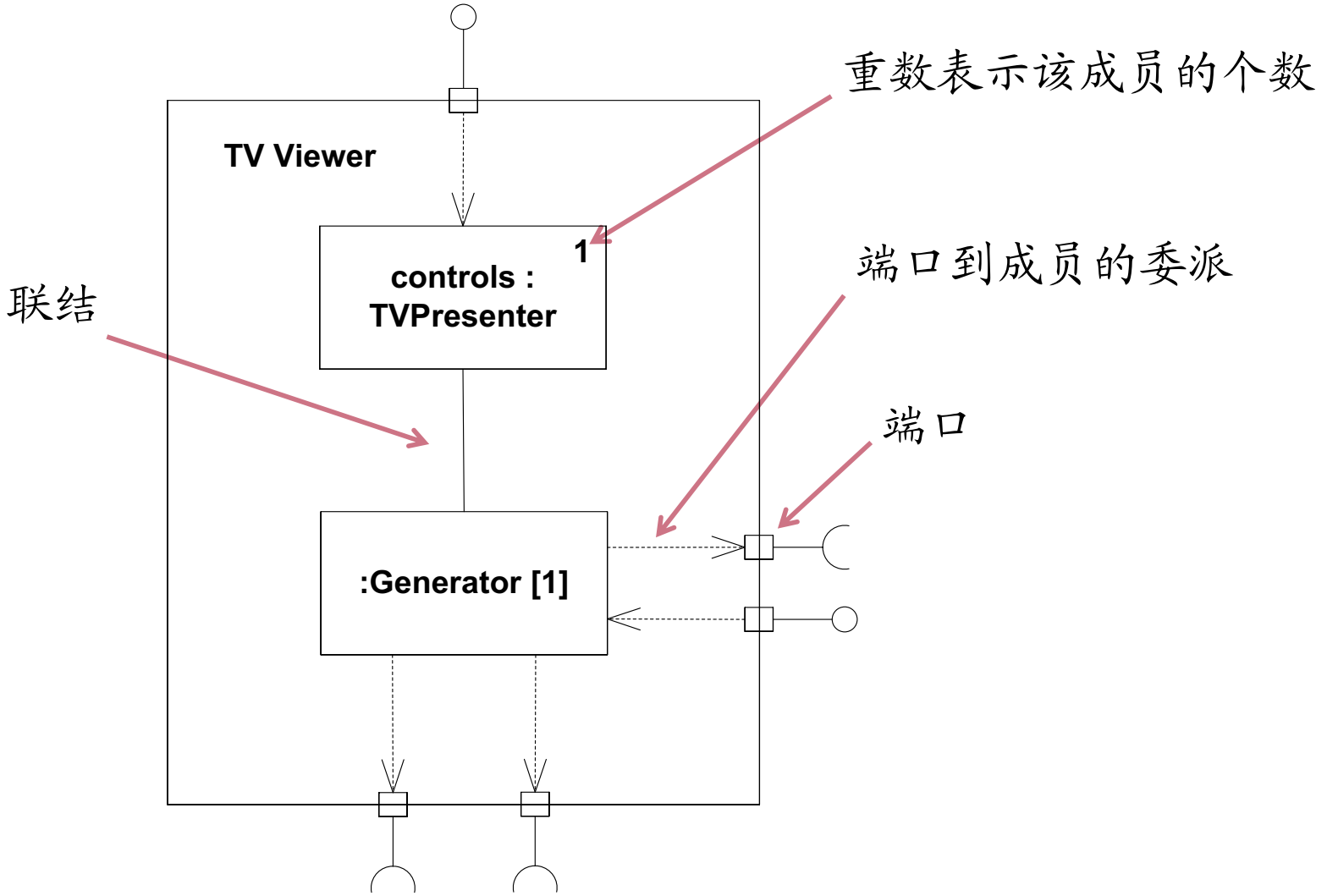




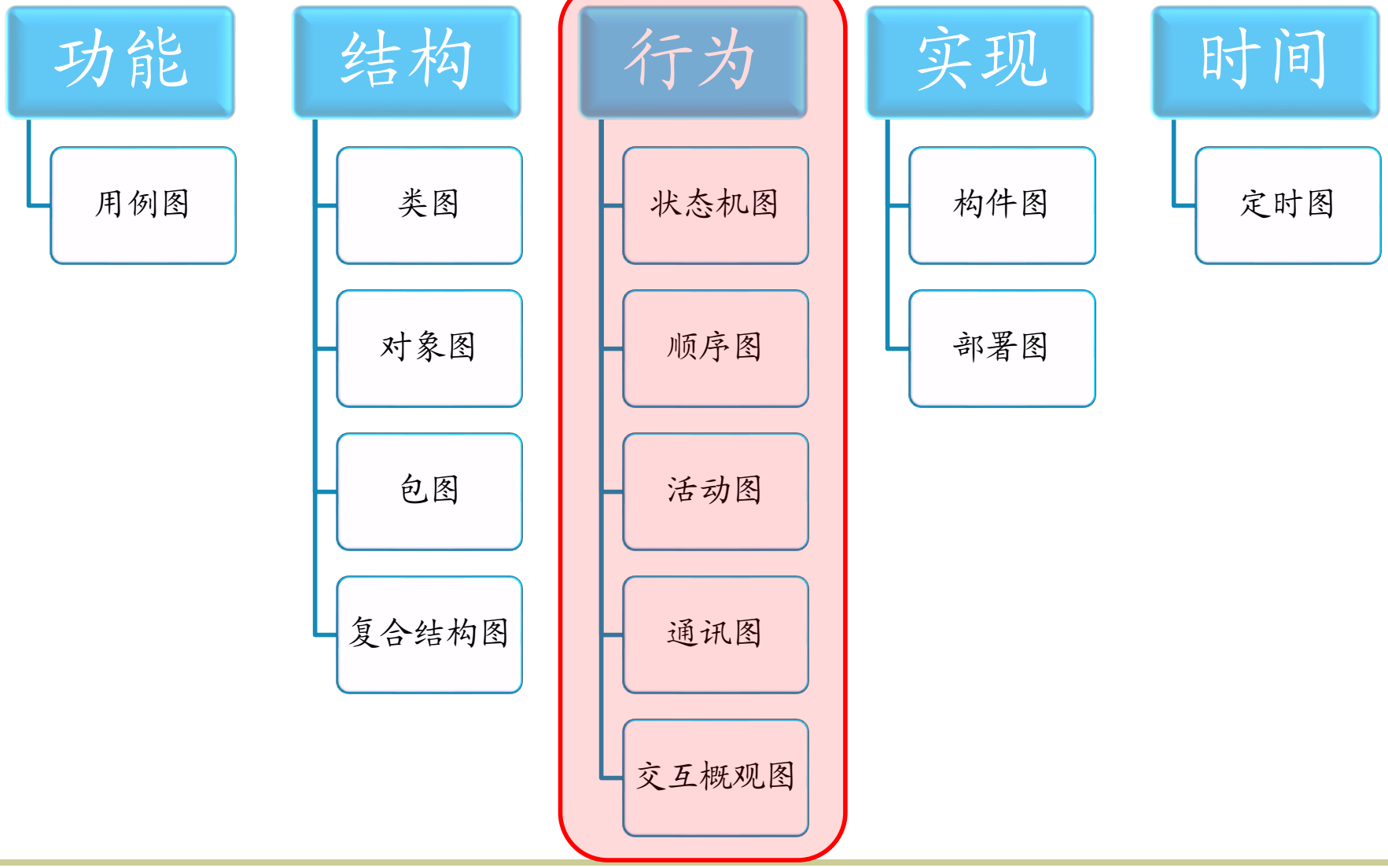
从内部结构的角度描述



TV Viewer的复合结构图



UML2.0的行为图

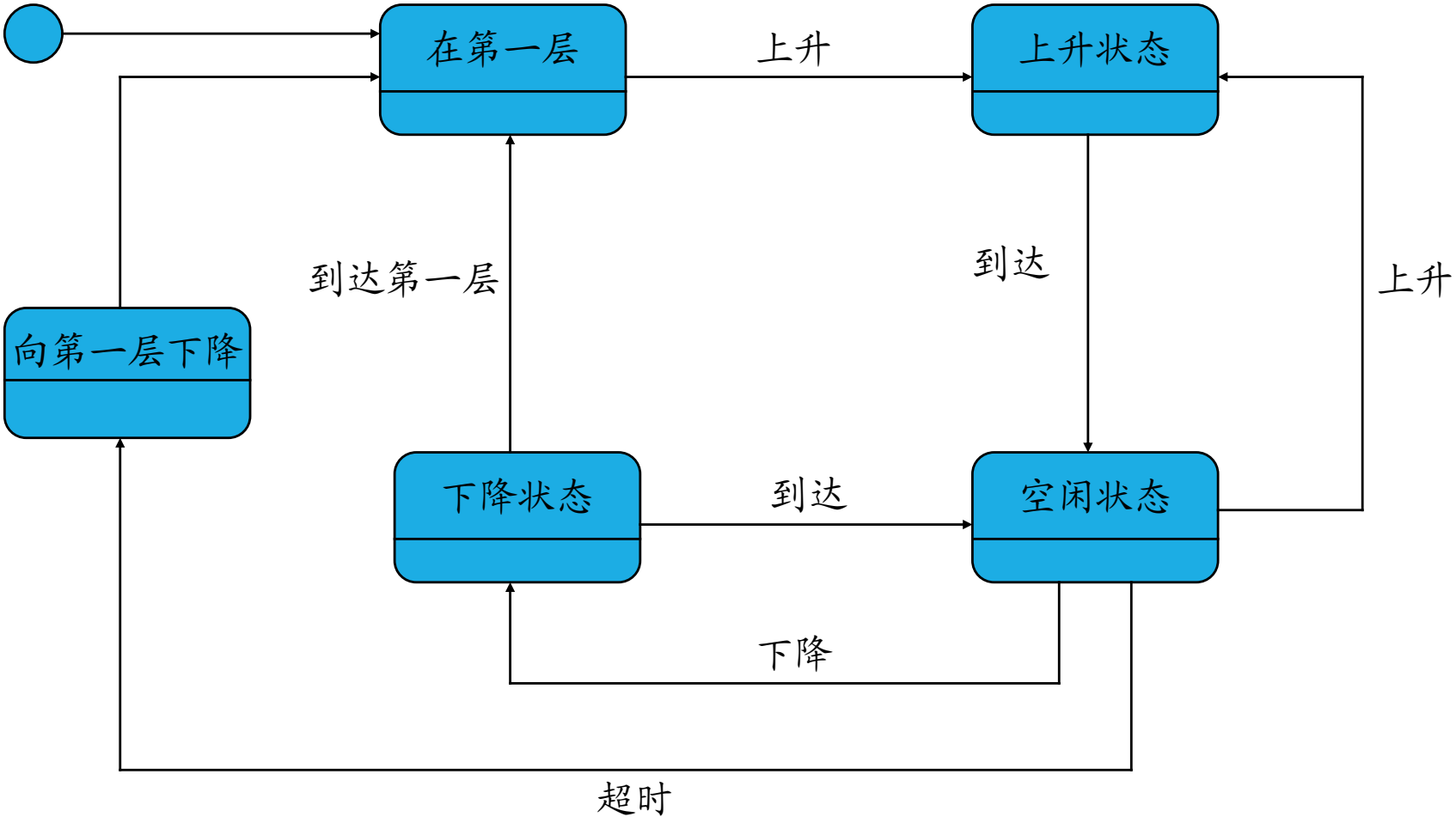

功能
用例图
结构
类图
对象图
包图
复合结构图
行为
状态机图
顺序图
活动图
通讯图
交互概观图
实现
构件图
部署图
时间
定时图

状态机图(State Machine Diagrams)

状态图

- 状态图是对类的一种补充描述，它展示了此类对象所具有的可能的状态以及某些事件发生时其状态的转移情况。
- 在状态图中，状态由圆角矩形表示。状态的改变称作转移，状态转移由箭头表示，箭头旁可以标出转移发生的条件。状态转移可以伴随有某个动作，它表明当转移发生时系统要做什么。

状态机图(State Machine Diagrams)

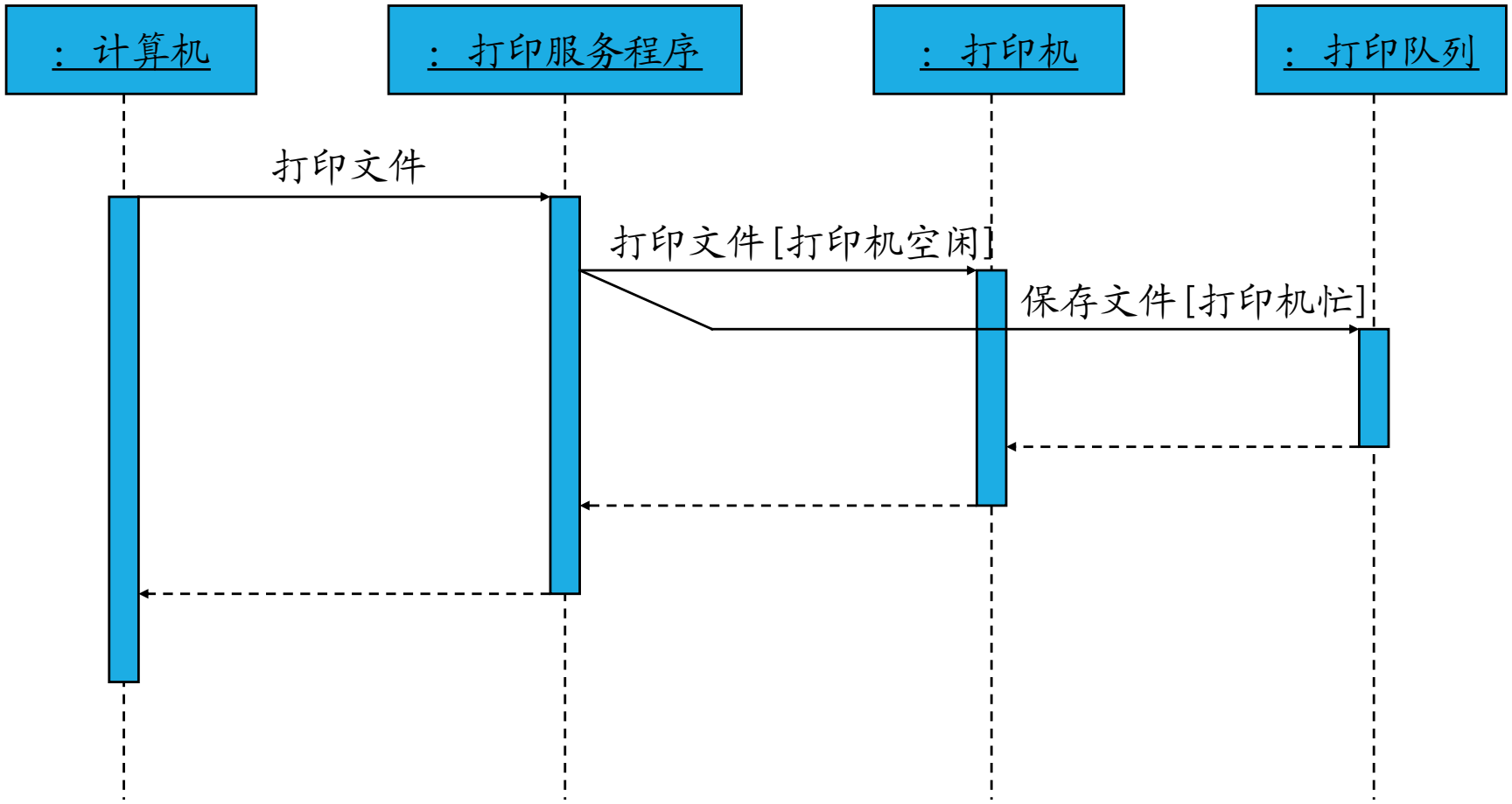


顺序图（Sequence Diagram）

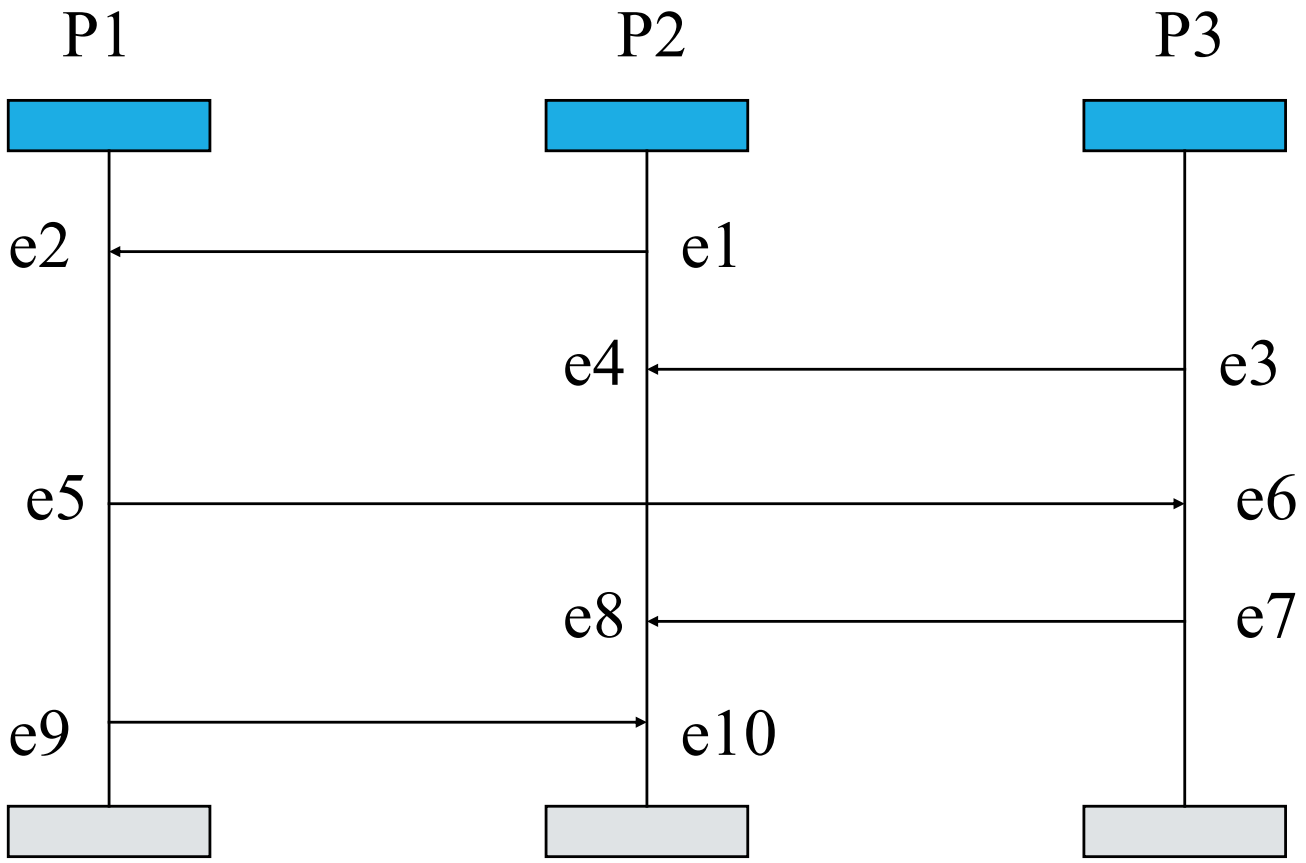
顺序图

- 顺序图描述了对象之间动态的交互关系，着重体现对象间消息传递的时间顺序。
- 顺序图由一组对象构成，每个对象分别带有一条竖线，称作对象的生命线，它代表时间轴，时间沿竖线向下延伸。顺序图描述了这些对象随着时间的推移相互之间交换消息的过程。消息用从一条垂直的对象生命线指向另一个对象的生命线的水平箭头表示。图中还可以根据需要增加有关时间的说明和其他注释。

顺序图（Sequence Diagram）



顺序图（Sequence Diagram）



顺序图（Sequence Diagram）

顺序图中的事件顺序:

- 因果性（Causality）:

  对同一消息而言，发送事件先于接收事件。

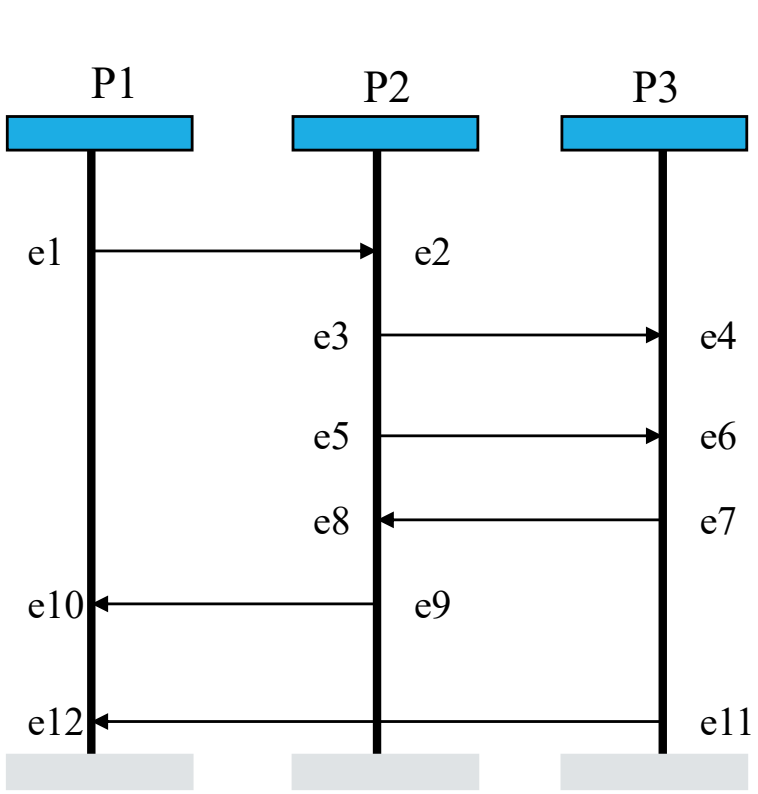
- 可控性（Controlability):

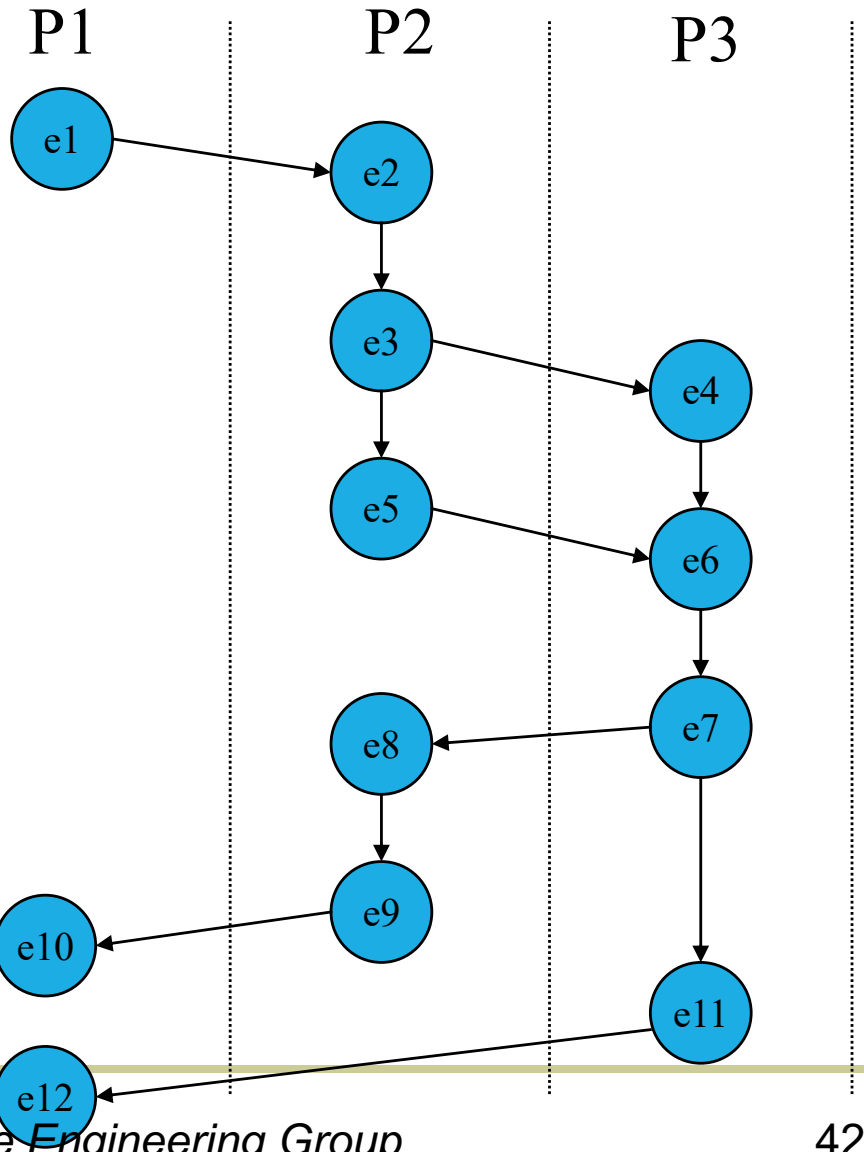
  对同一对象而言，事件p出现在发送事件q的上方，则p先于q。

- 队列性（FIFO):

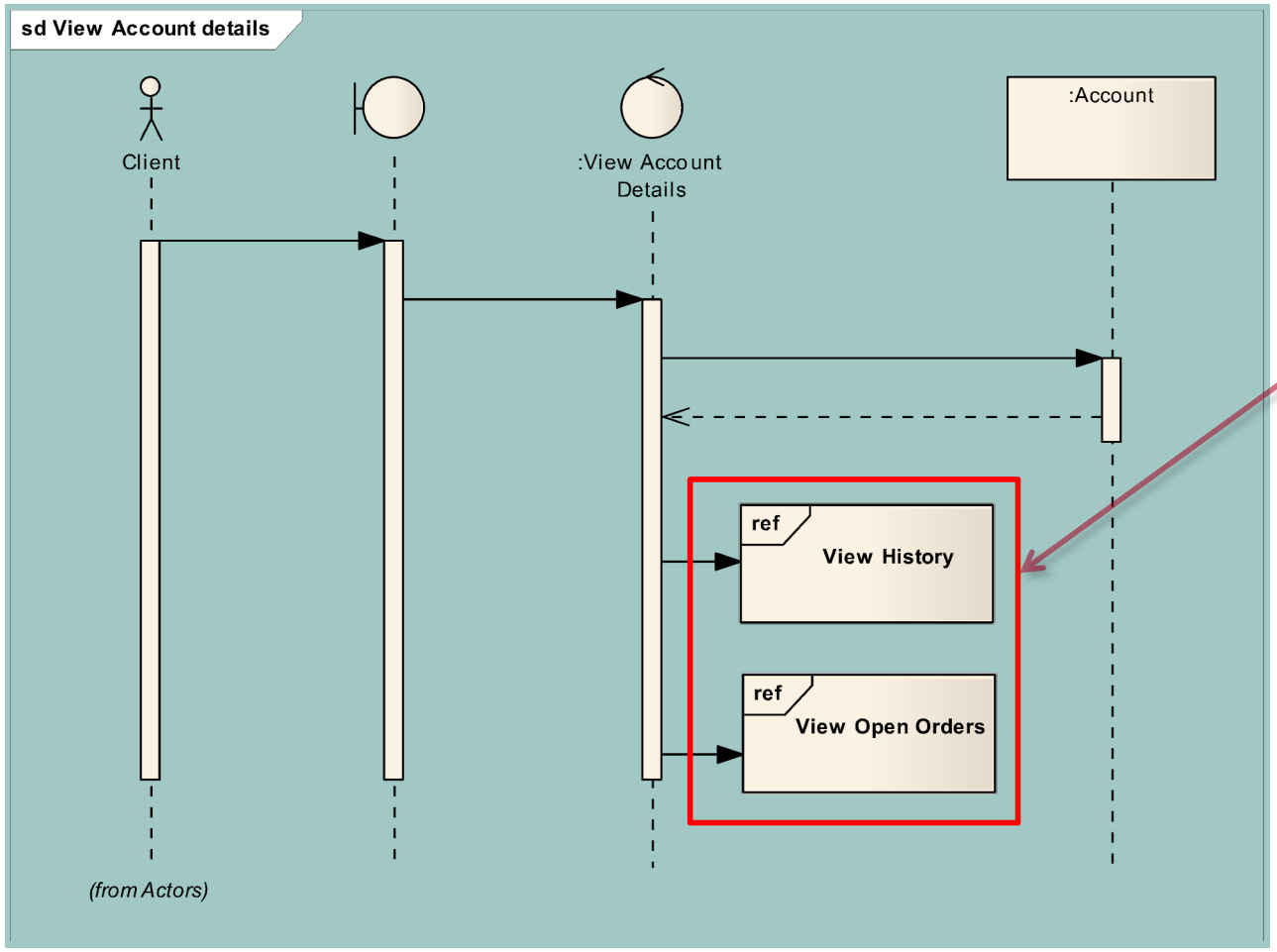
  对同一对象而言，接收事件p出现在接收事件q的上方，并且它们分别对应的发送事件也位于同一个对象，则p先于q。

顺序图（Sequence Diagram）



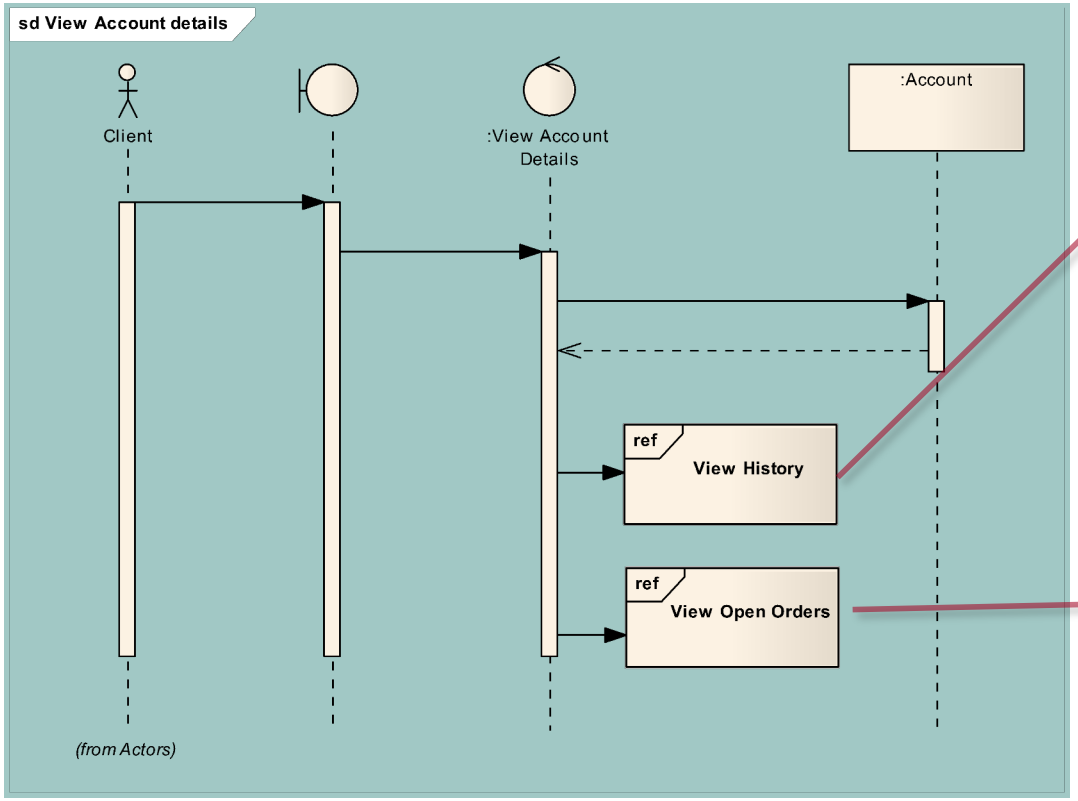


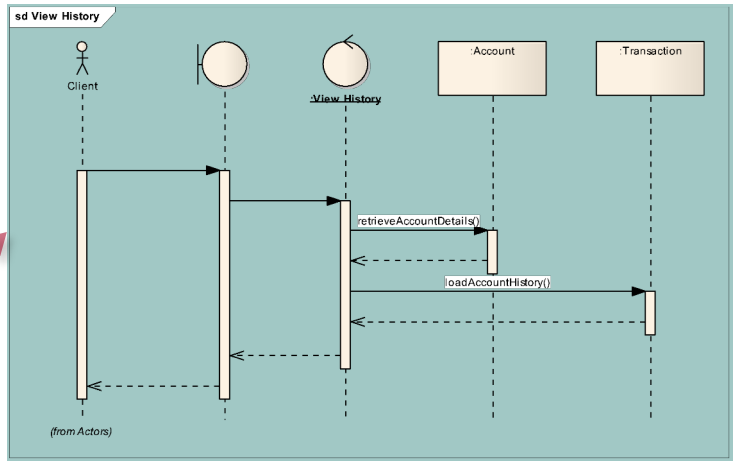
带引用片段的顺序图

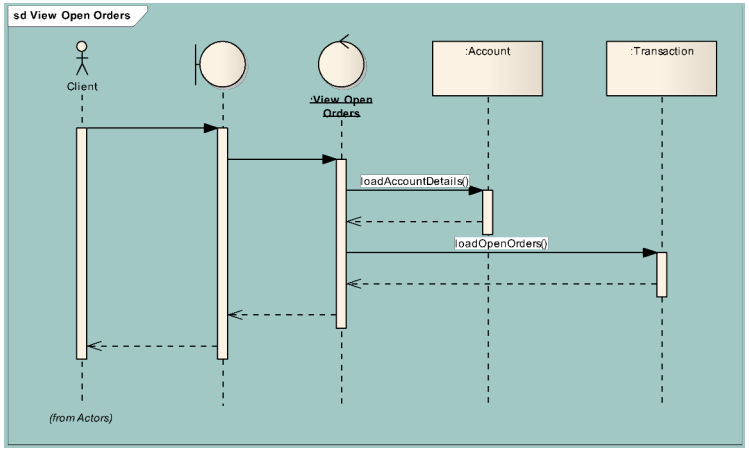


引用片段用ref标示，代表了一个复杂的顺序图

通过ref定义的顺序图片段





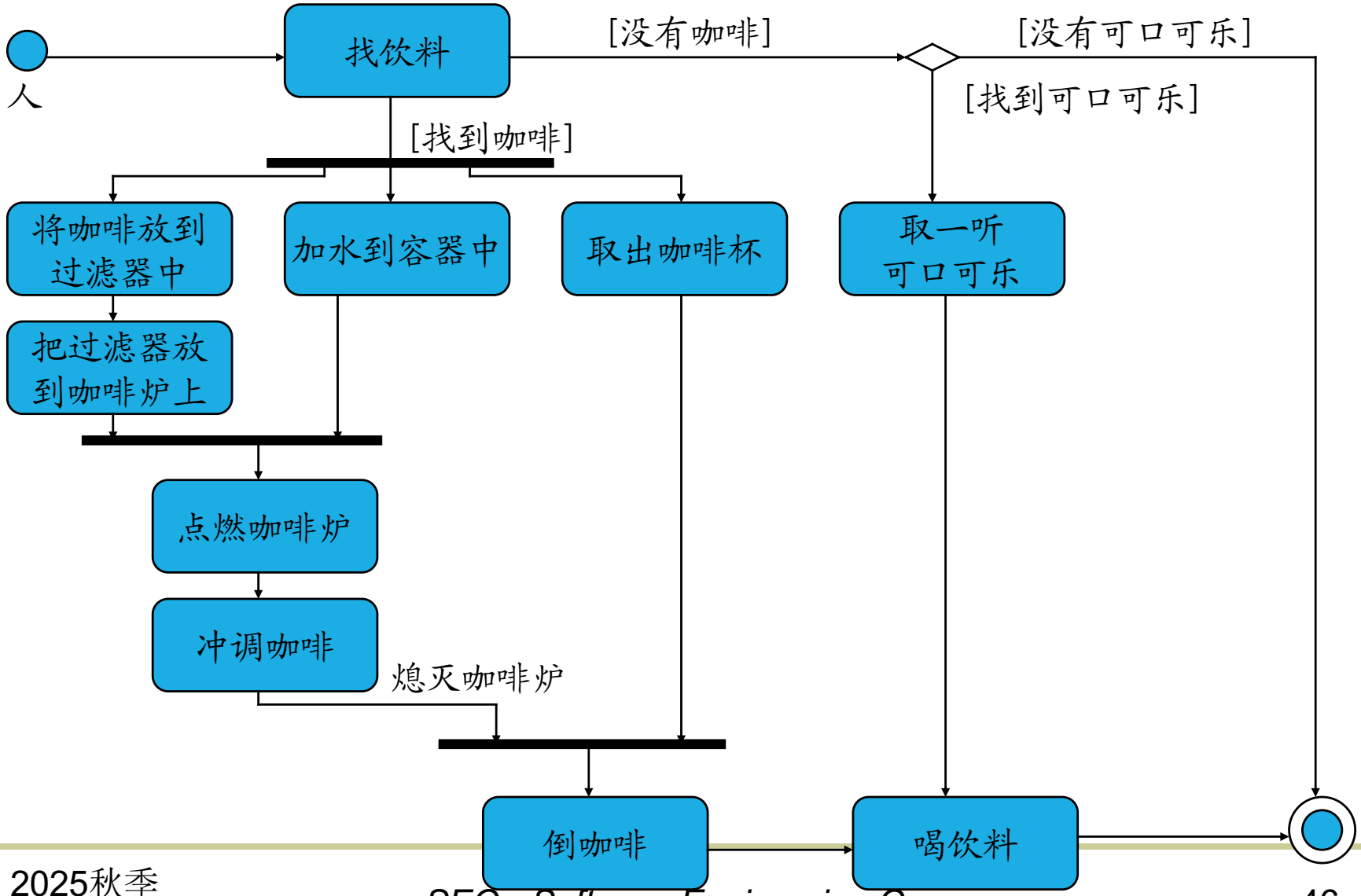


活动图（Activity Diagram）

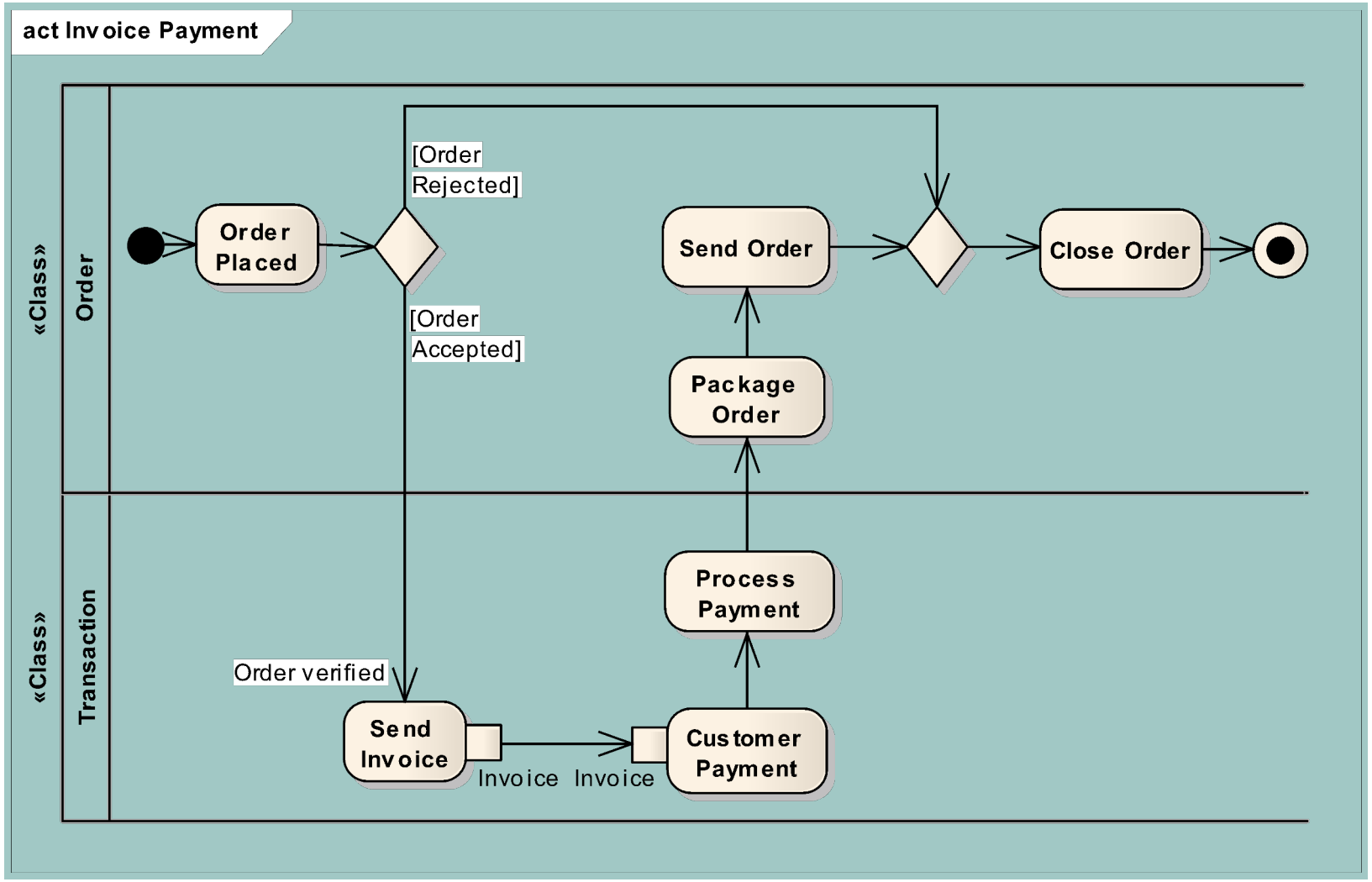
活动图

- 活动图描述系统中各种活动的执行顺序，通常用于描述一个操作中所要进行的各项活动的执行流程。同时，它也常被用来描述一个用例的处理流程，或者某种交互流程。
- 活动图由一些活动组成，图中同时包括了对这些活动的说明。当一个活动执行完毕之后，控制将沿着控制转移箭头转向下一个活动。活动图中还可以方便地描述控制转移的条件以及并行执行等要求。

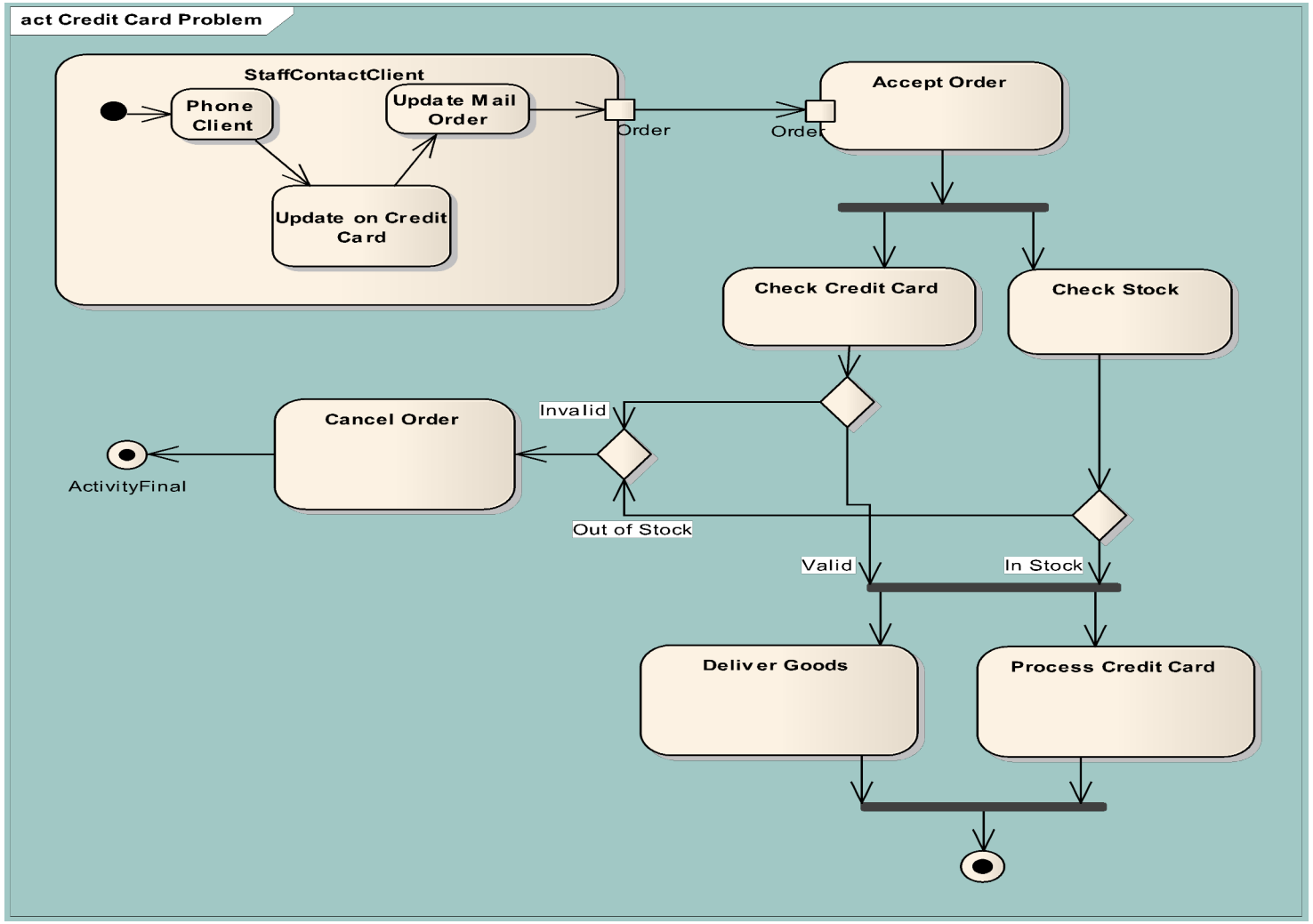
活动图（Activity Diagram）



区分多个对象的活动图


act Invoice Payment
«Class»
Order
«Class»
Transaction
Order Placed
[Order Rejected]
[Order Accepted]
Send Order
Close Order
Package Order
Process Payment
Order verified
Send Invoice
Invoice
Invoice
Customer Payment

包含子活动的活动图


act Credit Card Problem
StaffContactClient
Phone Client
Update Mail Order
Update on Credit Card
Order
Order
Accept Order
Check Credit Card
Check Stock
Invalid
Cancel Order
ActivityFinal
Out of Stock
Valid
In Stock
Deliver Goods
Process Credit Card

活动图（Activity Diagram）

- 活动图最适合支持对控制过程进行描述，这使之成为支持工作流建模的最好工具。
- 同时，在并发行为方面，活动图也具有非常清晰直观的特点。
- 活动图最大的缺点是很难清楚地描述动作与对象之间的关系。

活动图（Activity Diagram）

- 对于以下情况可以使用活动图:

（1）分析用例;

（2）理解牵涉多个用例的工作流;

（3）处理多线程应用。

- 在下列情况下，一般不要使用活动图:

（1）显示对象间合作;

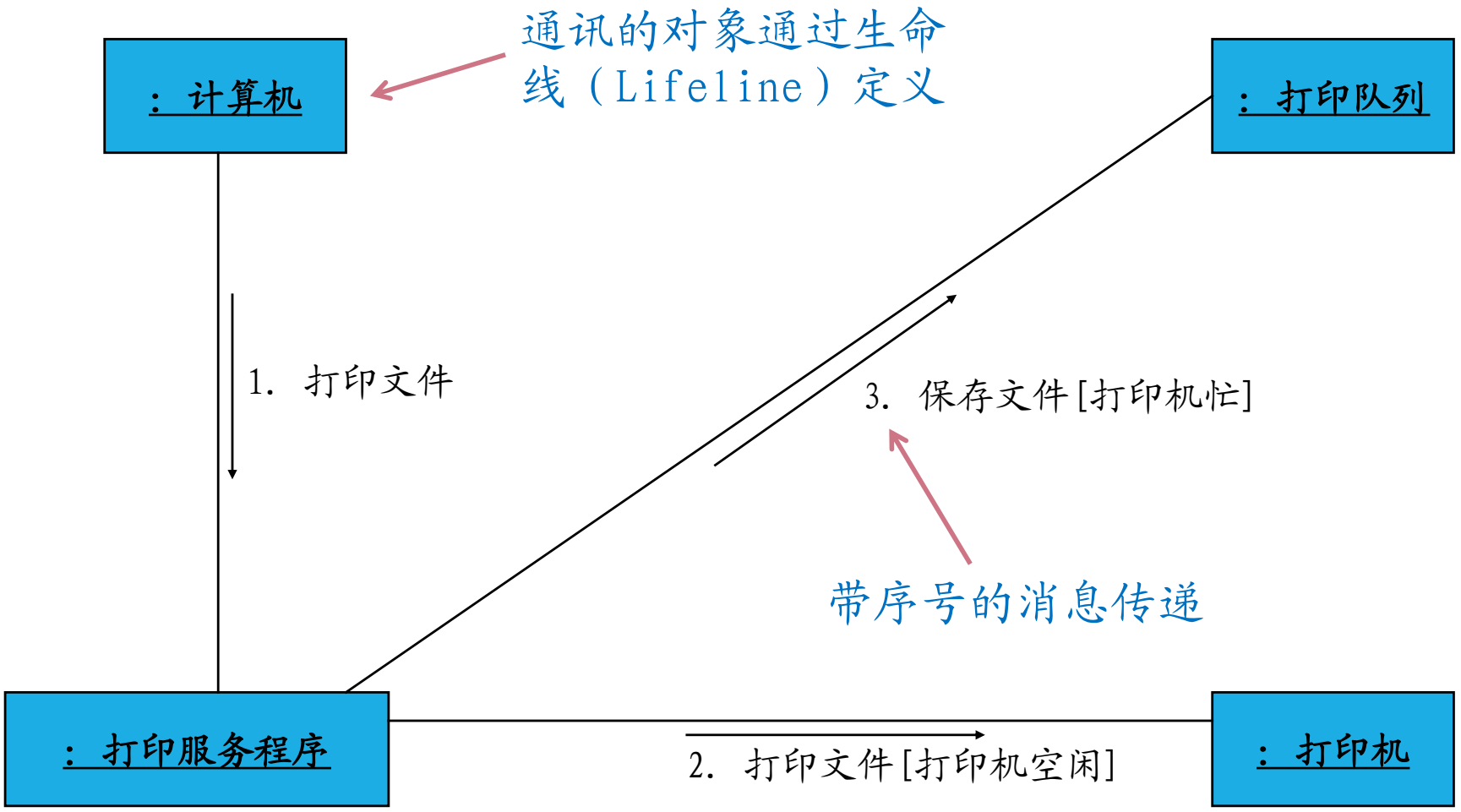
（2）显示对象在其生命周期内的运转情况。

通讯图（ Communication Diagrams ）

通讯图

- 通讯图与简单顺序图（既不包含结构片段的顺序图）具有对应关系
- 通讯图侧重于描述各个对象之间存在的消息收发关系（交互关系），而不专门突出这些消息发送的时间顺序。
- 在通讯图中，对象同样是用一个对象图符来表示，箭头表示消息发送的方向，而消息执行的顺序则由消息的编号来表明。

通讯图（ Communication Diagrams ）



通讯图（ Communication Diagrams ）

- 通讯图的布局方法能更清楚地表示出对象之间静态的连接关系。
- 顺序图突出执行的时序，能更方便地看出事情发生的次序。
- 如果要描述在一个用例中的几个对象协同工作的行为，通讯图是一种有力的工具。通讯图擅长显示对象之间的合作关系，尽管它并不对这些对象的行为进行精确的定义。
- 如果想要描述跨越多个用例的单个对象的行为，应当使用状态图；如果想要描述跨越多个用例或多个线程的多个对象的复杂行为，则需考虑使用活动图。

交互概观图(Interaction Overview Diagrams)

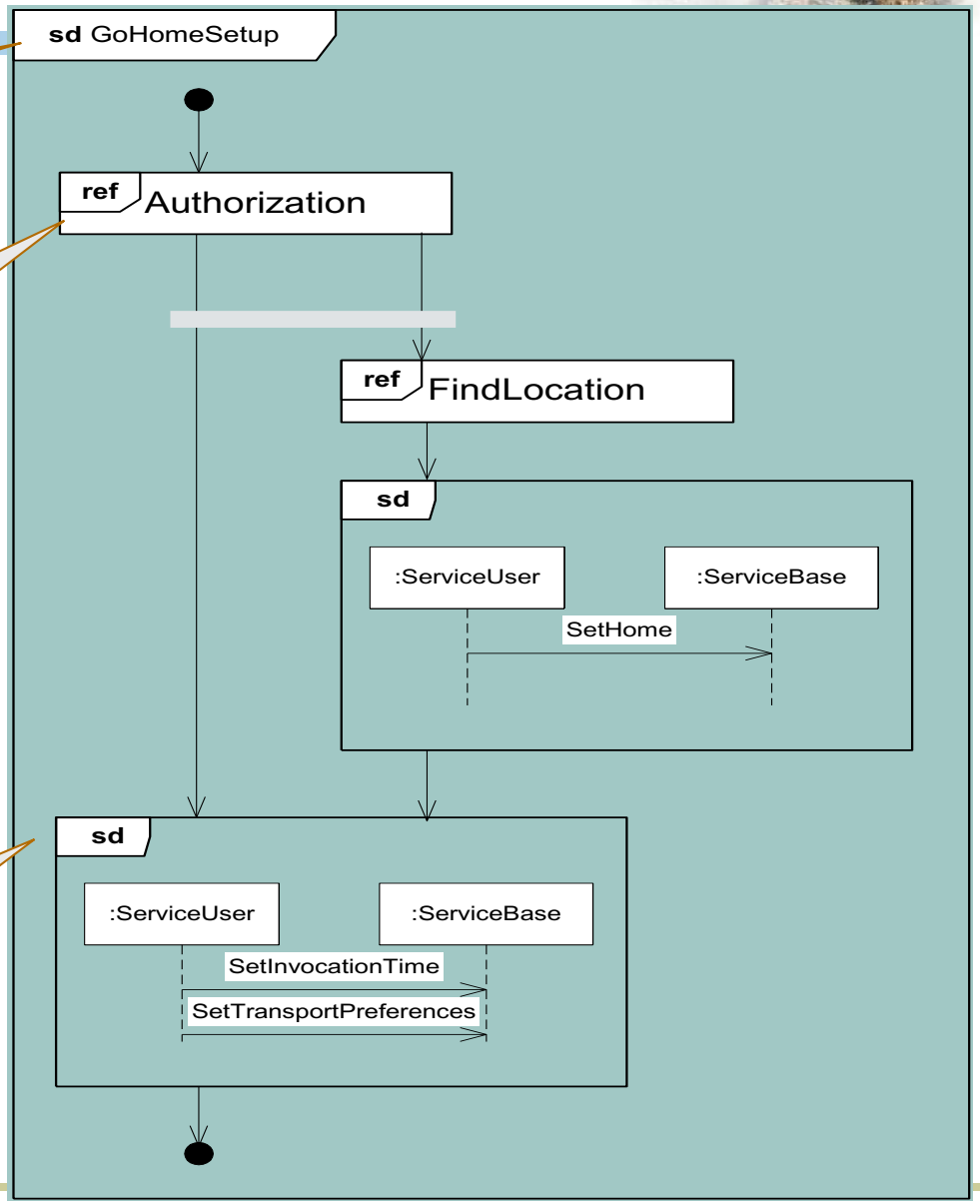
- 交互概观图可以看做是活动图和顺序图的结合体
- 交互概观图在整体上强调控制流的描述特点
- 生命线和消息只能在交互(Interaction)片断内部，而不能出现在概观的层次上

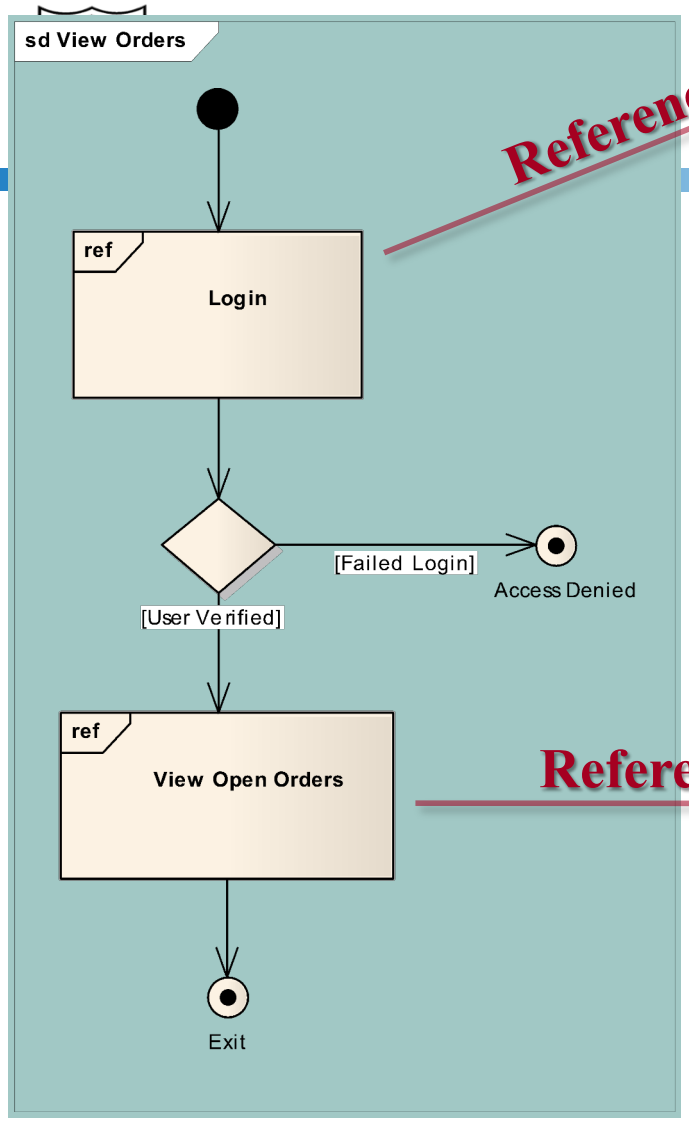
交互概观图示例

通过活动图语法描述的交互概观

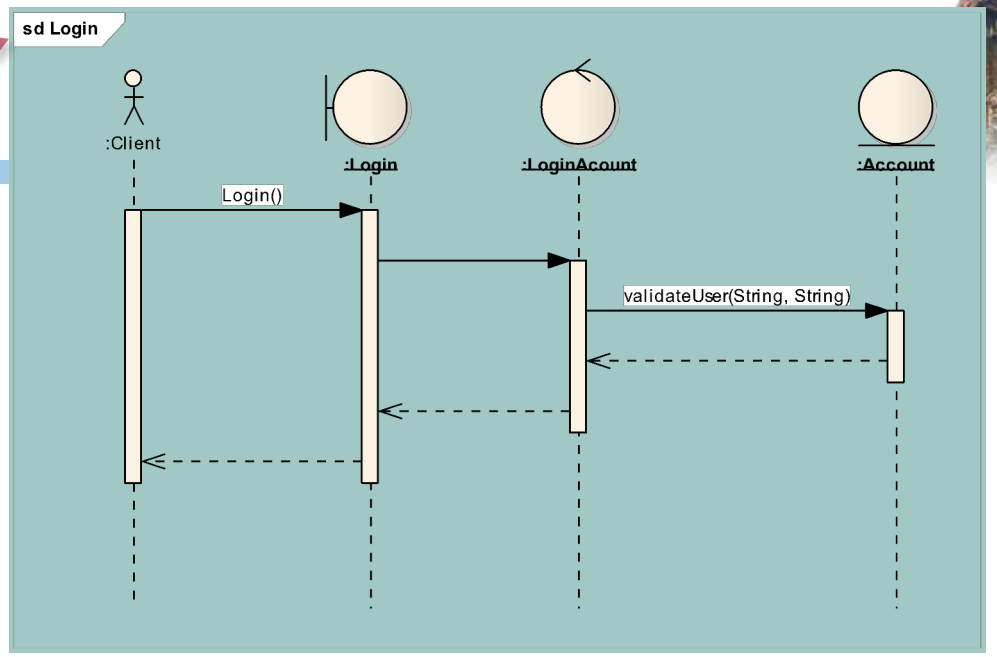
交互出现
(Interaction Occurrence)

展开的交互片段

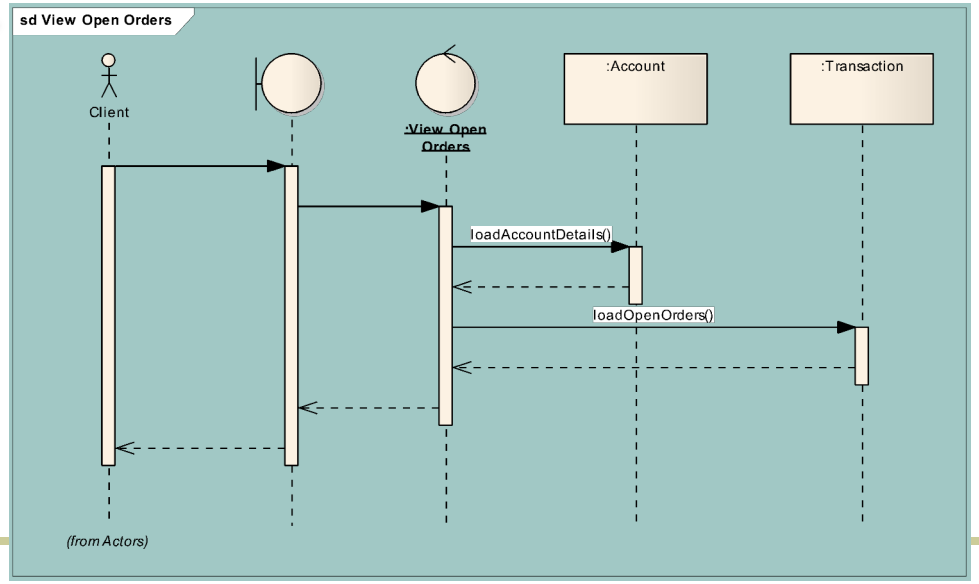




Reference



Reference



通过 Ref 定义的交互出现可以进一步展开

UML2.0的实现图

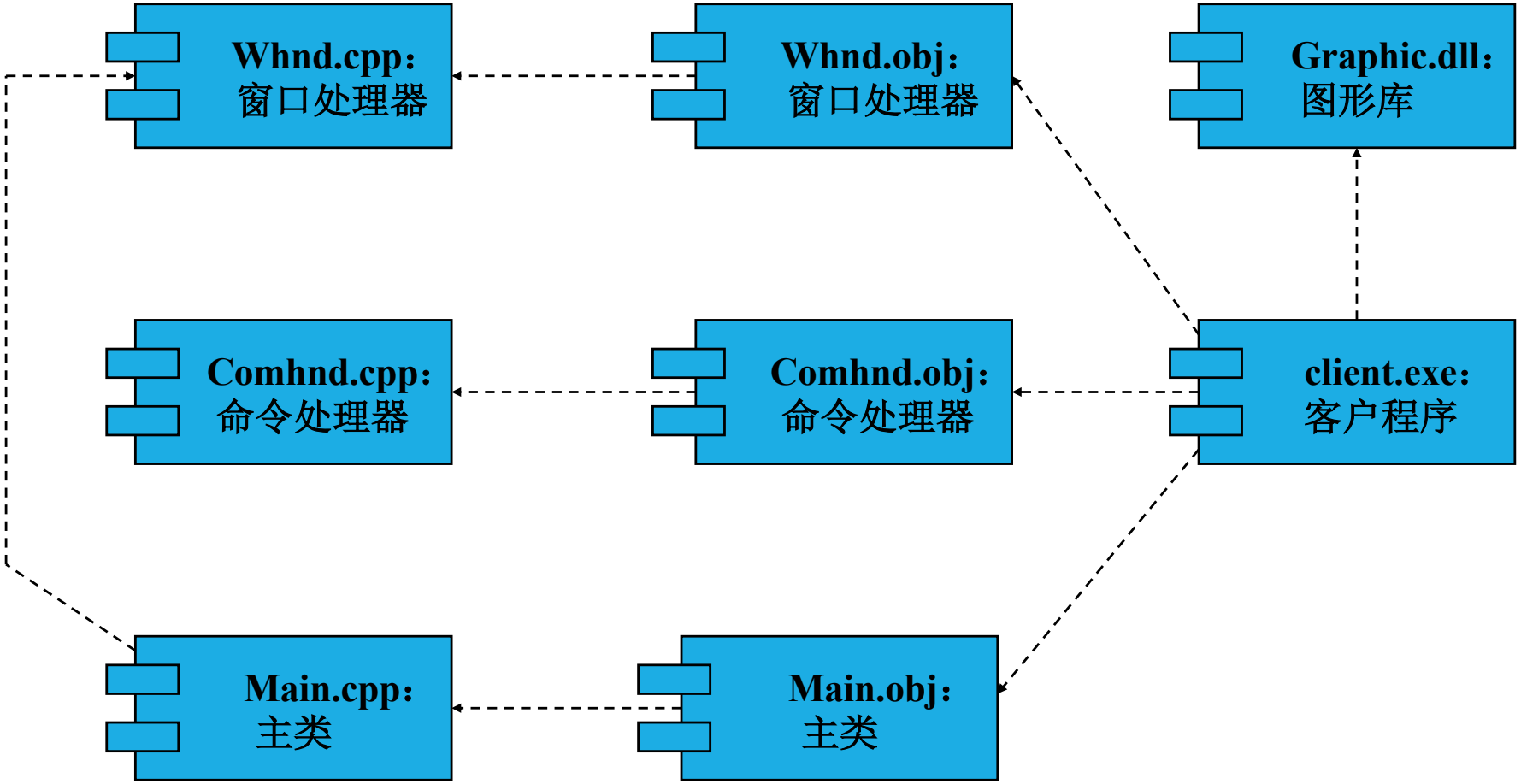
- 功能
  - 用例图
- 结构
  - 类图
  - 对象图
  - 包图
  - 复合结构图
- 行为
  - 状态机图
  - 顺序图
  - 活动图
  - 通讯图
  - 交互概观图
- 实现
  - 构件图
  - 部署图
- 时间
  - 定时图

构件图（Component Diagram）

构件图

- 构件图描述软件构件以及它们之间的依赖关系，从而便于人们分析和发现当修改某个构件时可能对那些构件产生影响，以便对它们做相应的修改或更新。构件可以是源代码构件、二进制目标码构件、可执行构件或文档构件。

构件图（Component Diagram）

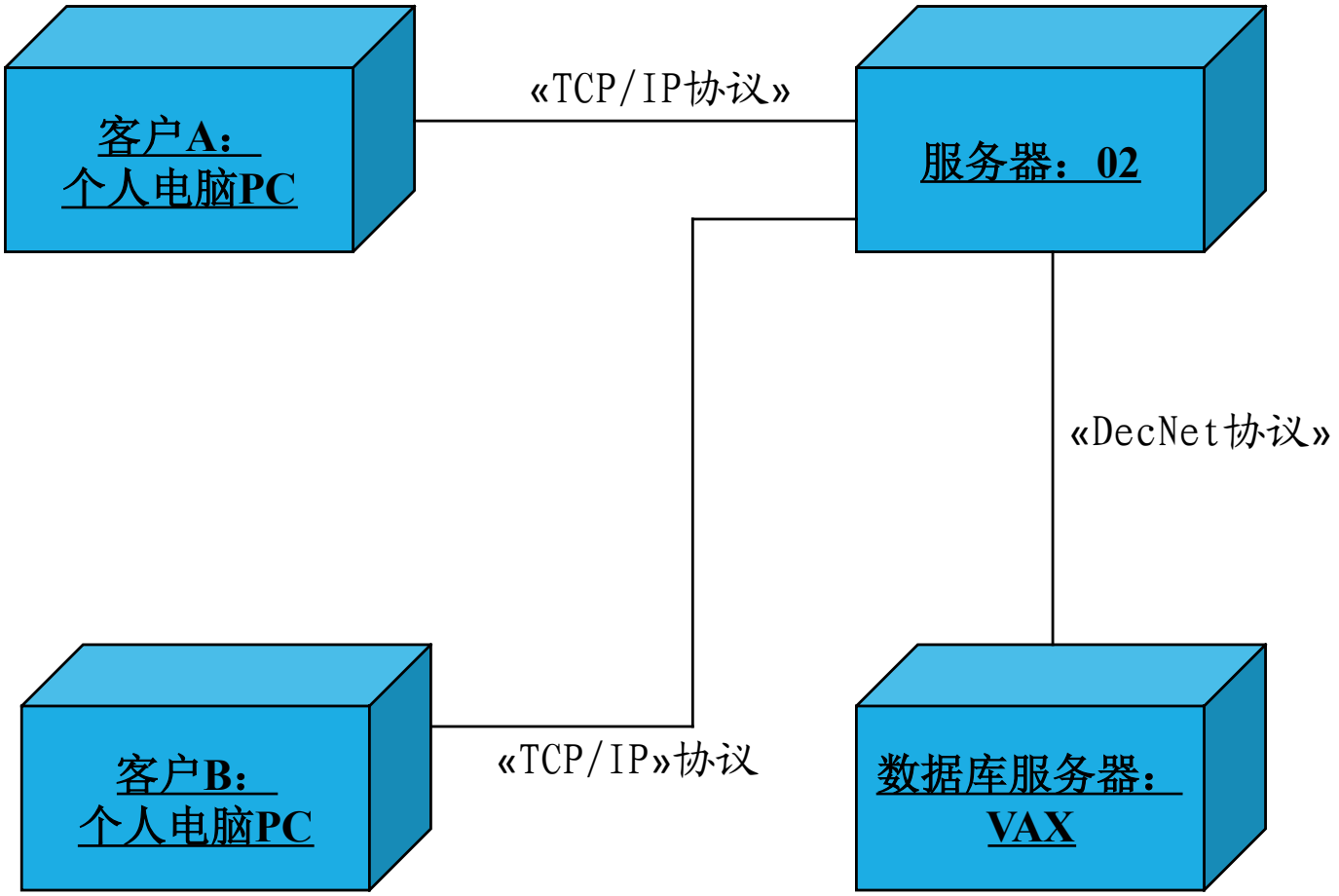


部署图（Development Diagram）

配置图

- 配置图描述系统中硬件和软件的物理配置情况和系统体系结构。
- 在配置图中，用结点表示实际的物理设备，如计算机和各种外部设备等，并根据它们之间的连接关系，将相应的结点连接起来，并说明其连接方式。在结点里面，说明分配给该结点上运行的可执行构件或对象，从而说明哪些软件单元被分配在哪些结点上运行。

部署图（Development Diagram）



UML2.0的实现图

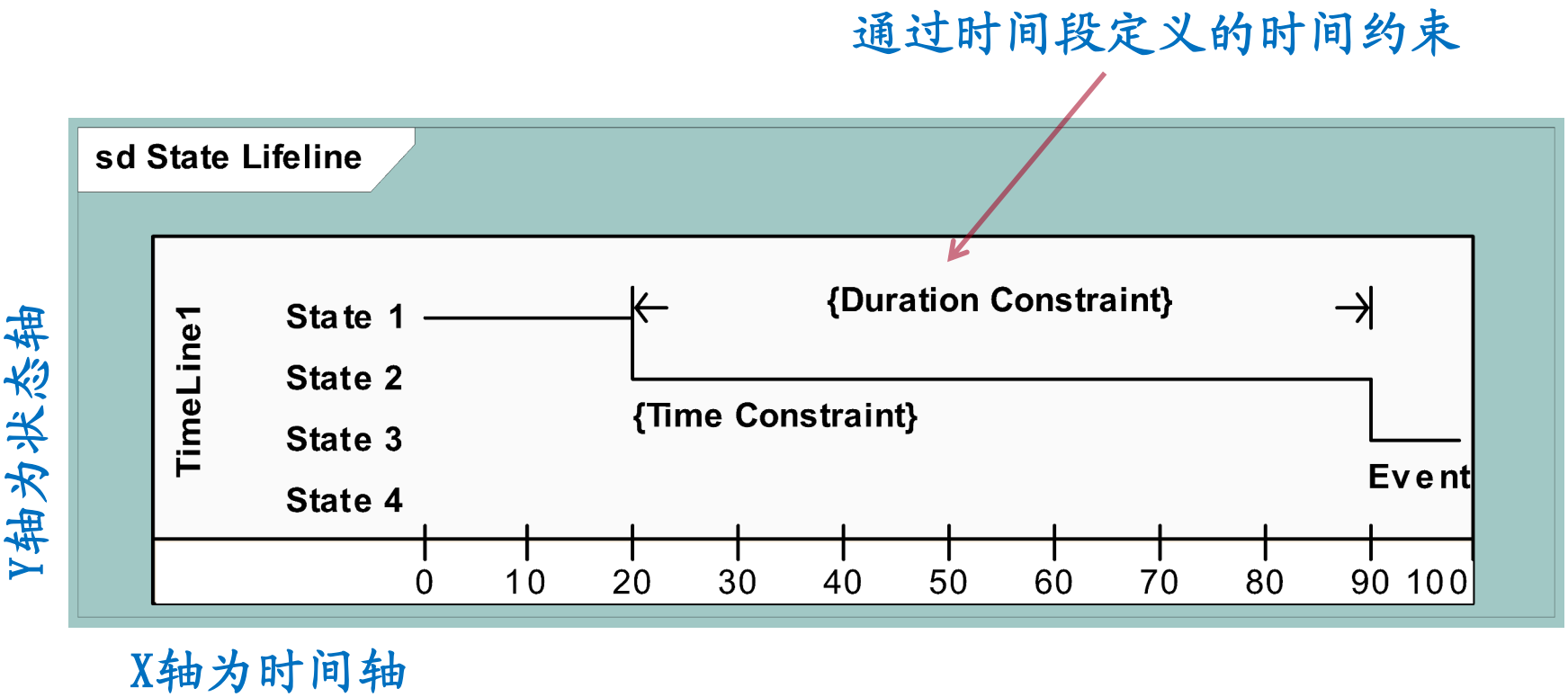
- 功能
  - 用例图
- 结构
  - 类图
  - 对象图
  - 包图
  - 复合结构图
- 行为
  - 状态机图
  - 顺序图
  - 活动图
  - 通讯图
  - 交互概观图
- 实现
  - 构件图
  - 部署图
- 时间
  - 定时图

定时图（Timing Diagram）

- 定时图主要用来描述状态（state）或值（value）随时间的变化情况
- 定时图也可以描述带时间约束的交互行为

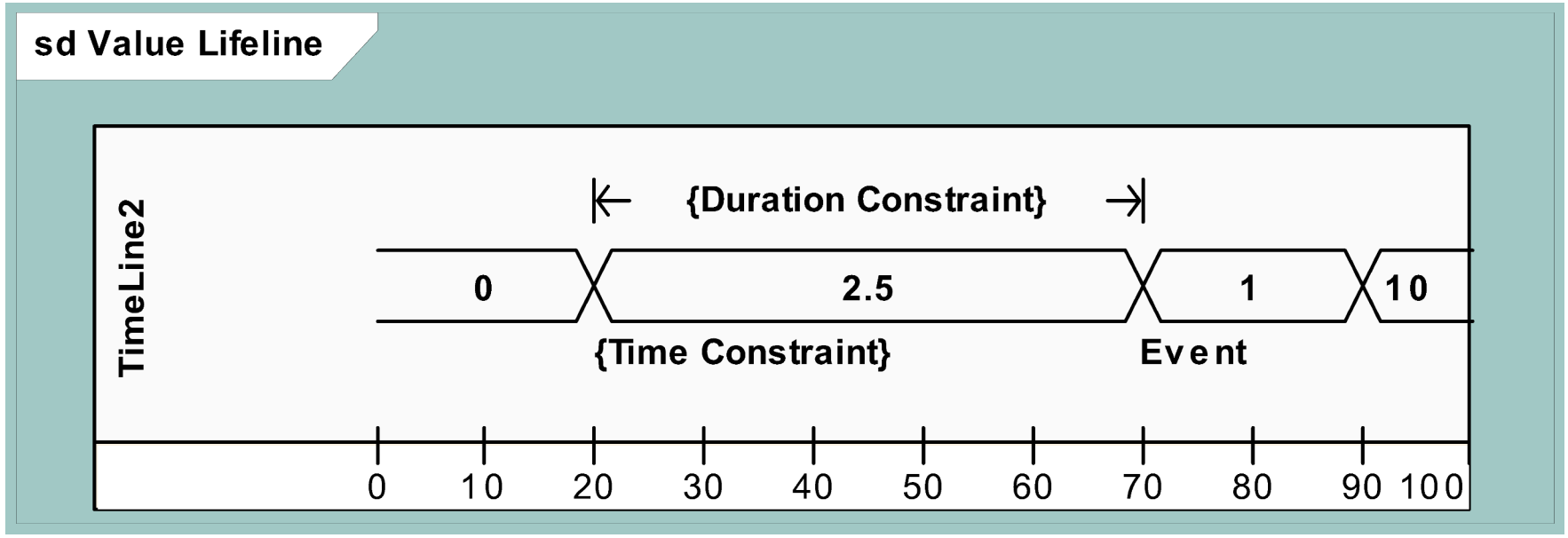
State Lifeline的定时图

State Lifeline描述了状态随时间的变化情况



Value Lifeline的定时图

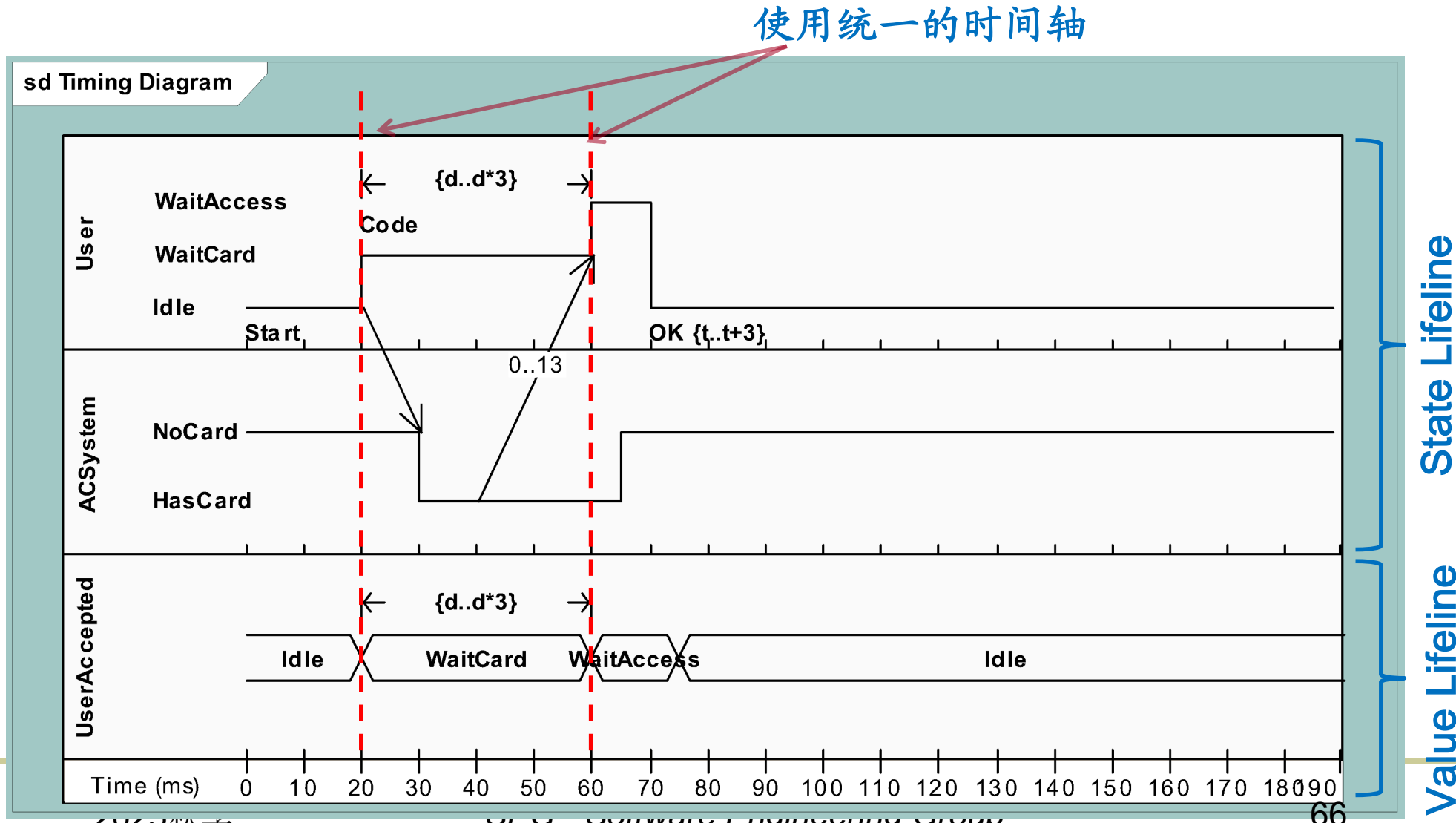
- Value Lifeline描述了值随时间的变化情况



Y轴为状态轴

X轴为时间轴

状态和值合并的定时图


使用统一的时间轴
sd Timing Diagram
User
WaitAccess
Code
WaitCard
Idle
Start
{d..d*3}
OK {t..t+3}
0..13
ACSystem
NoCard
HasCard
UserAccepted
{d..d*3}
Idle
WaitCard
WaitAccess
Idle
Time (ms)
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190
State Lifeline
Value Lifeline

UML支撑环境

Software Architecture

- 基于UML的模型驱动的软件开发环境
- 全面支持团队整体合作的开发形式
- 集成了最新软件开发技术和思想

UML的扩展

- 实时模型　UML-RT
- 可执行模型
- 企业计算

  Enterprise Distributed Object Computing (EDOC)

  Enterprise Application Integration (EAI)
- 软件过程　Rational Unified Process (RUP)
- 其他　　Standard for Data Warehousing

  CORBA maps to UML

  XMI format for the exchange of UML models in text format

UML框架下的软件工程

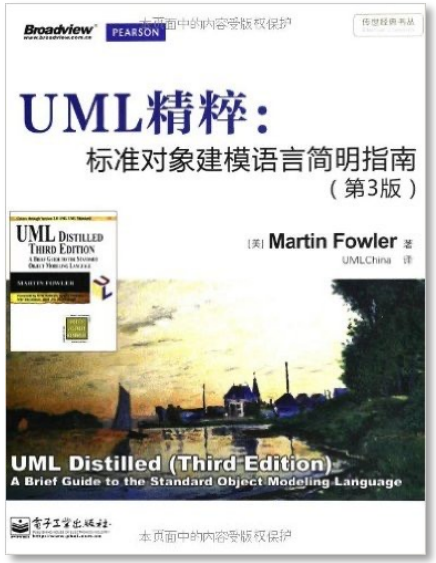
- 我们已经有了统一的建模语言UML
- 我们正在拥有统一软件过程（RUP?）
- 下一步是什么？
  - A Software Component Marketplace
  - Quality from the Beginning
  - Give Soul to Software Process
  - A Complete UML Based Software Platform

Ivar Jacobson

推荐书籍

软件建模与设计:UML、用例、模式和软件体系结构

UML用户指南(第2版)(修订版)



UML精粹:标准对象建模语言简明指南(第3版)