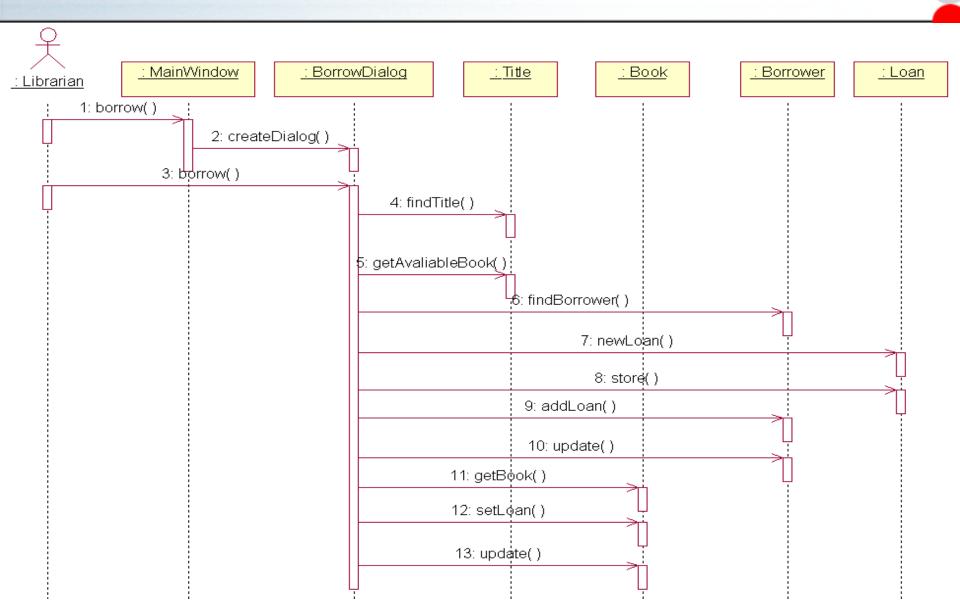


交互图

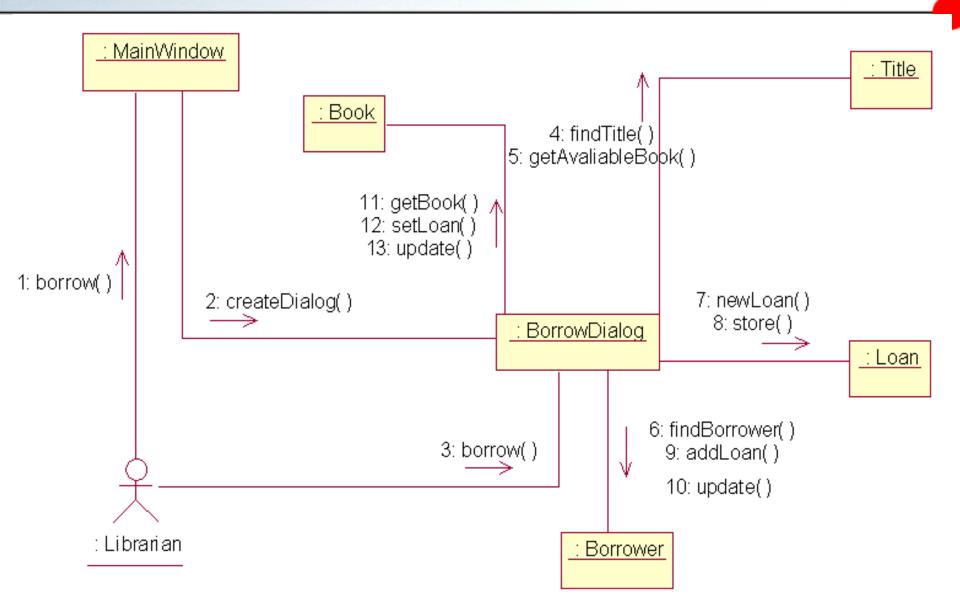


- 描述系统的动态建模以及如何在一个模型中捕获它。
- 介绍交互图的两种形式
 - 顺序图 (sequence diagram)
 - 协作图(collaboration diagram)

顺序图示例



协作图示例



对象需要交互

- 对象相互链接的地方就有交互
- 具有对象协作的系统、子系统语境中
 - 如Web商务系统,客户对象、服务器对象间交互
- 操作实现语境中
 - 操作的参数、局部变量、全局对象相互交互完成操作的实现 算法
- 类语境中
 - 通过交互显示类的属性是如何相互协作的

对象通过消息交互

- Messages,传递信息的对象之间所进行的通信,消息 带有对将要发生的活动的期望
- 一个消息实例的接收可看作一个事件的实例
- · 发送消息引发的动作: 执行可执行语句——导致状态改变

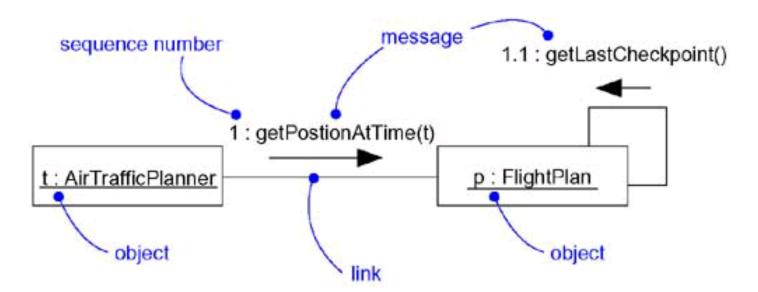
RegisterControl

getCourseOfferings(forSemes...

CourseCatalogSystem

交互图(interaction diagram)

- · 交互图是一种详细表示对象之间以及对象与系统外部的参与者 (actor)之间动态联系的图形文档。
- 交互图是用来描述对象之间的动态协作关系以及协作过程中的行为次序,它常常用来描述一个用例的行为,显示该用例中所涉及的对象和这些对象之间的消息传递情况。

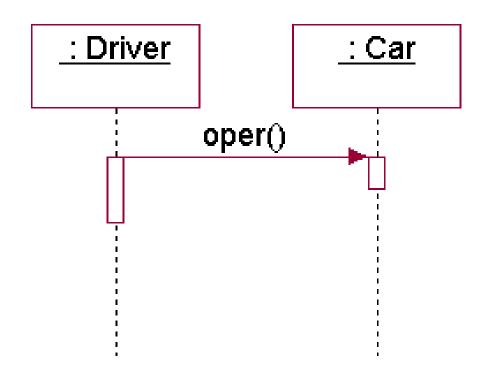


交互图的两种形式

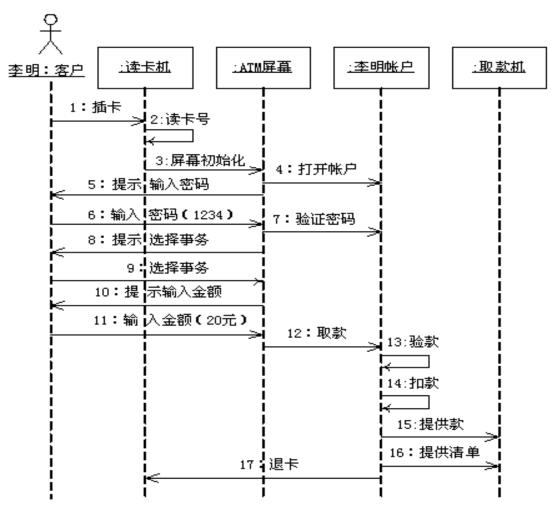
- 交互图有两种形式:
 - 顺序图(sequence diagram)
 - 协作图(collaboration diagram)
- 顺序图:强调消息的时间顺序的交互图。图形上是一张表,对象沿X轴排列,消息沿Y轴按时间顺序排列。
- 协作图:强调发送和接收消息的对象之间的组织结构 的交互图。图形上是顶点和弧的结合

顺序图(Sequence diagram)

• 定义: 顺序图描述了对象之间传送消息的时间顺序,它用来表示用例中的行为顺序。



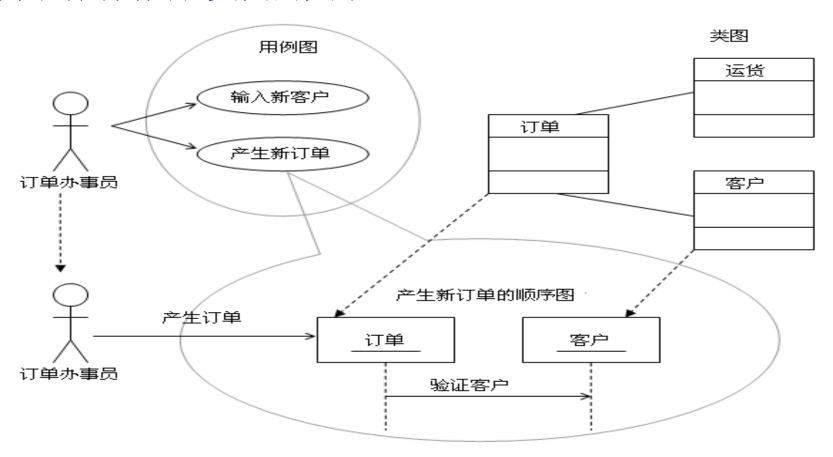
◇ 顺序图



- □ 顺序图显示对象之间的 动态合作关系,它强调 对象之间消息发送的顺 序,同时显示对象之间 的交互。
- □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引起状态转换的触发事件。

- 顺序图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事件。
- 顺序图展示对象之间的交互,这些交互是指在场景或用例的事件流中发生的。顺序图属于动态建模。
- 顺序图的重点在消息序列上,也就是说,描述消息是如何在对象间 发送和接收的。表示了对象之间传送消息的时间顺序。
- 浏览顺序图的方法是: 从上到下查看对象间交换的消息。

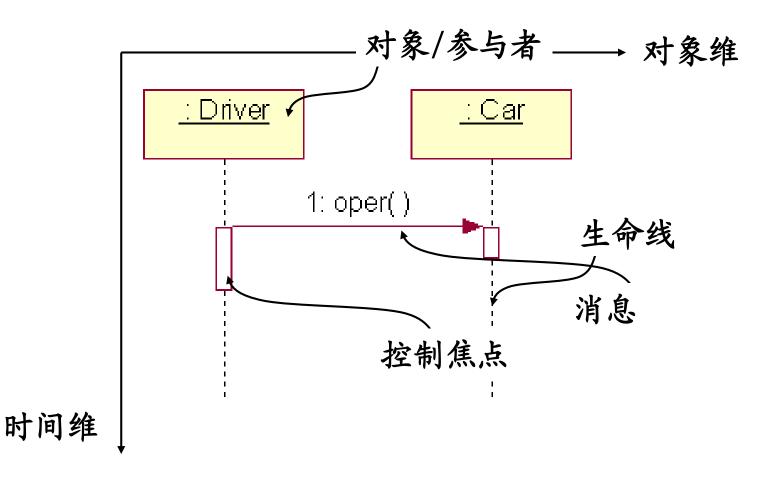
顺序图与用例图和类图的关系



Sequence图中的组成元素

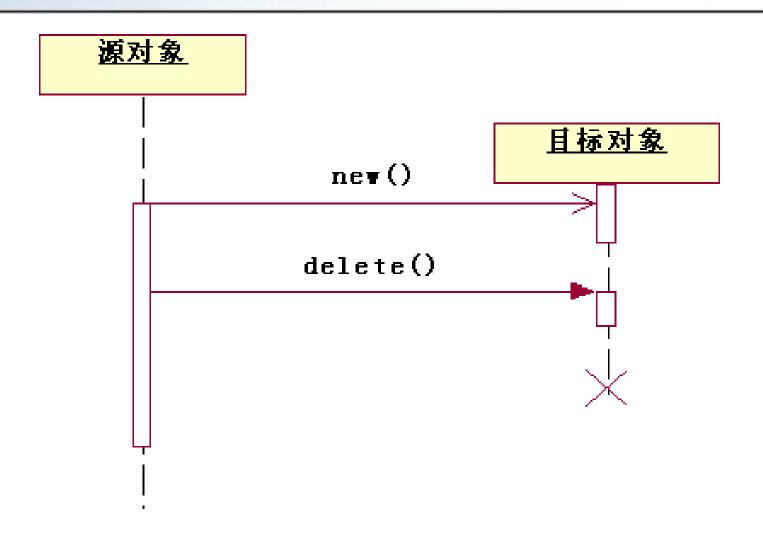
顺序图的组成元素

- · 参与者(actor)或者 对象(object)
- 生命线(lifeline)
- 激活期(activation)/ 控制焦点(focus of control)
- · 消息 (message)



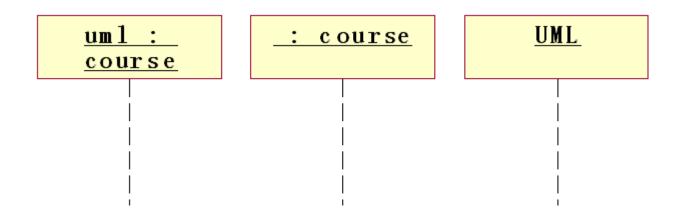
参与者或对象

- 参与者和对象按照从左到右的顺序排列;
- 一般最多两个参与者,他们分列两端。启动这个用例的参与者往往排在最左边,接收消息的参与者则排在最右端;
- 对象从左到右按照重要性排列或按照消息先后顺序排列;
- 将对象置于顺序图的顶部意味着在交互开始的时候对象 就已经存在了,如果对象的位置不在顶部,那么表示对 象是在交互的过程中被创建的。



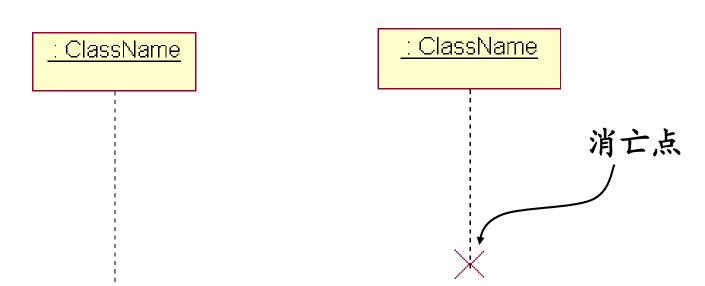
对象的命名方式有三种:

- 包括对象名和类名
- 类名(匿名对象)
- 对象名(不关心类)



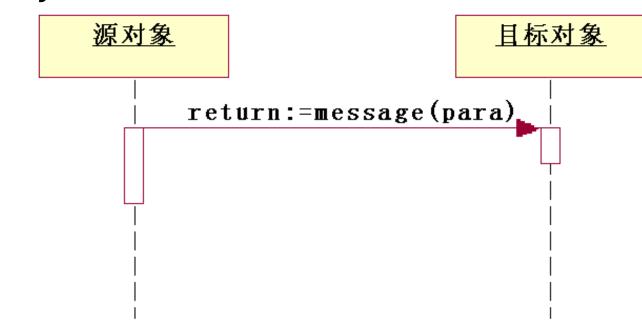


- ·每个对象都有自己的生命线,用来表示在该用例中一个对象在一段时间内的存在
- •垂直的虚线
- •如果对象生命期结束,则用注销符号表示



激活期(activation)/控制焦点(focus of control)

- 对象在一段时间内获得了焦点,也称激活期
- 对象执行某个动作的时期
- 空心矩形条
- 激活期的长短意味着对象执行某个动作的时间有多长,可以通过约束{10ms}来限制执行时间的长短。



顺序图中的消息

面向对象方法中,消息是对象间交互信息的主要方式。

结构化程序设计中,模块间传递信息的方式主要是过程 (或<mark>函数</mark>)调用。

对象A向对象

B发送消息,可以简单地理解为对象A调用对象B的一个操作(operation)。

- 顺序图中,尽力保持消息的顺序是从左到右排列的。
- 一个顺序图的消息流开始于左上方,消息2的位置比消息1低,这意味着消息2的顺序比消息1要迟。
- 顺序图中消息编号可显示,也可不显示。

UML1.4后定义的消息有:

- 调用消息 procedure call
- 异步消息 asynchronous
- 返回消息 return

UML1.3前的消息还有:

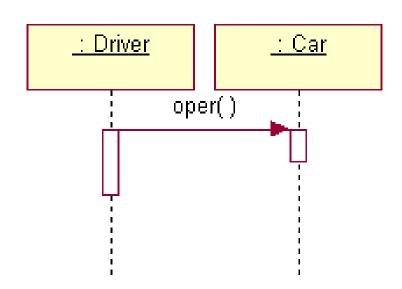
- 简单消息 simple

Rose扩充的消息

- 阻止消息 balking
- 超时消息 time-out

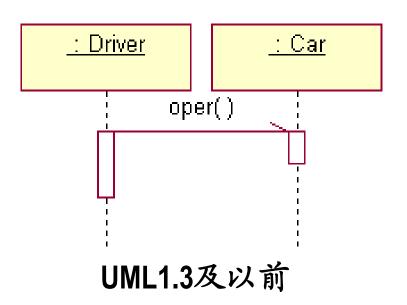
调用消息:消息的发送者把控制传递给消息的接收者,等待接收者返回或放弃控制,可以表示同步。

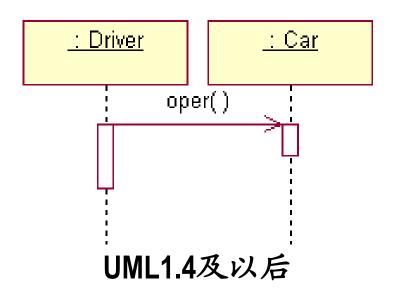
实心箭头符号



调用消息必有一个与之配对的返回消息,但是可以不用画出。

异步消息:消息的发送者把控制传递给消息的接收者,然后继续自己的工作,不等待接收者返回或放弃控制。

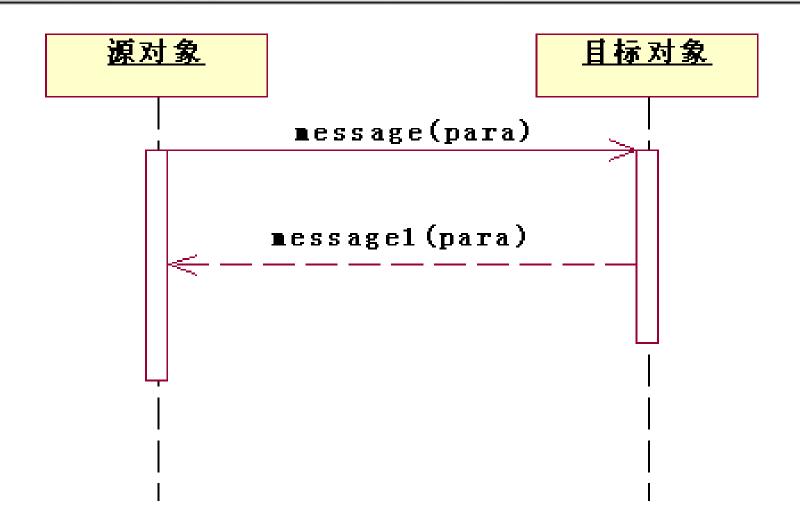




长同省自(Deturn)

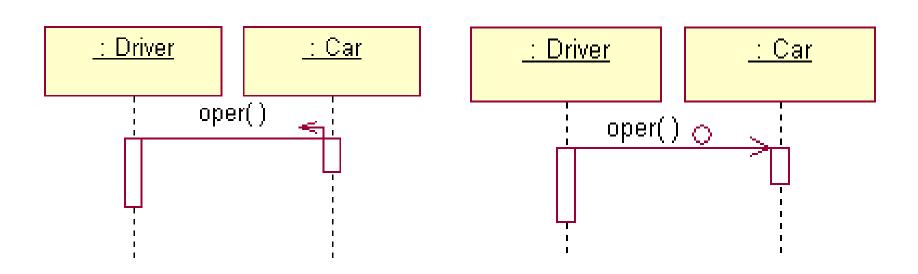
返回消息(Return):

- 表示消息的返回。消息上方放置返回值
- 同步消息的返回可以画出(如果想明确表达返回值),也可以不画出,直接隐含。
- 异步消息可以有返回,也可以没有。(可以响应异步消息, 也可以不响应该异步消息。)
- 虚线箭头表示,和依赖关系不要混淆



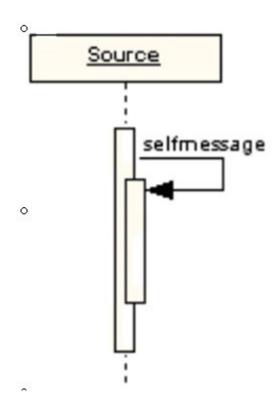
阻止消息:消息的发送者传递消息给接收者,如果接收者无法立即接收,则发送者放弃该消息。

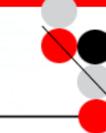
超时消息:消息的发送者发出消息给接收者并按指定时间等待,若接收者无法在指定时间内接收,则发送者放弃该消息。



自调用(Self Call)

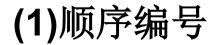
- 某对象自己调用自己的操作
- 嵌套的矩形条





消息编号:

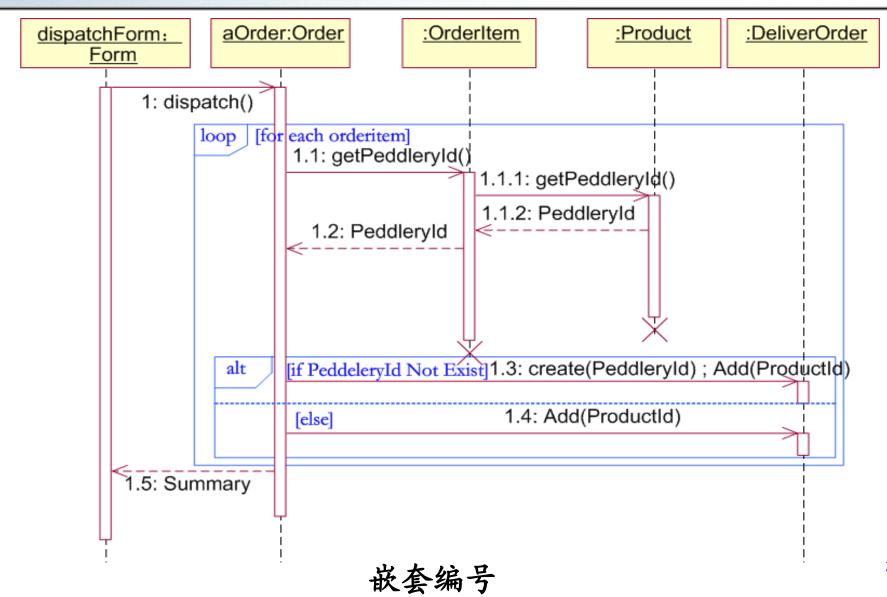
按消息产生的先后顺序,给消息编号,有两种编号方案,一种是顺序编号,另一种是嵌套编号。



整个消息的传递过程就形成了一个完整的序列,因此通过在每个消息的前面加上一个用冒号隔开的顺序号(按照消息的先后顺序,从1开始对消息编号)来表示其顺序。

(2) 嵌套编号

由于顺序图本身已经充分表现出消息执行的顺序,因此在UML模型中并没有要求对消息进行编号。只是要求在协作图中,对消息进行了编号。但在Rose等建模工具中,为了能够自动实现顺序图与协作图的转换,在顺序图中也默认采取嵌套编号方案。



顺序图的特点

- 顺序图是一个二维图形。在顺序图中水平方向为对象维,沿水平方向 排列参与交互的对象;竖向方向为时间维,沿垂直向下方向按时间递 增顺序列出各对象所发出和接收的消息。
- 水平轴上的对象间的相互顺序并不重要。
- 顺序图不表示对象间的关联(associations)关系。

建立顺序图的步骤

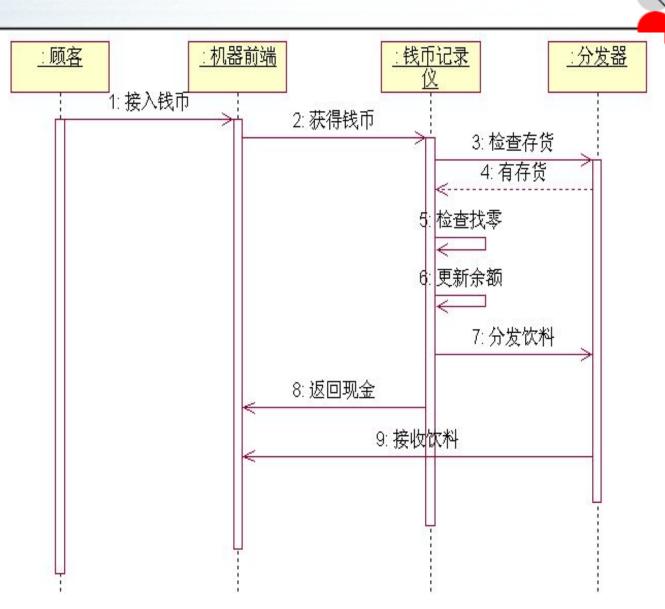
- 1. 确定交互过程的上下文(context);
- 2. 识别参与交互过程的对象;
- 3. 为每个对象<mark>设置生命线</mark>,即确定哪些对象存在于整个交互过程中 ,哪些对象在交互过程中被创建和撤销;
- 4. 从引发这个交互过程的初始消息开始,在生命线之间从顶到下依 次画出随后的各个消息;
- 5. 如果需要表示消息的嵌套,或/和表示消息发生时的时间点,则采 用控制焦点;
- 6. 如果需要说明时间约束,则在消息旁边加上约束说明;
- 7. 如果需要,可以为每个消息附上前置条件和后置条件。

阅读顺序图

· 下面以饮料自动销售系统为例,对"买饮料"的3种场景进行建模, 对每一个场景,我们绘制其对应的顺序图。

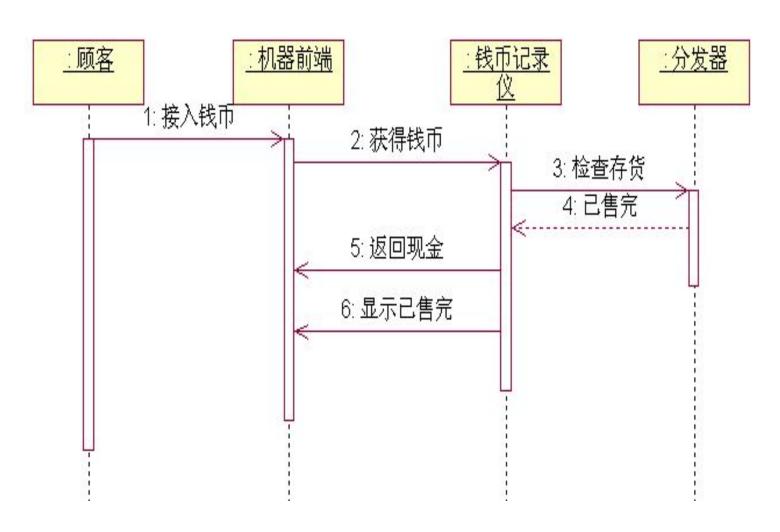
阅读顺序图

- 买饮料的正常场景
 - 顾客从机器的前端钱币口投入钱币,然后选择想要的饮料;
 - + 钱币到达钱币记录仪更录仪,记录仪更新自己的存储;
 - 记录仪通知分配器分发饮料到机器前端;



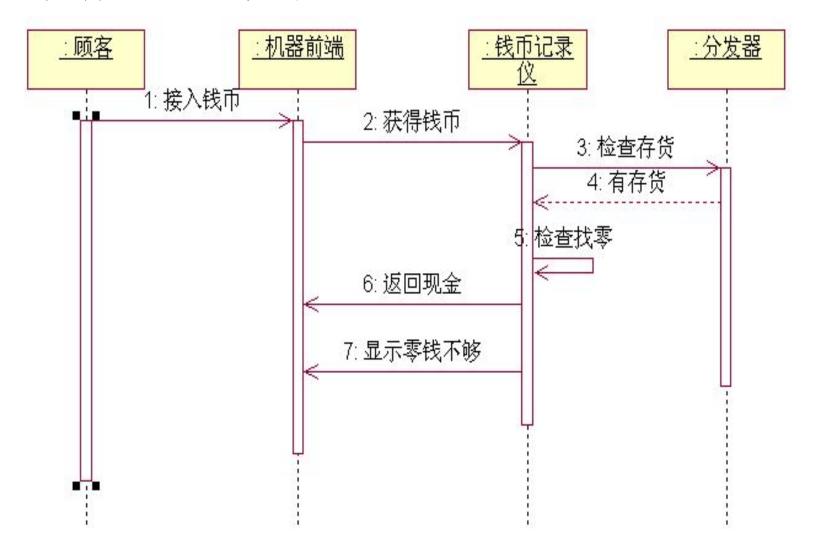
阅读顺序图

• 饮料"已售完"的场景

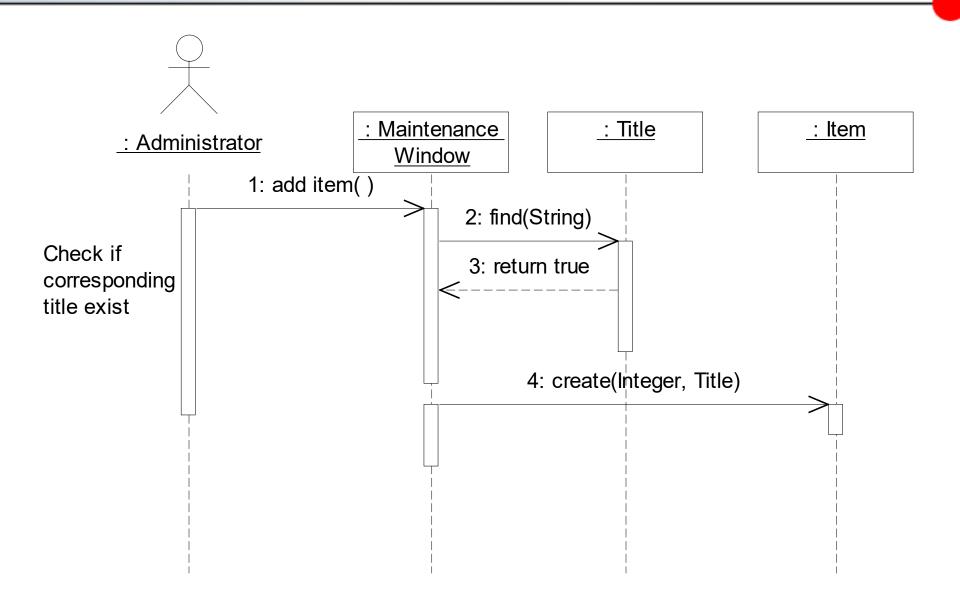


阅读顺序图

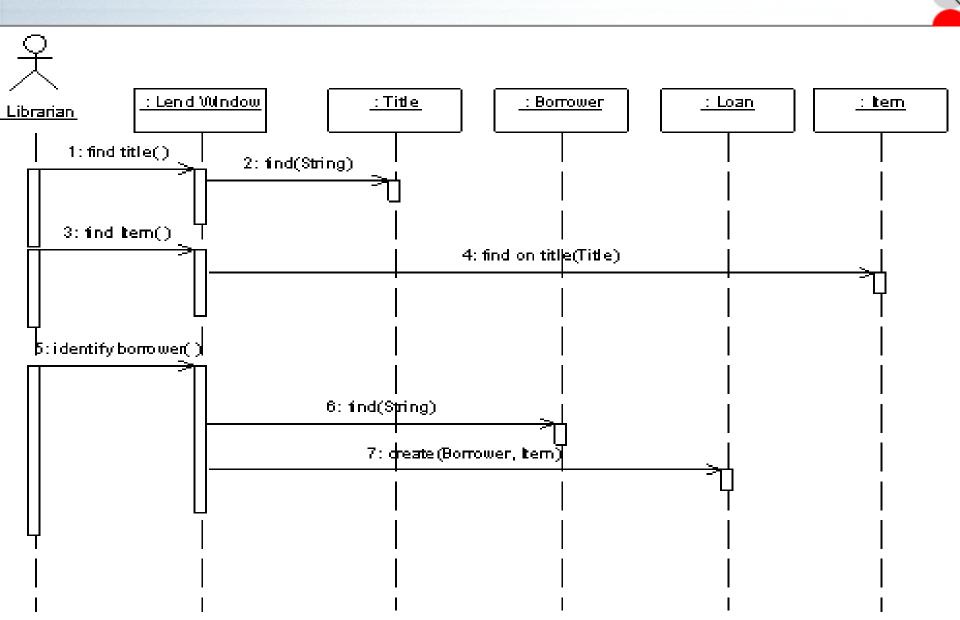
·机器没有合适的零钱



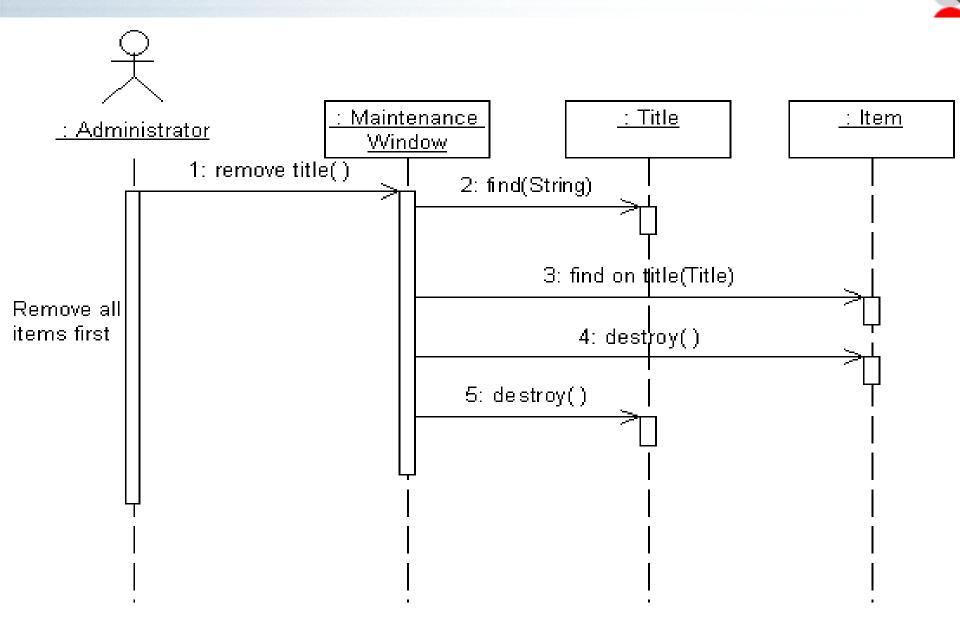
阅读顺序图——添加书籍的顺序图



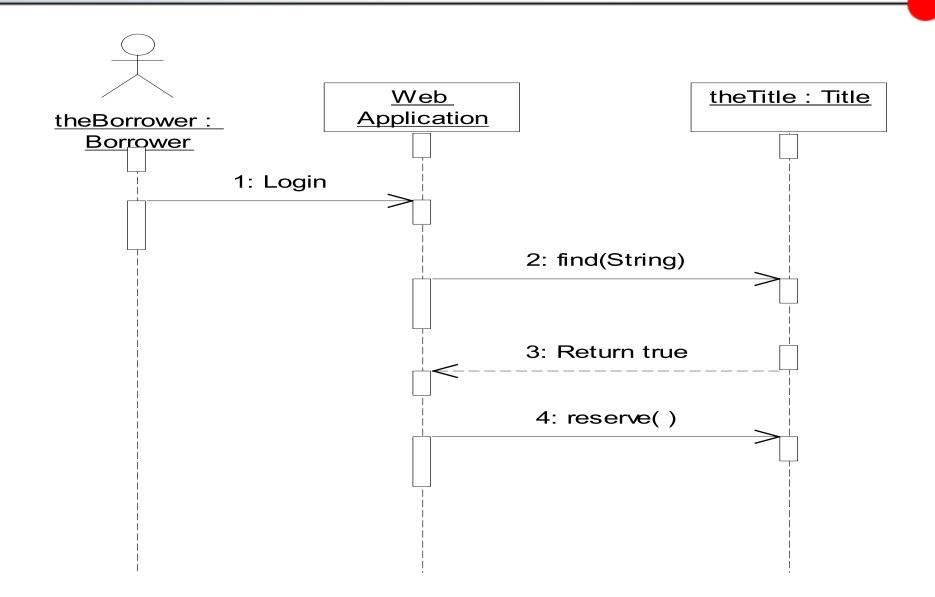
阅读顺序图——借书的顺序图



阅读顺序图——删除书目的顺序图



阅读顺序图——借阅者预留书籍的顺序图

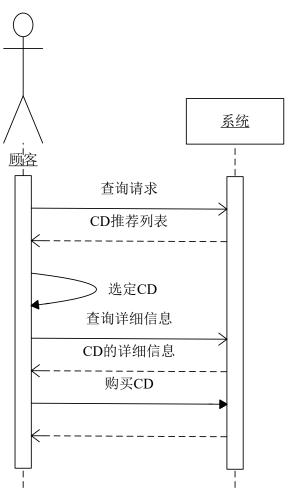


系统顺序图

- 将整个系统看作一个黑箱的对象,强调外部参与者和系统的 交互行为,重点展示系统级事件

用例描述:

- 1. 顾客向系统提起查询请求
- 2. 系统根据请求为顾客提供一个CD的推荐列表
- 3. 顾客在推荐列表中选定一个CD, 然后要求查 看更详细的信息
- 4. 系统为顾客提供选定CD的详细信息
- 5. 顾客购买选定CD.
- 6. 顾客离开.

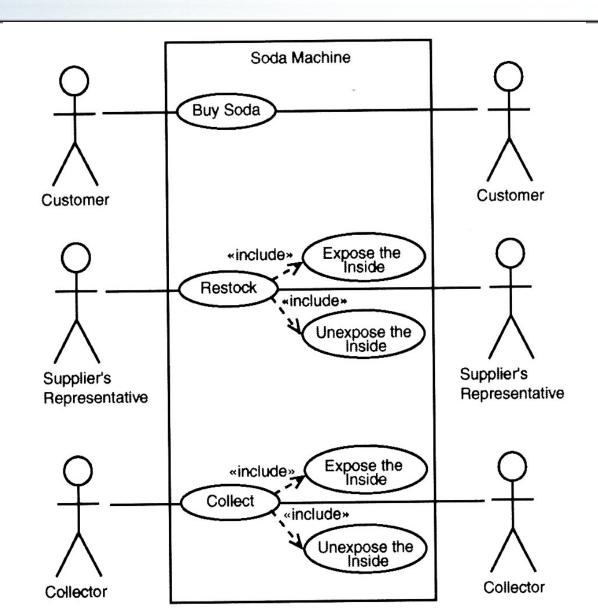


顺序图:从用例到类

- 顺序图使你可以可视化地对系统逻辑建模。
- 对象、类和参与者都在顺序图中进行了描述。
- 理解分析阶段的基本逻辑,在设计阶段详细将其突出出来。

实例: 饮料销售机系统

用例图



• 用例阐述

- "购买饮料"用例的主要场景:
- -1) 顾客从投币口塞入钱币,然后选择想要的饮料;
- 2) 系统将钱币送入钱币记录仪;
- 3) 记录仪检查是否还有存货;
- 4) 记录仪更新自己的钱币存储记录;
- 5) 记录仪通知饮料分配器传送一罐饮料到出货口。

用例阐述:

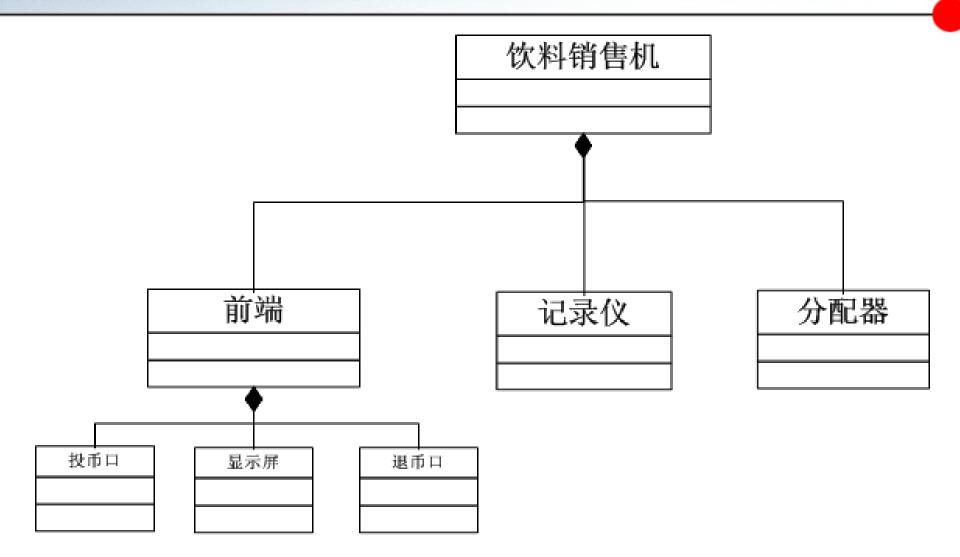
"购买饮料"用例的次要场景1—饮料已售完

- 1) 若饮料已售完,记录仪要求显示屏显示"已售完"
- 2) 记录仪将钱币从退币口返回给顾客
 - "购买饮料"用例的次要场景2—需要找零
- 1) 记录仪查找自己的现金储备以便找零;
- 2) 记录仪更新自己的钱币存储记录;
- 3) 记录仪将找回的钱通过退币口返还给顾客;
- 4)记录仪通知饮料分配器传送一罐饮料到出货口。

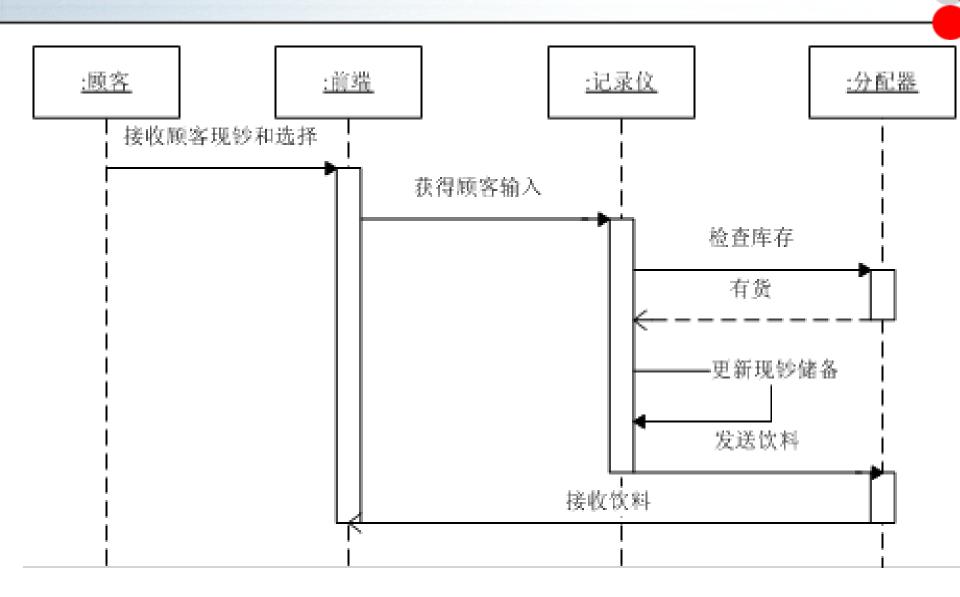
"购买饮料"用例的次要场景3—零钱找不开

- 1) 记录仪查找自己的现金储备以便找零;
- 2) 如果无法找零,记录仪要求显示屏显示"投入正好的货币"
- 3) 记录仪将钱币从退币口返回给顾客

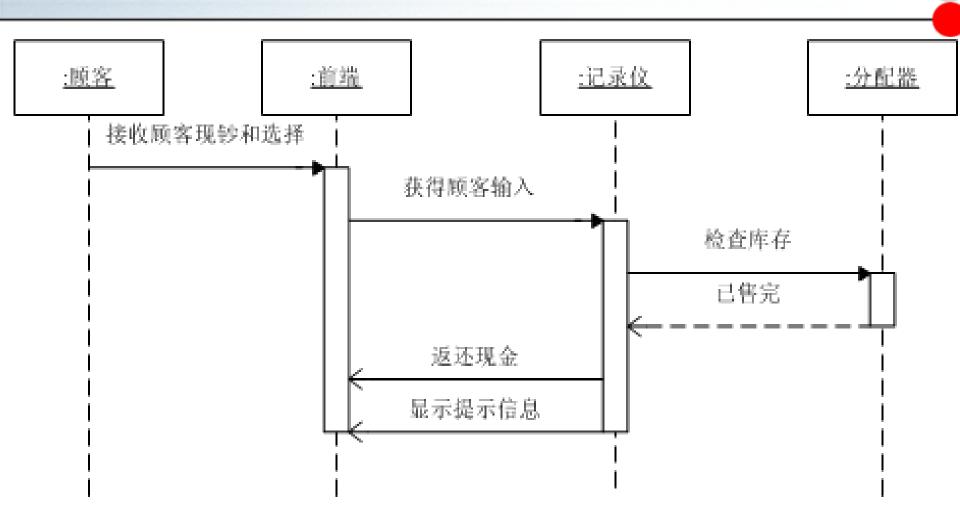
初步类图



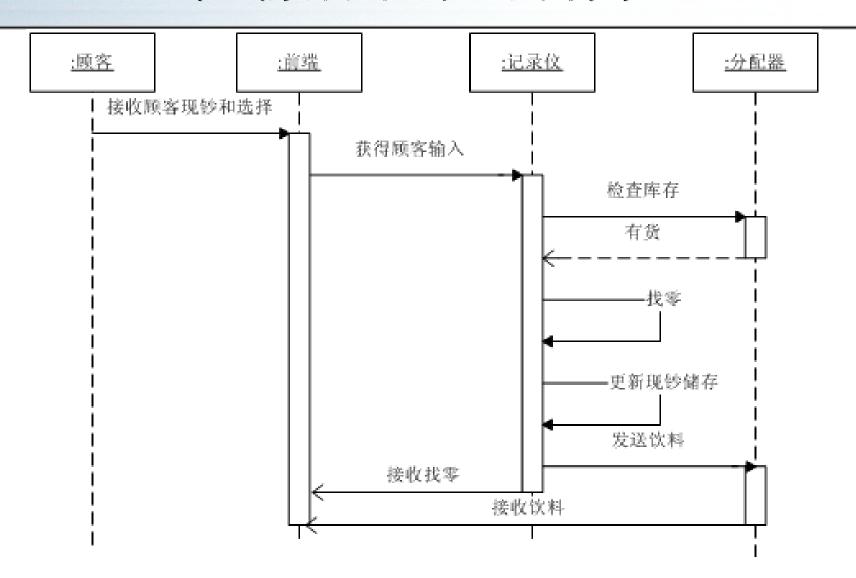
购买饮料主要场景的顺序图



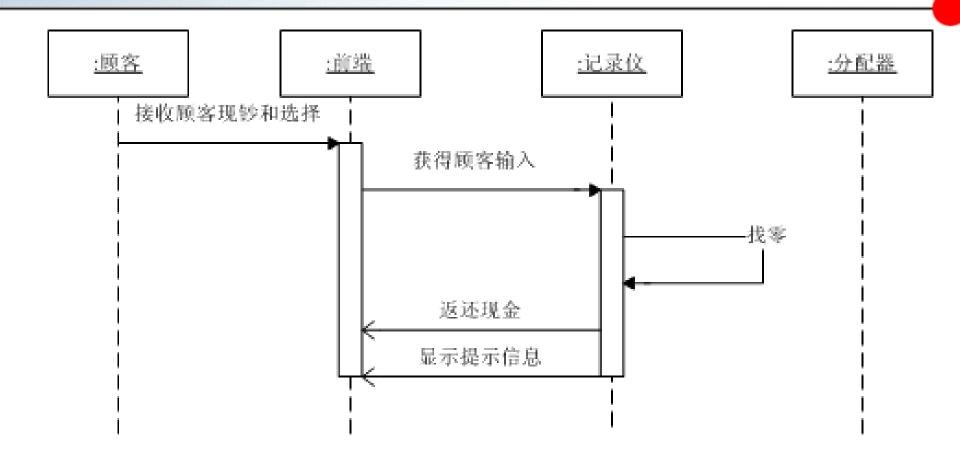
己售完场景的顺序图



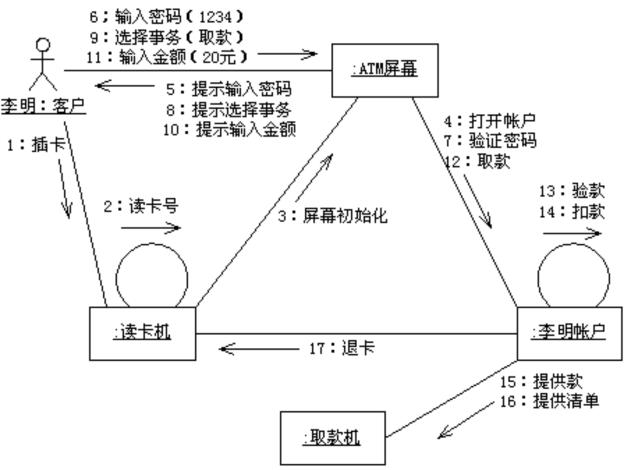
"零钱数目不对"的场景



"零钱找不开"场景的顺序图



◇ 协作图



协作图和顺序图合称为 交互图。协作图跟顺序 图相似,显示对象间的 动态合作关系。

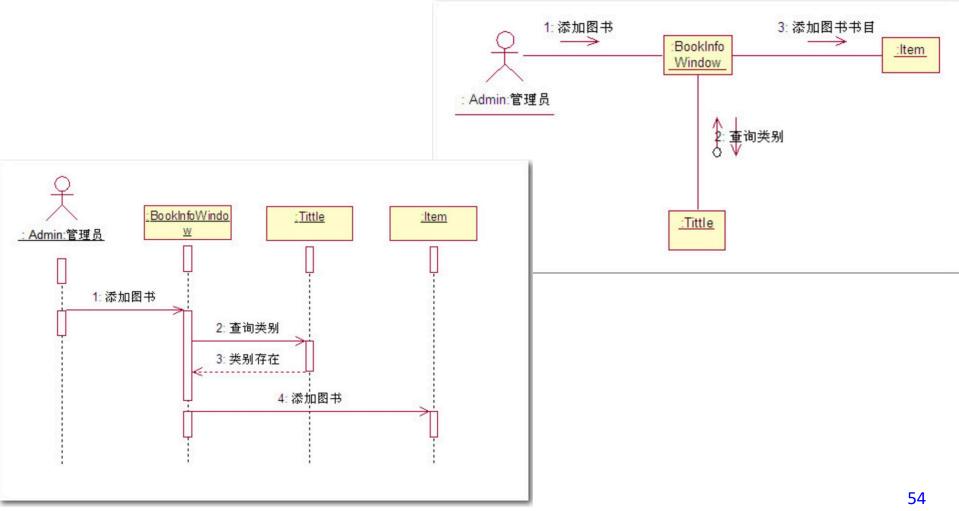
顺序图着重描述对象按 时间顺序的消息交换; 协作图着重描述系统成 分如何协同工作。

- 协作图和顺序图一样,都是用来展示对象之间的交互关系的,都属于交互图。
- 对象图是展现对象之间的静态关系,而协作图是对象图的扩展 ,协作图除了展现对象之间的关联,还显示对象之间的消息传 递。
- 协作图和顺序图很相似,在语义上是等价的,用两种图表达的同一种信息,并且可以将此两种图进行等价转换。

实例: 图书管理系统

涉及到的对象 : Admin (管理员), BookInfoWindow (图书信息窗口), Tittle (图书类别), Item (图书条目);

流程 : 管理员 查询 图书信息窗口, 查看有没有相应的 图书类别, 如果有 插入该图书条目;

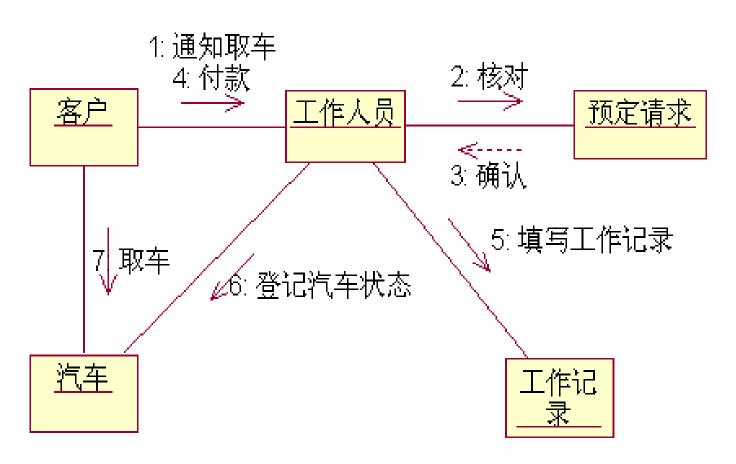


- 协作图和顺序图类似也是描述对象之间的交互,强调 了对象在交互行为中承担的角色。顺序图是从时间的 角度,协作图是从空间的角度。
- 顺序图强调的是交互中消息传递的时间顺序。
- 协作图强调的是交互对象的整体组织(的链接及其消息传递)。

• 协作图的作用

- 一 协作图常用来描述业务或软件系统中,每个对象在交互发生时承担 的角色,即强调了交互发生时,每个对象承担的职责。
- 使用协作图可以显示对象相互协作时充当的角色。如果需要强调时间和序列,最好选择顺序图建模;如果需要强调上下文相关,最好选择协作图建模。
- 协作图用于显示对象之间如何进行交互,以实现特定用例或用例中特定部分的行为。设计员使用协作图和顺序图确定并阐明对象的角色,这些对象执行用例的特定事件流。这些图提供的信息主要用来确定类的职责和接口。

• 汽车租赁系统中客户取车的协作图



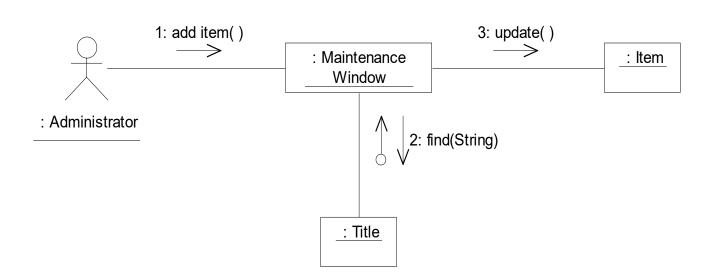
• 协作图的组成元素

- 协作图的组成元素包括对象、消息、链(连接器)。消息表示了对象间的通信,对象通过链连接在一起。

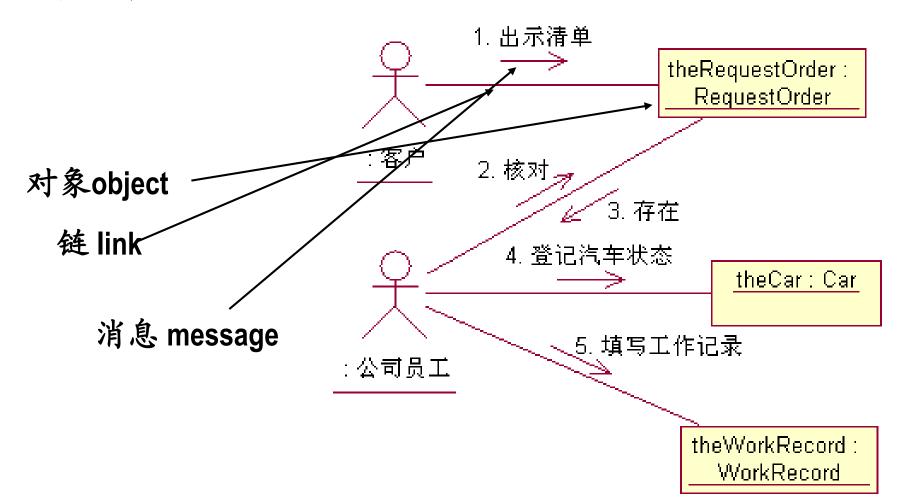
• 协作图的表示

- UML中,表示一个协作图,主要是标识系统中的对象、对象间交互的消息、对象间的链。第一个消息(Additem())表示,管理员要求维护窗口添加书籍;第二个消息(find(String))表示,维护窗口要求:Title对象根据书名获得书的目录;第三个消息(update())表示修改书目下书的数量。

系统管理员添加 书籍的协作图:



协作图中的元素:



协作图中的元素

• 对象:

一协作图与顺序图中的对象的概念是一样,只不过在协作图中 ,无法表示对象的创建和撤销,所以对于对象在图中的位置 没有限制。

• 链:

 表示对象之间的语义关系,链是关联的一个实例。协作图中 链的符号和对象图中链所用的符号是一样的,即一条连接两 个对象的实线。

消息

 - 协作图中的消息类型与时序图中的相同,只不过为了说明交 互过程中消息的时间顺序,需要给消息添加顺序号。顺序号 是在消息的前面加一个整数。每个消息都必须有唯一的顺序 号。

协作图中的元素

• 消息编号:

- 消息的编号有两种,一种是无层次编号(按顺序编号),它简单 直观,另一种是嵌套的编号,它更易于表示消息的包含关系.

• 迭代标记

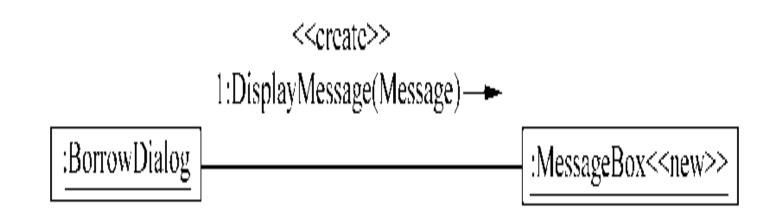
- 迭代标记用*表示,表示循环,通常还有迭代表达式,用来说明循环规则

迭代表达式	语义
[i:=1n]	迭代n次
[I=110]	I迭代10次
[while(表达式)]	表达式为true时才进行迭代
[until(表达式)]	迭代到表达识为true时,才停止迭代
[for each(对象集合)]	在对象集合上迭代

• 监护条件

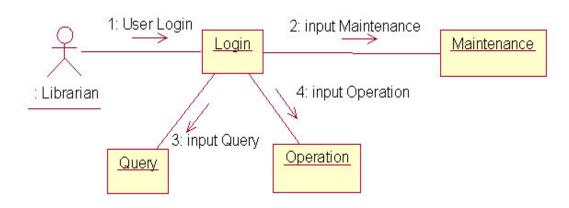
监护条件通常用来表示分支,也就是表示"如果条件为 true,才发送消息"的语义,在UML中,监护条件是以" 【条件表达式】"的格式表示的。

 与顺序图中的消息相同,消息也可以用来在协作图中创建对象。 为此,一个消息将会发送到新创建的对象实例。对象实例使用 new固化类型,消息使用create固化类型,以明确指示该对象是 在运行过程中创建的。



阅读协作图

• 下图是图书管理员登录系统的协作图。



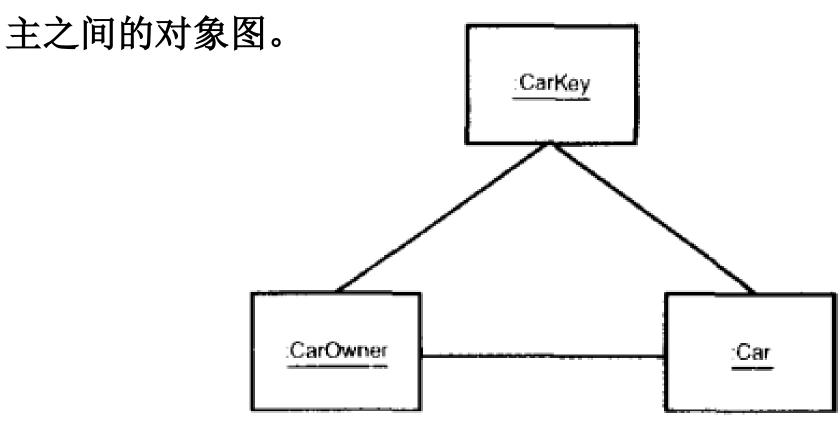
- 由Librarian角色向Login发送UserLogin信息,要求登录系统
- 登录界面(Login)向Maintenance发送发送Input Maintenance的信息,做一些维护工作;
- 登录界面(Login)向Query角色发送inputQuery消息,做一些查询工作;
- 登录界面(Login向Operation发送inputOperation信息,进行一些输入操作。

通信(协作)图基本概念

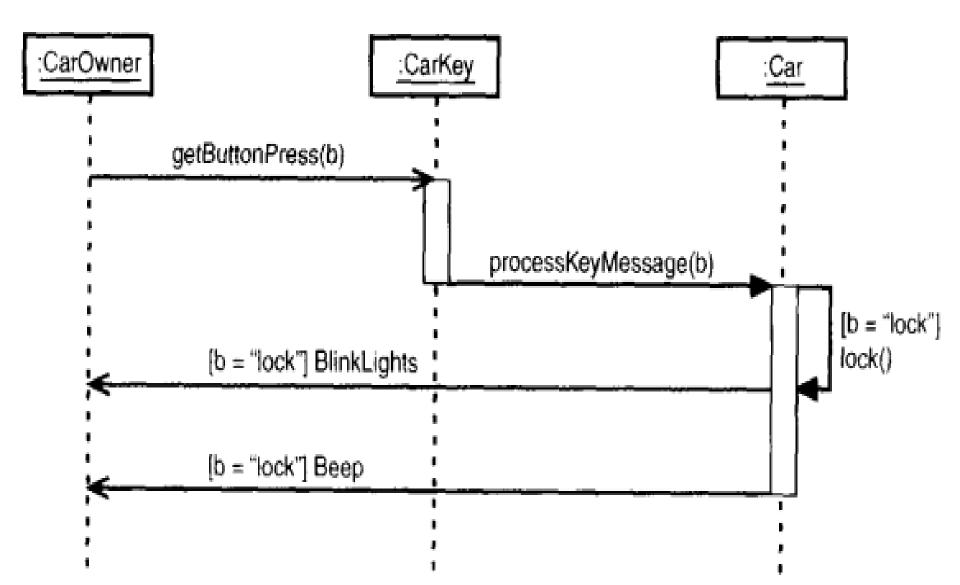
- 顺序图与协作图在语义上具有等价地位,也就是说可以从一种图转换为另一种图,而不丢失任何信息,但两者侧重点不同。顺序图清楚表达了交互作用中的时间顺序,协作图清楚表示了对象之间的关系,但时间顺序从序号上获得。
- 顺序图经常用于用例的描述和详细设计时对象之间的关系的描述。 协作图多用于详细设计过程中描述对象之间的关系。

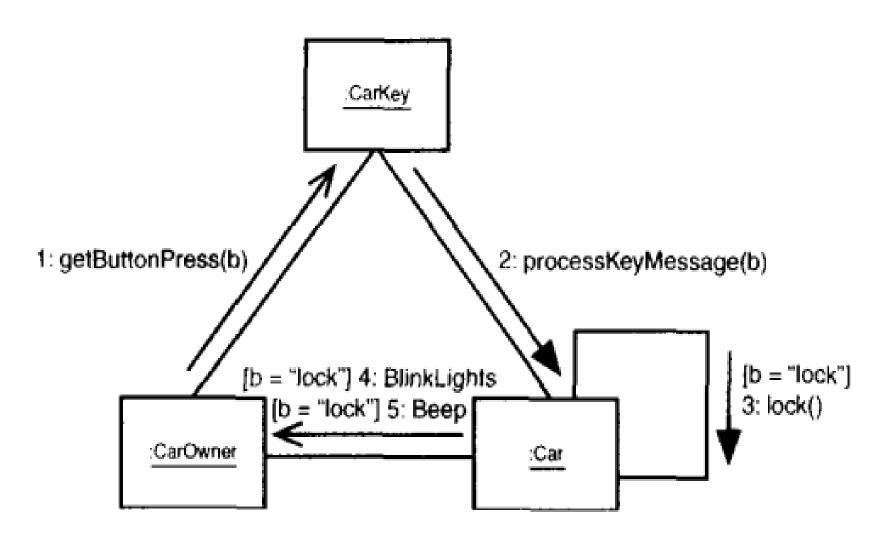
类图

• 汽车与车钥匙以及车

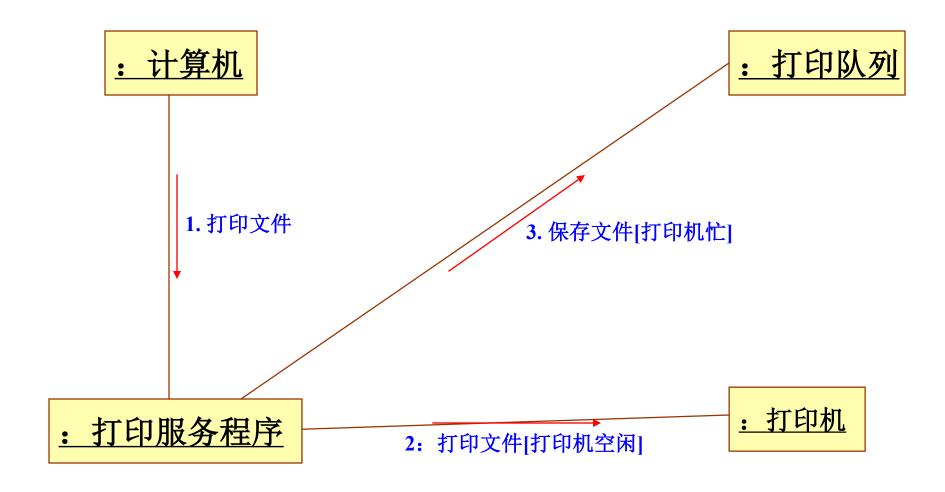


顺序图

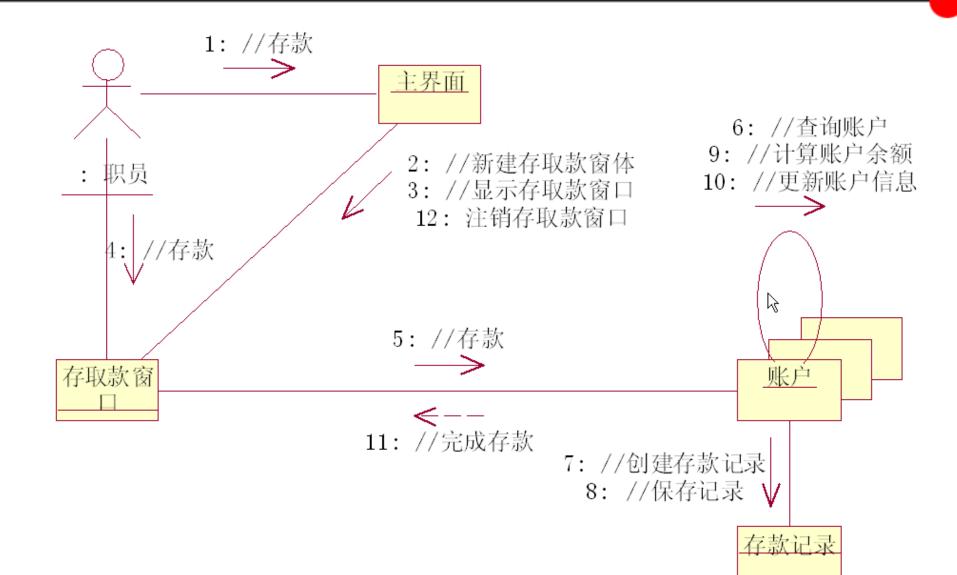




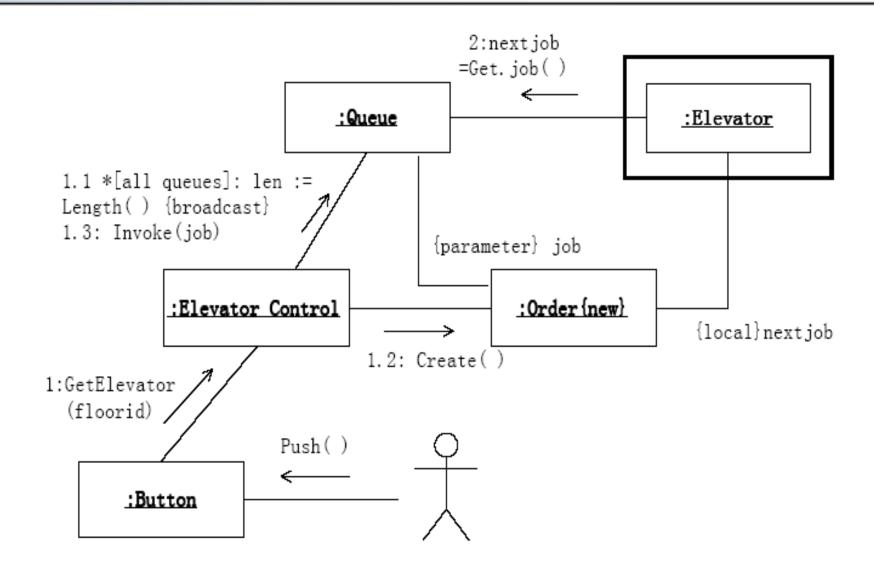
阅读协作图



阅读协作图-存款

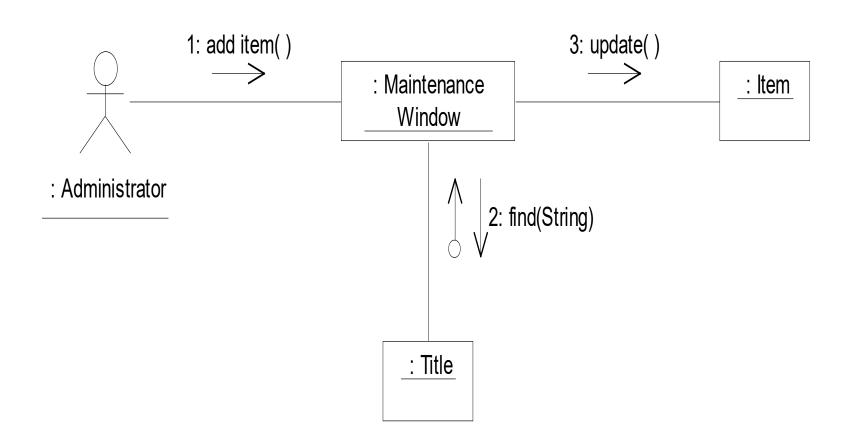


阅读协作图——电梯

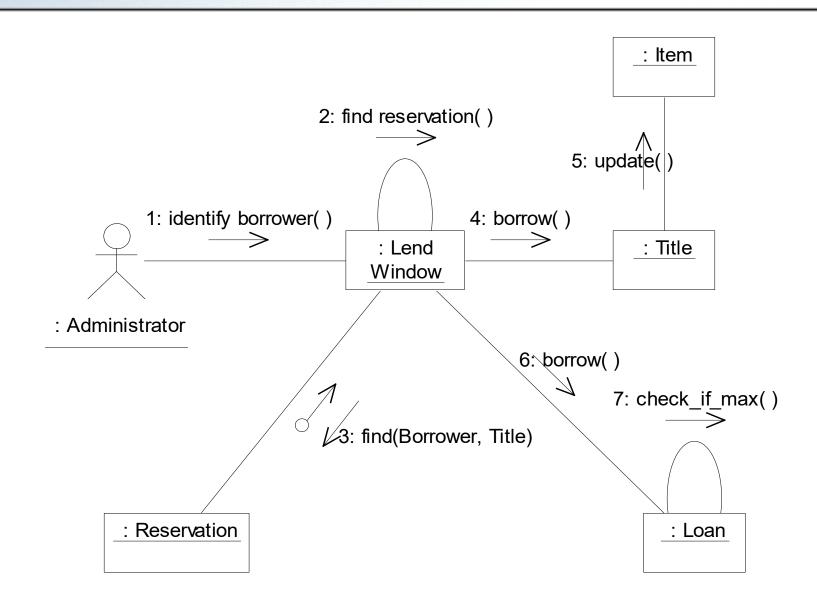


阅读协作图——图书馆管理系统

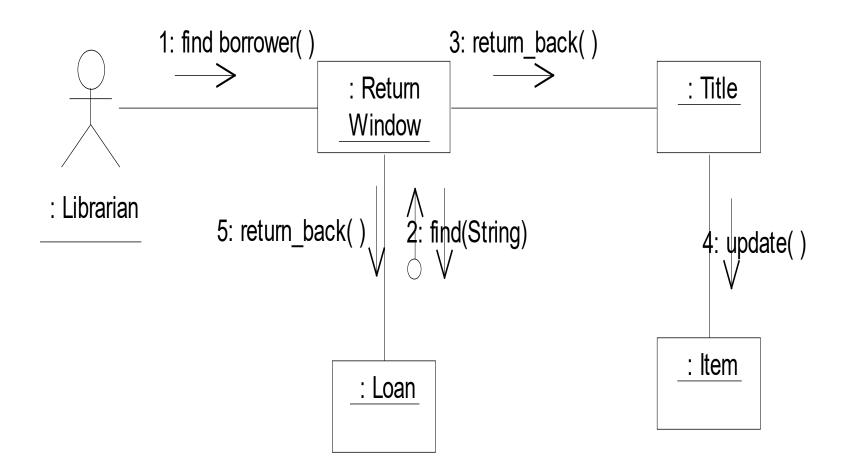
添加书籍协作图



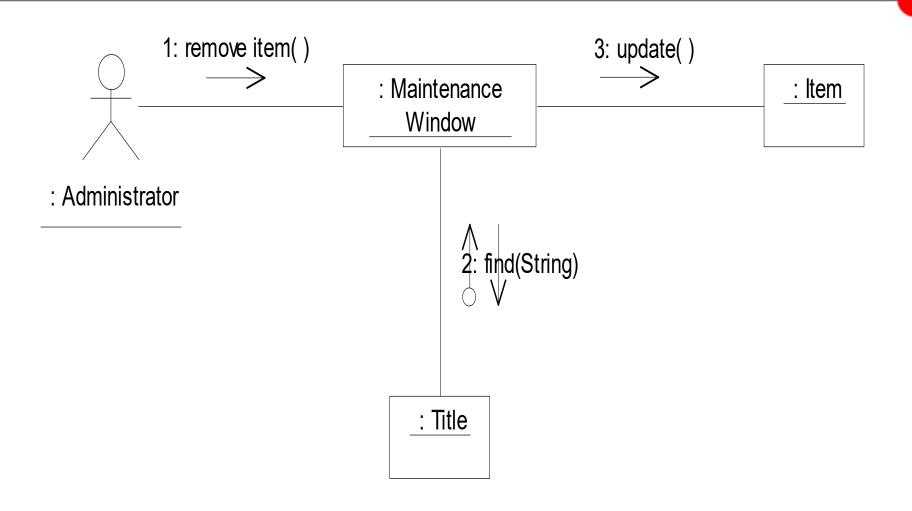
阅读协作图——借书



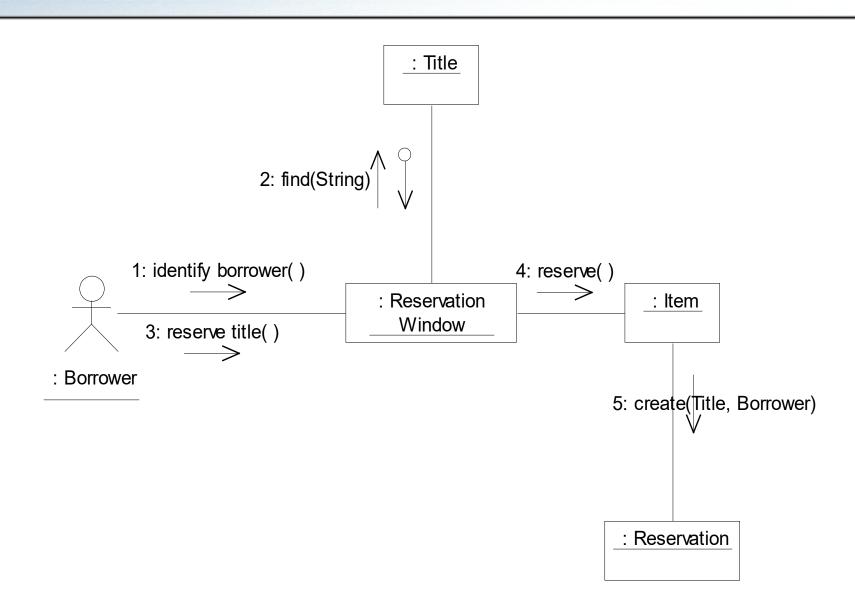
阅读协作图——还书



阅读协作图——删除书籍



阅读协作图——预留书籍



建立协作图的步骤

- 1. 确定交互过程的上下文
- 2. 识别参与交互过程的对象
- 3. 如果需要,为每个对象设置初始特性
- 4. 确定对象之间的链, 及沿着链的消息
- 5. 从引发该交互过程的初始消息开始,将每个消息附到相应的链上
- 6. 如果需要表示消息嵌套,则有Dewey十进制表示
- 7. 如果需要说明时间约束,则在消息旁边加上约束说明
- 8. 如果需要,可以为每个消息设置前置条件和后置条件

• 建模分数记录系统的协作图。具体用例是Teacher 查看学生成绩。





可以通过更加具体的描述来确定工作流程,基本工作流程如下: 张老师希望通过系统查询某名学生的学科成绩。

- ① 张老师通过用户界面录入学生的学号以及学科科目请求学生信息
- ② 用户界面根据学生的学号向数据库访问层请求学生信息
- ③ 数据库访问层根据学生的学号加载学生信息
- ④ 数据库访问层根据学生信息和学科科目获取该名学生的分数信息
- ⑤ 数据库访问层将学生信息和分数信息提供给用户界面
- ⑥ 用户界面将学生信息和分数信息显示出来。

1.确定协作图的元素



: WebInterface

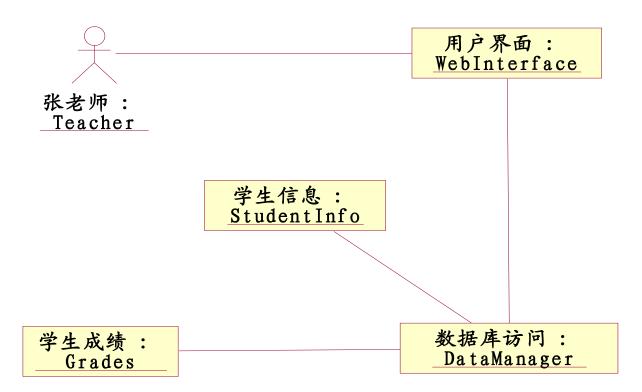
: StudentInfo

: Grades

: DataManager

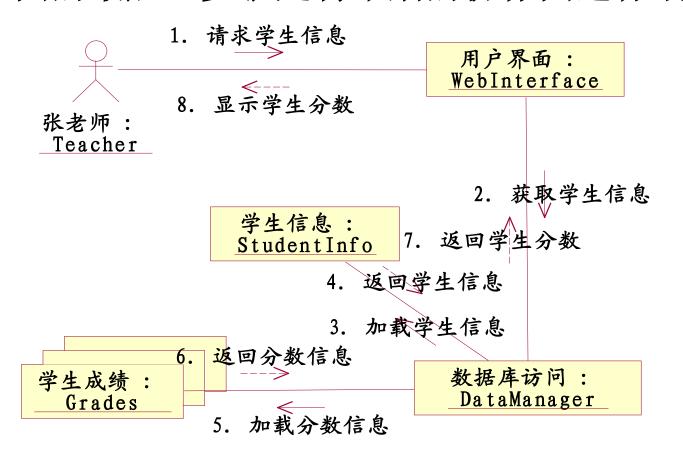
2.确定元素之间的结构关系

创建协作图的下一步是确定这些对象之间的连接关系,使用链与角色将这些对象连接起来。



3.细化协作图

创建协作图的最后一步就是将早期的协作图进行细化。



顺序图与协作图的比较

- 顺序图和协作图都属于交互图,用于描述系统中对象之间的动态关系。
- 顺序图和协作图在语义上是等价的,两者可以相互转换。
- 协作图强调参与交互的对象的组织结构; 顺序图强调消息的时间顺序。
- 顺序图可以反映对象的创建/激活/撤消: 协作图不能。
- 协作图能反映动作路径,消息必须有顺序号;顺序图没有。

组成元素

顺序图的组成元素

- ① 参与者(actor)或者 对象(object)
- ② 生命线 (lifeline)
- ③ 激活期(activation)/ 控制焦点(focus of control)
- ④ 消息 (message)

协作图的组成元素

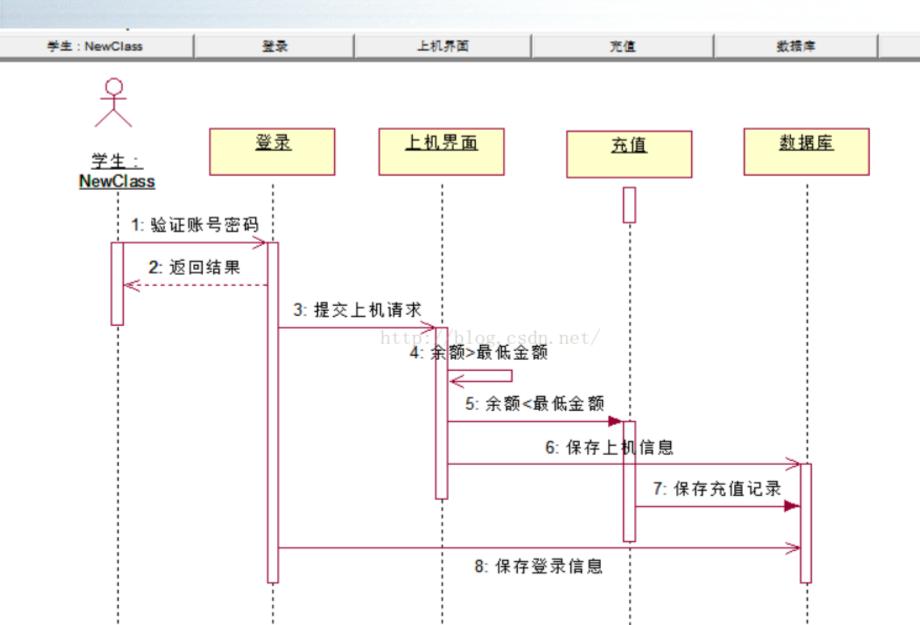
- ① 对象
- ② 链
- ③ 消息



消息编号

- 在交互图中可以选择显示消息编号。
- 在顺序图中,消息可以按两种方式编号: Top-Level(顶级编号)方式,或者Hierarchical(嵌套编号)方式。
- 在协作图中,消息只能采用顶级编号,但如果协作图是由顺序图转换而来,图中也可以使用嵌套编号。

机房收费系统关于上机的顺序图



机房收费系统关于上机的协作图

