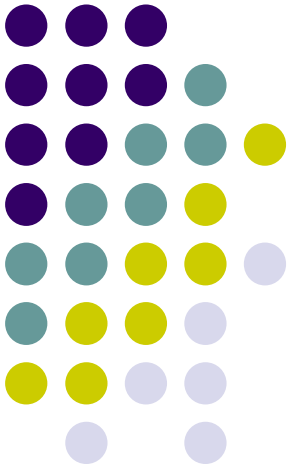
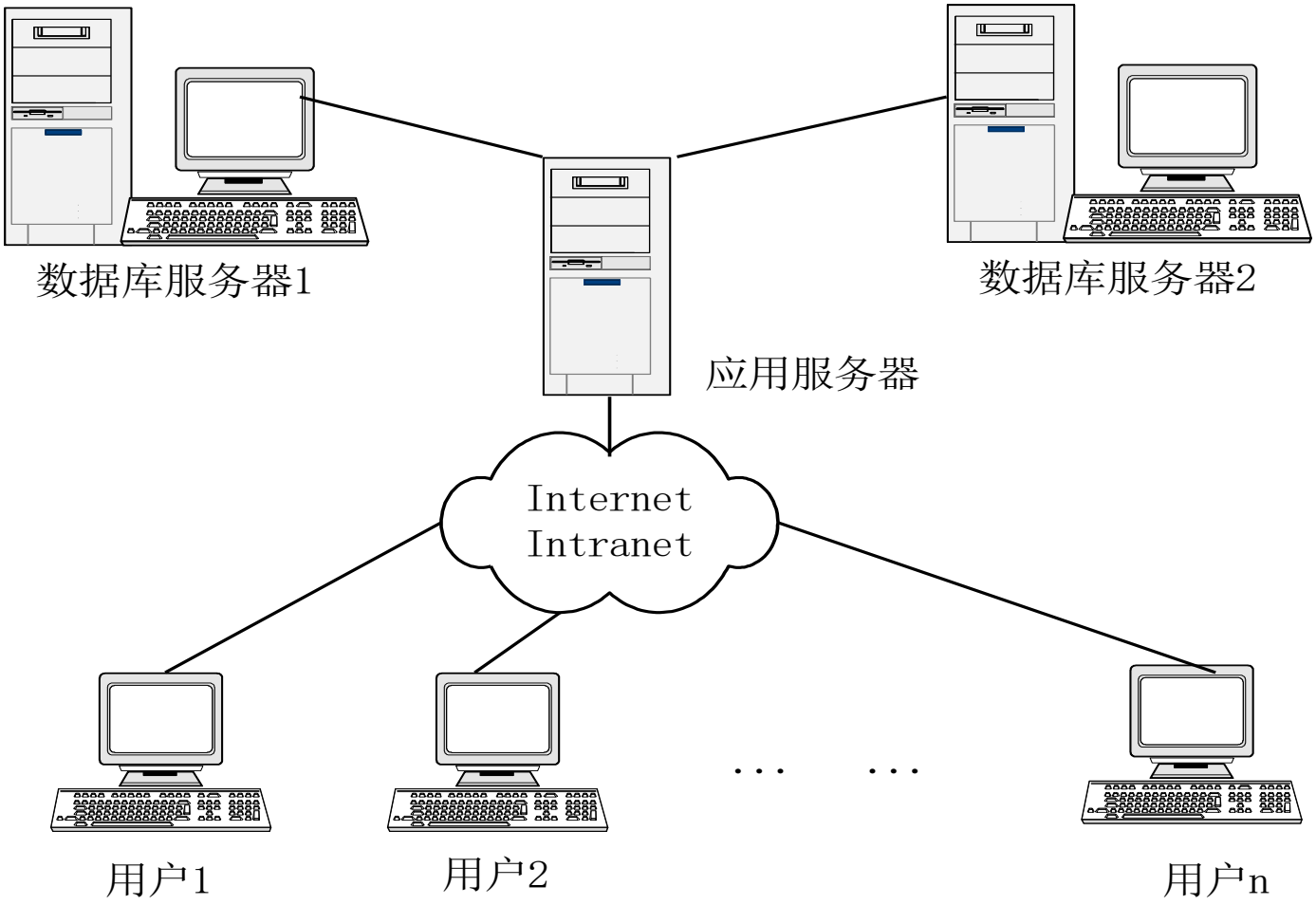
软件系统分析与设计

Lecture 6 Dynamic modeling (1)

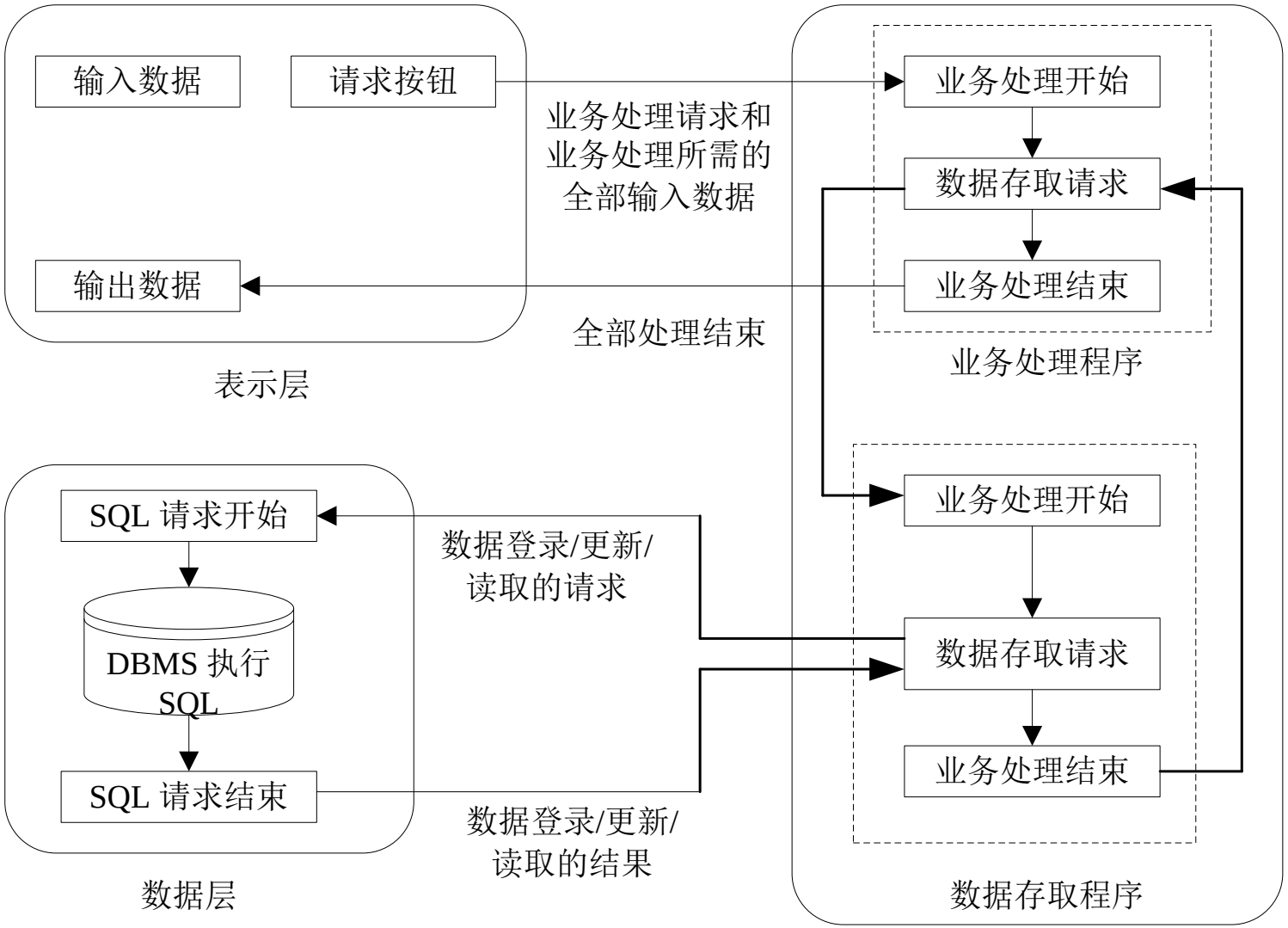


by Dr Y.Yang
Associate Professor
School of Computer Science & Technology
Soochow University
yyang@suda.edu.cn

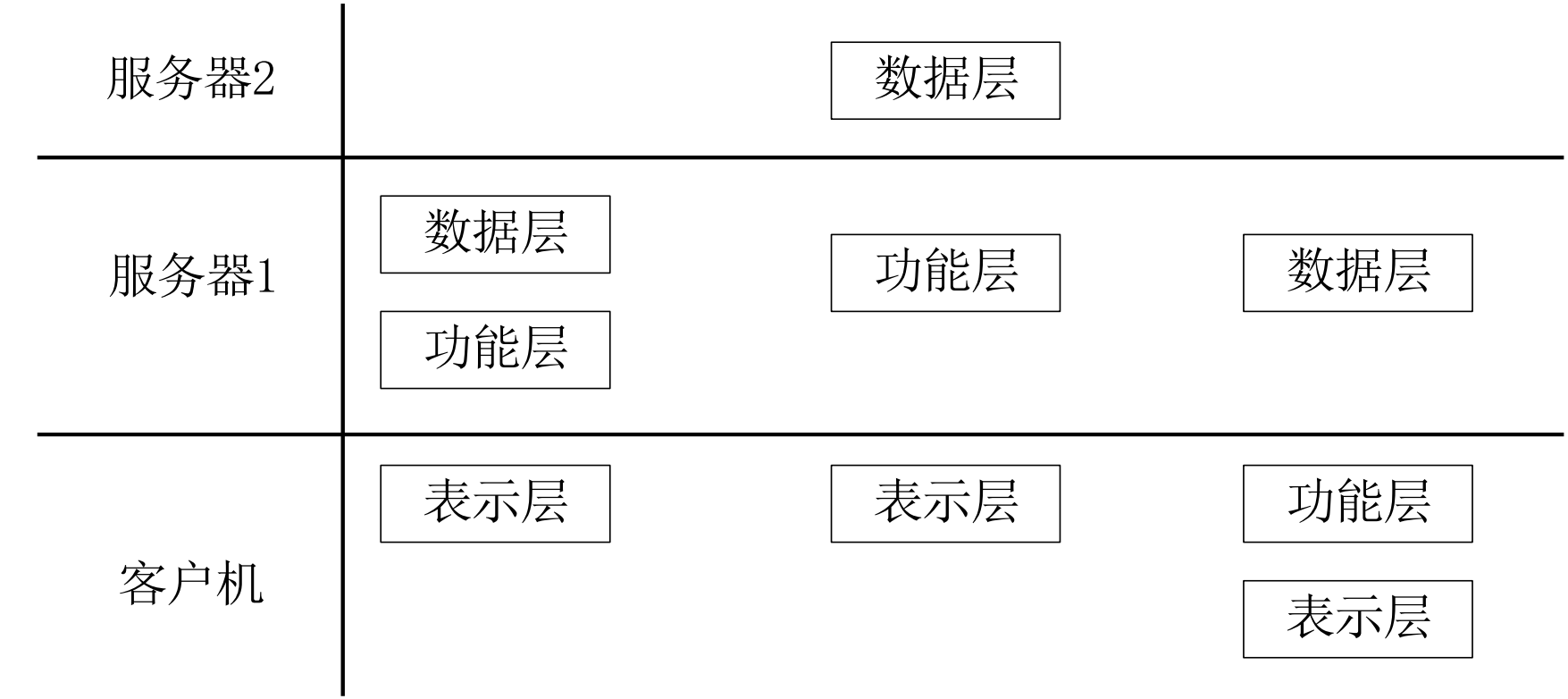
关于三层架构的一点补充说明



三层C/S结构的一般处理流程



物理结构

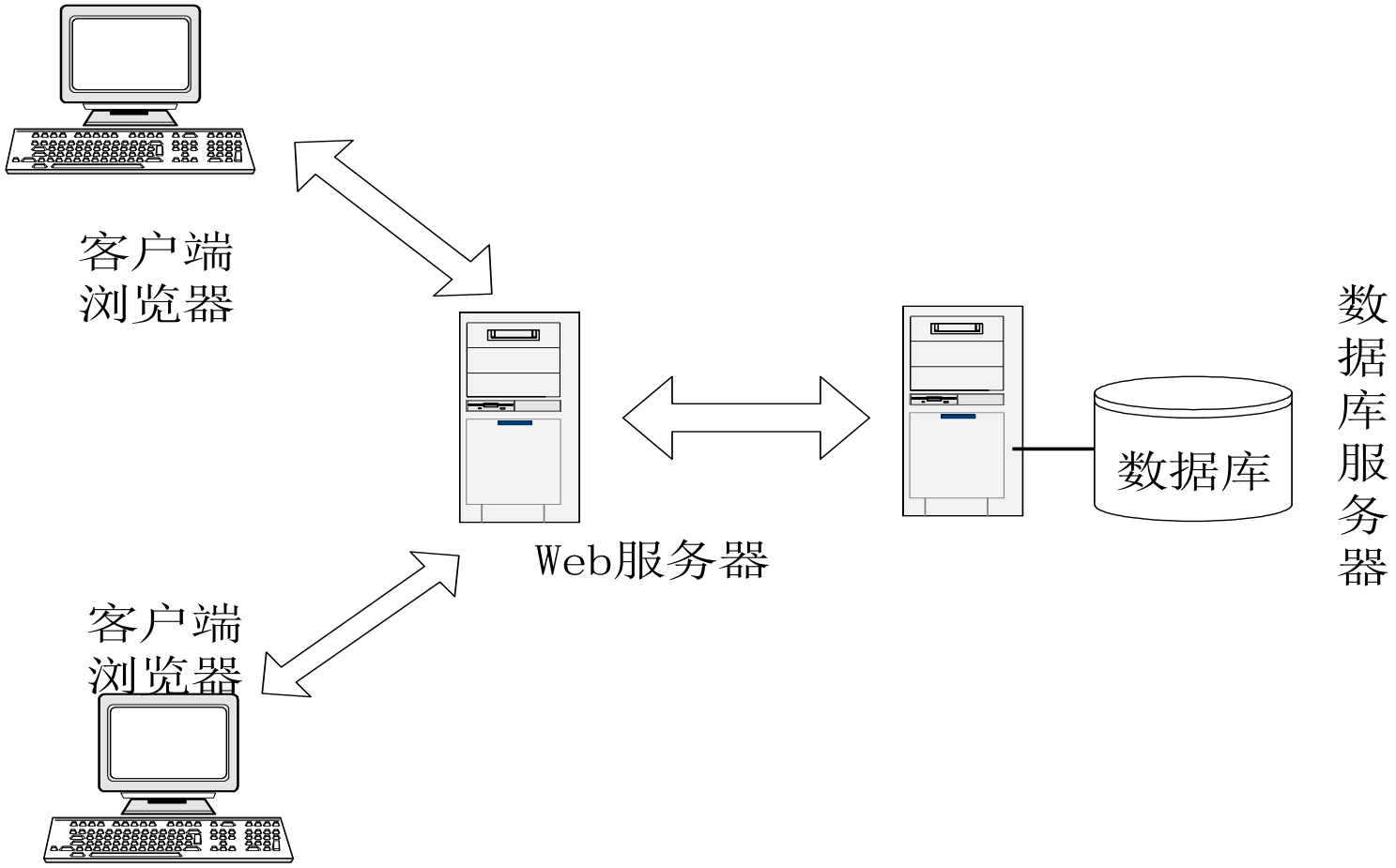


（1）将数据层和功能层放在同一台服务器上

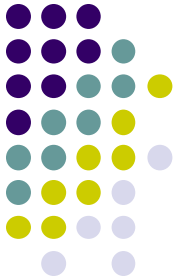
（2）将数据层和功能层放在不同的服务器上

（3）将功能层放在客户机上

B/S结构

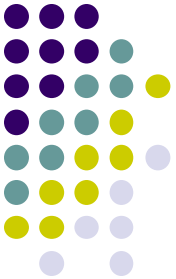


Review



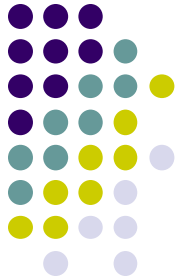
- 寻找和确定一个系统的对象类
- **UML**中类和对象类属性的定义和描述方法
- **UML**中类和对象类操作的定义和描述方法
- 对象类之间的关系及其表示方法
- 几个典型的设计样式
- 设计样式与软件开发关系
- 如何描述包和子系统的层次关系
- **UML**中接口的描述方法
- 类和对象建模方法

Today's Topics



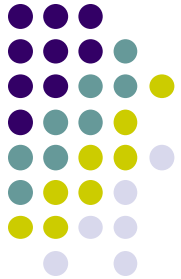
- 了解系统中传递的消息类型和表示方法
- 掌握UML中时序图的描述方法
- 掌握同步消息和异步消息的定义和描述方法
- 掌握UML中协作图的描述方法
- 掌握时序图和协作图的区别

引言



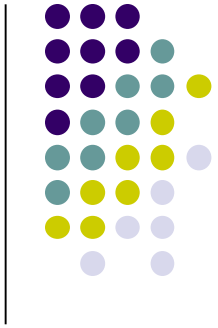
- UML中的动态模型由时序图、协作图、状态图和活动图组成。
- 时序图和协作图用来表达对象之间的交互，是描述一组对象如何合作完成某个行为的模型化工具
- 时序图和协作图描述系统行为的角度不同。
  - 时序图主要描述对象之间信息交换的时间顺序
  - 协作图用来描述系统成分之间如何协作共同完成系统功能要求
  - 两者之间互为补充并可以相互转化

消息

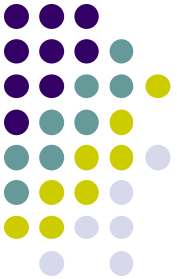


- 在面向对象技术中，对象间的交互是通过对象间消息的传递来完成的。在**UML**的四个动态模型中均用到消息这个概念。
- 通常，当一个对象调用另一个对象中的操作时，即完成了一次消息传递。
- 当操作执行后，控制便返回到调用者。
- 对象通过相互间的通信(消息传递)进行协作，并在其生命周期中根据通信的结果不断改变自身的状态。

消息的格式



[序号] [警戒条件] *[重复次数] [回送值表：＝] 操作名 （参数表）

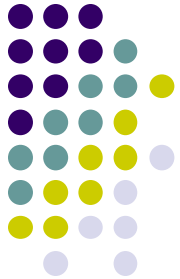


[序号]

表示消息在对象间交互的时间顺序号

1. 一般用正整数1、2、3…表示
2. 嵌套消息用1.1、1.2 、2.1、 2.2…表示
3. 序号在协作图中必不可少
4. 序号在时序图中可省略

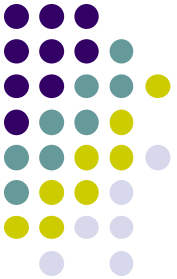
[序号] [警戒条件] *[重复次数] [回送值表：=] 操作名 （参数表）



[警戒条件]

选择项，为布尔表达式

1. 满足警戒条件的时候才能发送消息
2. 缺省时，表示消息无条件发送



*[重复次数]

选择项，表示消息重复发送的次数

1. 只有*号，无[重复次数]，表示消息多次发送，次数未定
2. 缺省，表示只发送一次

[回送值表：＝] 操作名 （参数表）

- 回送值表：以“，”区分的名字表列，分别表示完成指定操作后返回的系列值。
- [回送值表：＝] 可缺省
- 操作名必须是接收该消息的对象类角色的操作名
- （）内的参数表是以“，”区分的实参表，传递给接收消息的对象中的某个操作，实参的个数、次序、类型必须与该操作的实参一致。

消息的类型

发送消息可以触发的动作有：

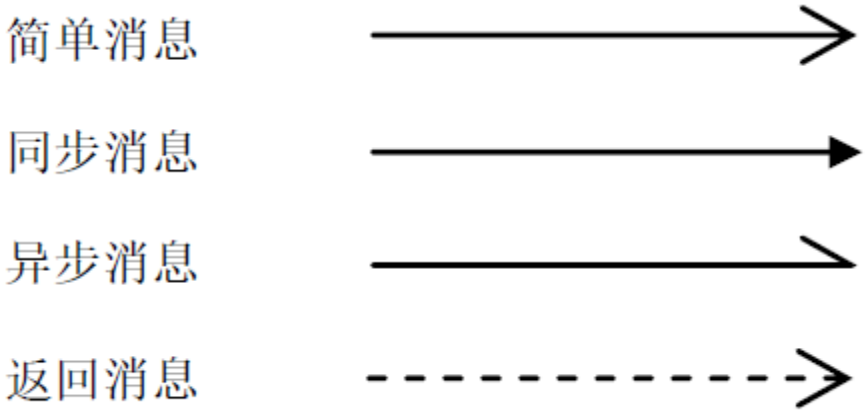
- 创建一个对象
- 释放一个对象
- 调用另一个对象的操作
- 调用本对象的操作
- 发送消息给另一个对象
- 返回值给调用者

消息分为4种控制流

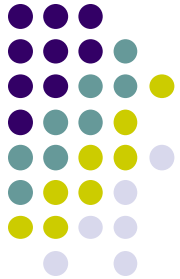
- 简单消息
- 同步消息
- 异步消息
- 返回消息

UML中消息的表达方式

- 简单消息(Simple Message) 表示简单的控制流。用于描述控制如何在对象间进行传递，而不考虑通信的细节。
- 同步消息(Synchronous Message) 表示嵌套的控制流。操作的调用是一种典型的同步消息。调用者发出消息后必须等待消息返回，只有当处理消息的操作执行完毕后，调用者才可继续执行自己的操作。
- 异步消息(Asynchronous Message) 表示异步控制流。当调用者发出消息后不用等待消息的返回即可继续执行自己的操作。异步消息主要用于描述实时系统中的并发行为。
- （返回消息：一般可省略）

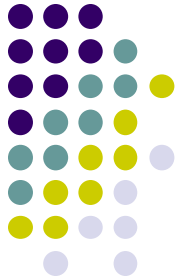


时序图（顺序图Sequence Diagram）



- 顺序图(Sequence Diagram)用来描述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。
- 顺序图存在两个轴:**水平轴表示不同的对象,垂直轴表示时间。**
- 顺序图中的对象用一个带有垂直虚线的矩形框表示,并标有对象名和类名。**垂直虚线是对象的生命线,用于表示在某段时间内对象是存在的。对象间的通信通过在对象的生命线间画消息来表示。**消息的箭头指明消息的类型。

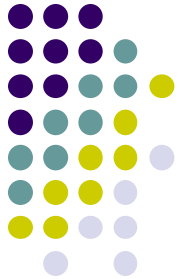
more



- 顺序图中的消息可以是信号(Signal)、操作调用。当收到消息时,接收对象立即开始执行活动,即对象被激活了。通过在对象生命线上显示一个细长矩形框来表示激活。

- 消息可以用消息名及参数来标识。消息也可带有顺序号,但较少使用。消息还可带有条件表达式,表示分支或决定是否发送消息。如果用于表示分支,则每个分支是相互排斥的,即在某一时刻仅可发送分支中的一个消息。

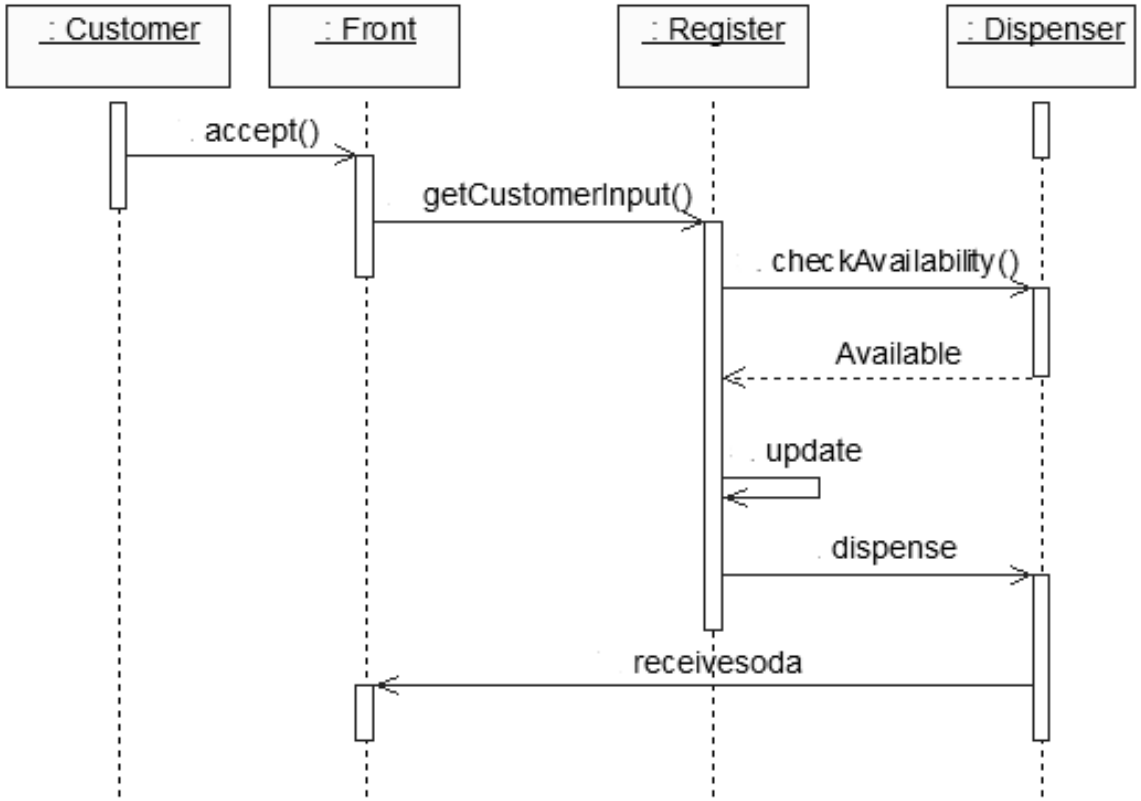
more



- 在顺序图的左边可以有说明信息,用于说明消息发送的时刻、描述动作的执行情况以及约束信息等。一个典型的例子就是用于说明一个消息是重复发送的。另外,可以定义两个消息间的时间限制。

- 一个对象可以通过发送消息来创建另一个对象,当一个对象被删除或自我删除时,该对象用"X"标识。

举例：自动饮料控制系统



该系统中，
顾客从前端塞入钱币，选择想要的饮料，

前端将钱送到钱币记录仪，

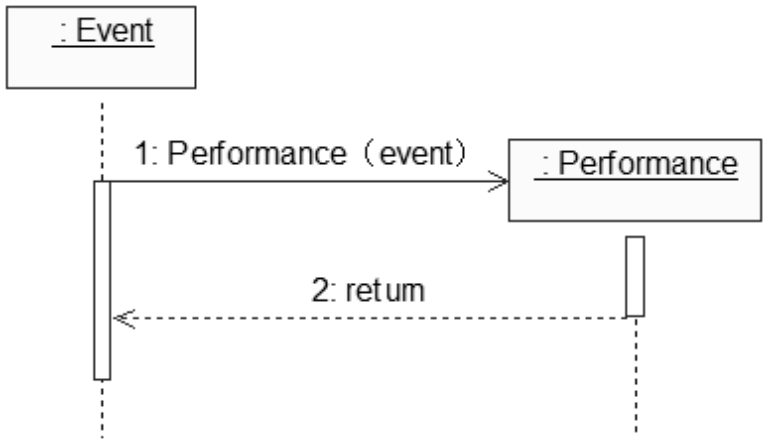
记录仪更新自己的存储信息，

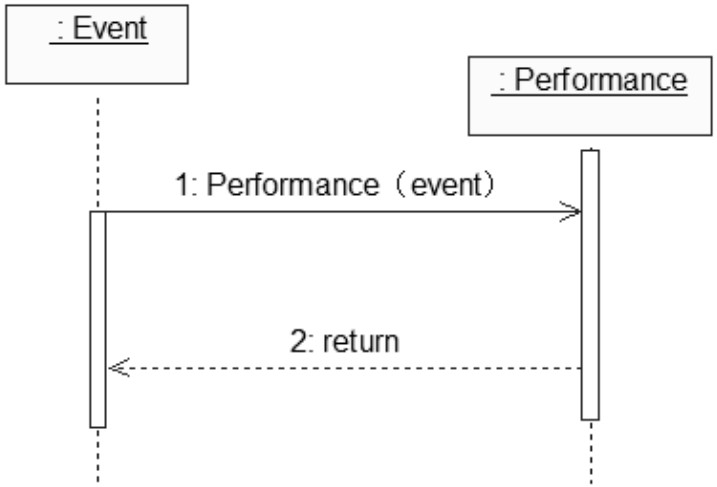
分配器检查系统有存货，

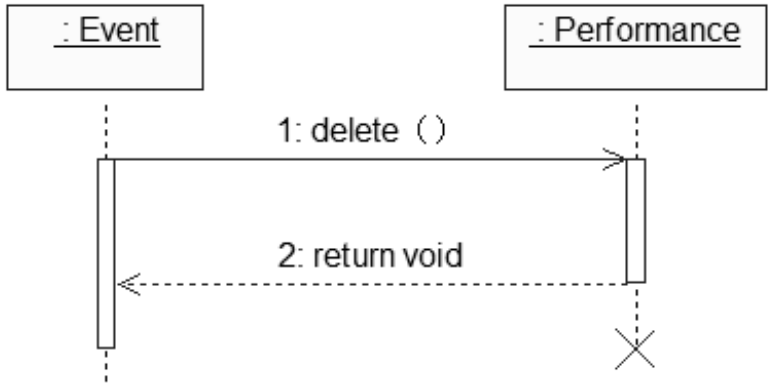
记录仪通知分配器分发饮料到前端。

完成一系列消息的传递。

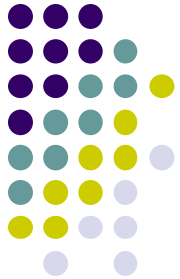
创建和撤销对象


: Event
1: Performance（event）
: Performance
2: retum


: Event
: Performance
1: Performance（event）
2: return

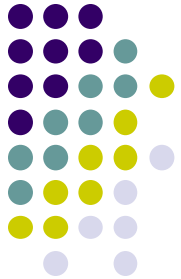

: Event
: Performance
1: delete（）
2: return void

同步消息和异步消息



1. 同步消息

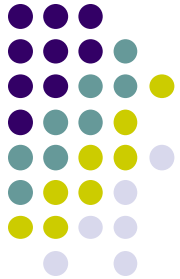
- 同步消息通过操作调用完成同步操作的嵌套控制流
- 同步消息的接收者必须是一个被动对象
- 同步消息必须用一条带实心三角箭头的箭线表示
- 一般同步消息有一个配对的返回消息，可以省略。



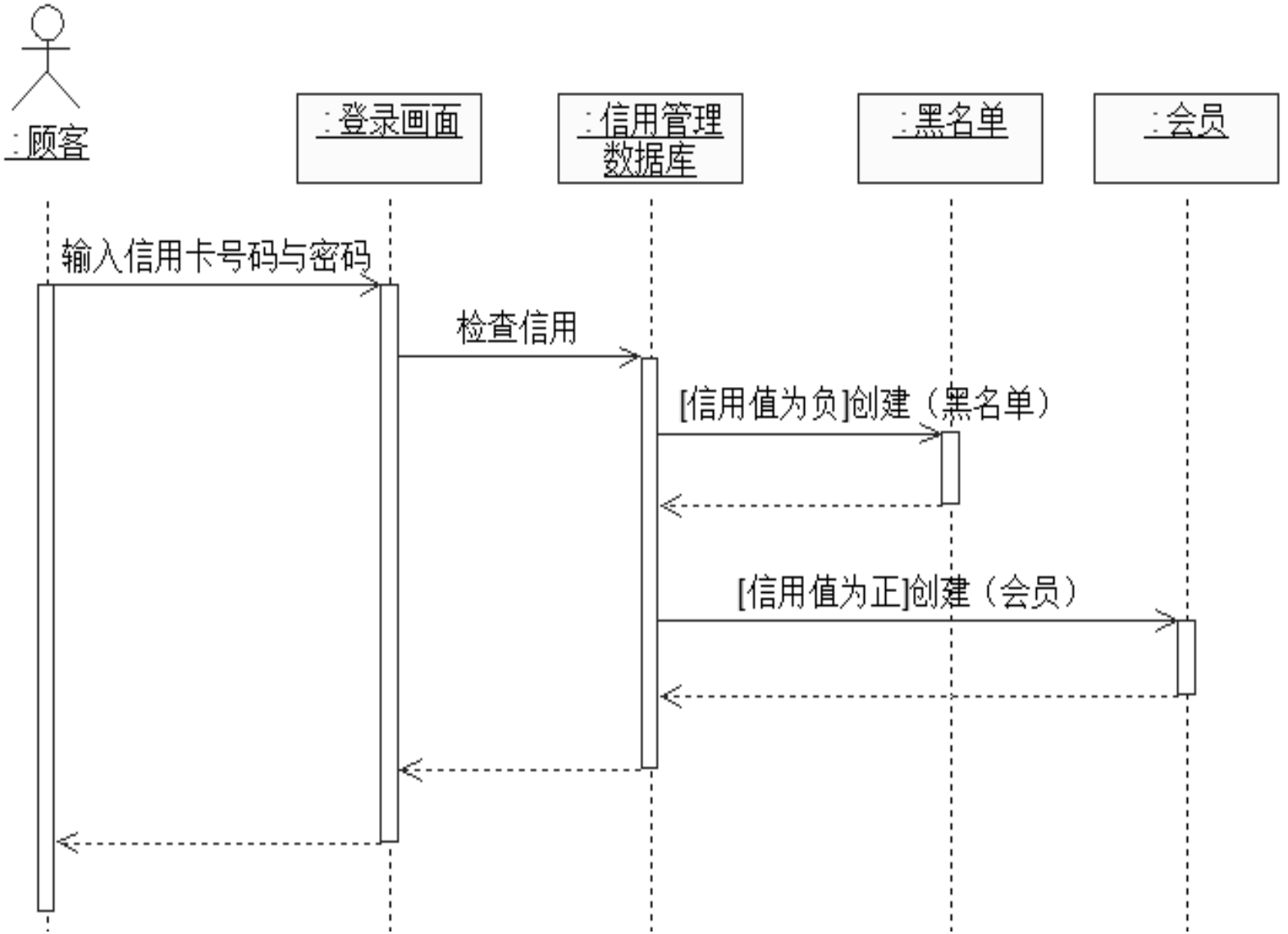
2.异步消息

- 异步消息是表达异步请求的一个操作的非嵌套的控制流。
- 异步消息由一条带半叉箭头或半实心箭头的箭线表示
- 异步消息的发送和接受者采用并发工作方式

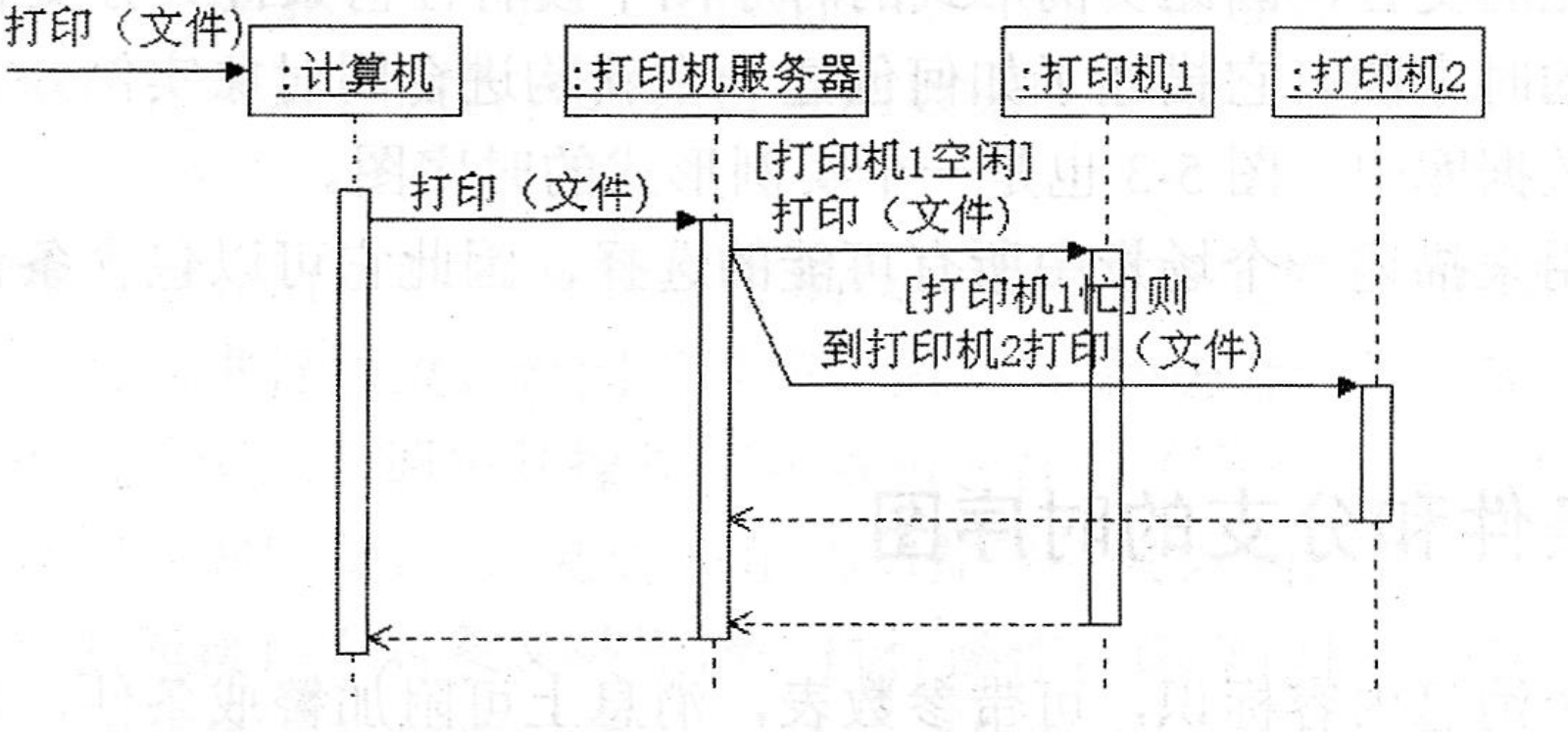
带条件和分支的时序图



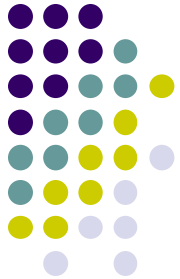
- 消息有一个内容标识，可带参数表，消息上可附带警戒条件，当条件为真时消息才被发送或接收。
- 条件可用于描述分支。
- 当几个消息箭头上的条件互斥时，表示某一时刻只有一个消息被发送，称为条件分支；如果条件不是互斥的，则这些消息会被并行发出。


:顾客
:登录画面
:信用管理
数据库
:黑名单
:会员
输入信用卡号码与密码
检查信用
[信用值为负]创建（黑名单）
[信用值为正]创建（会员）

Example: 带条件和分支同步执行的时序图



more



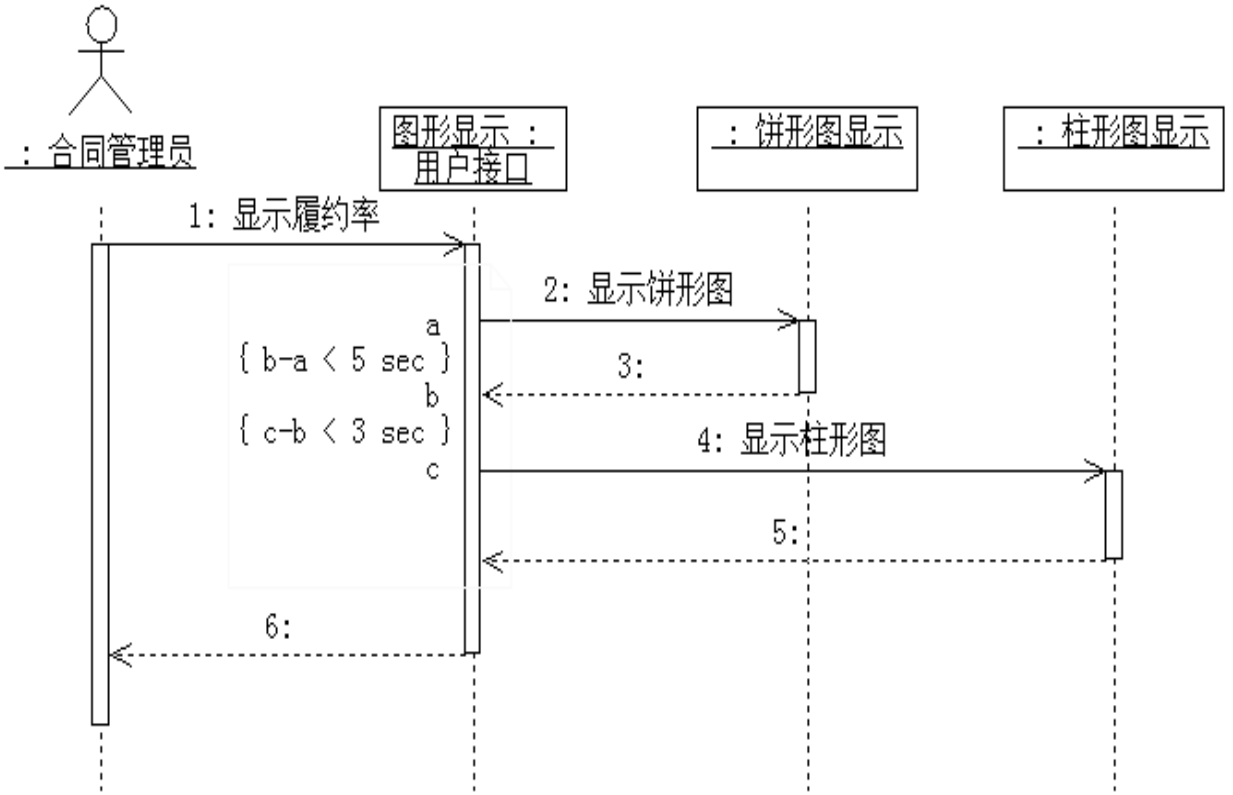
- 当一台计算机接收到请求打印文件的消息后，立即向“打印机服务器”对象发送“打印文件”请求。

- “打印机服务器”对象接收到消息后，同时发出两条带警戒条件的消息：一条发送给“：打印机 1 ”对象，警戒条件为：“如果「打印机 1 空闲]，则打印文件”；另一条发送给“：打印机 2 ”对象，其警戒条件为：“如果[打印机1忙]，则到打印机2打印（文件）”。

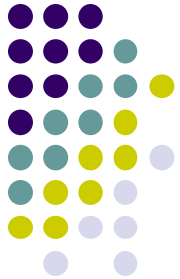
- 当两台打印机都处于空闲状态，而计算机只有两个文件需要打印时，这两条消息可以并发执行。

带约束标记的时序图

- 约束用分隔符{ }围起。
- 标记可以是任何类型的，如时间标记和事件约束。

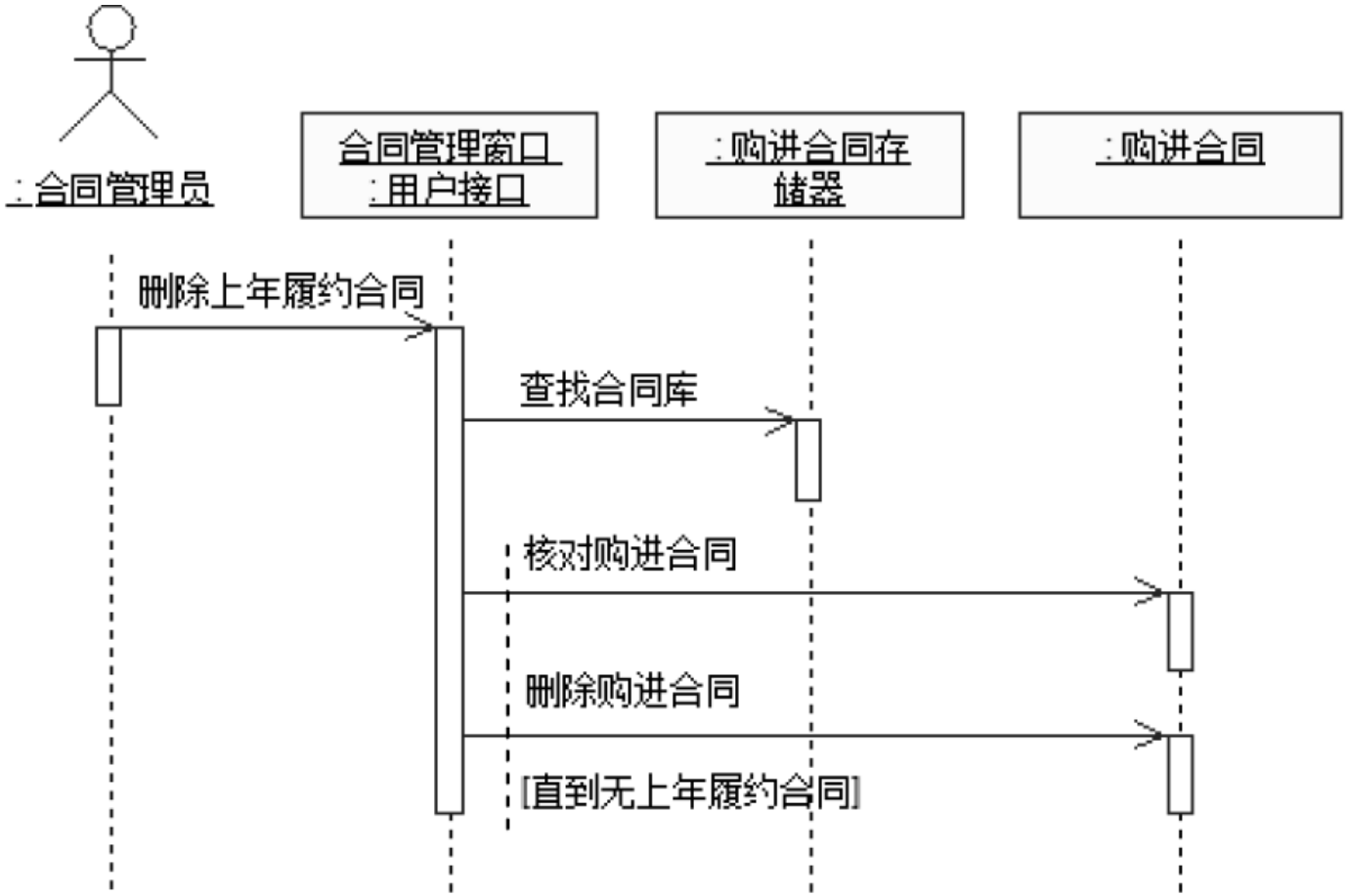


带循环标记的时序图

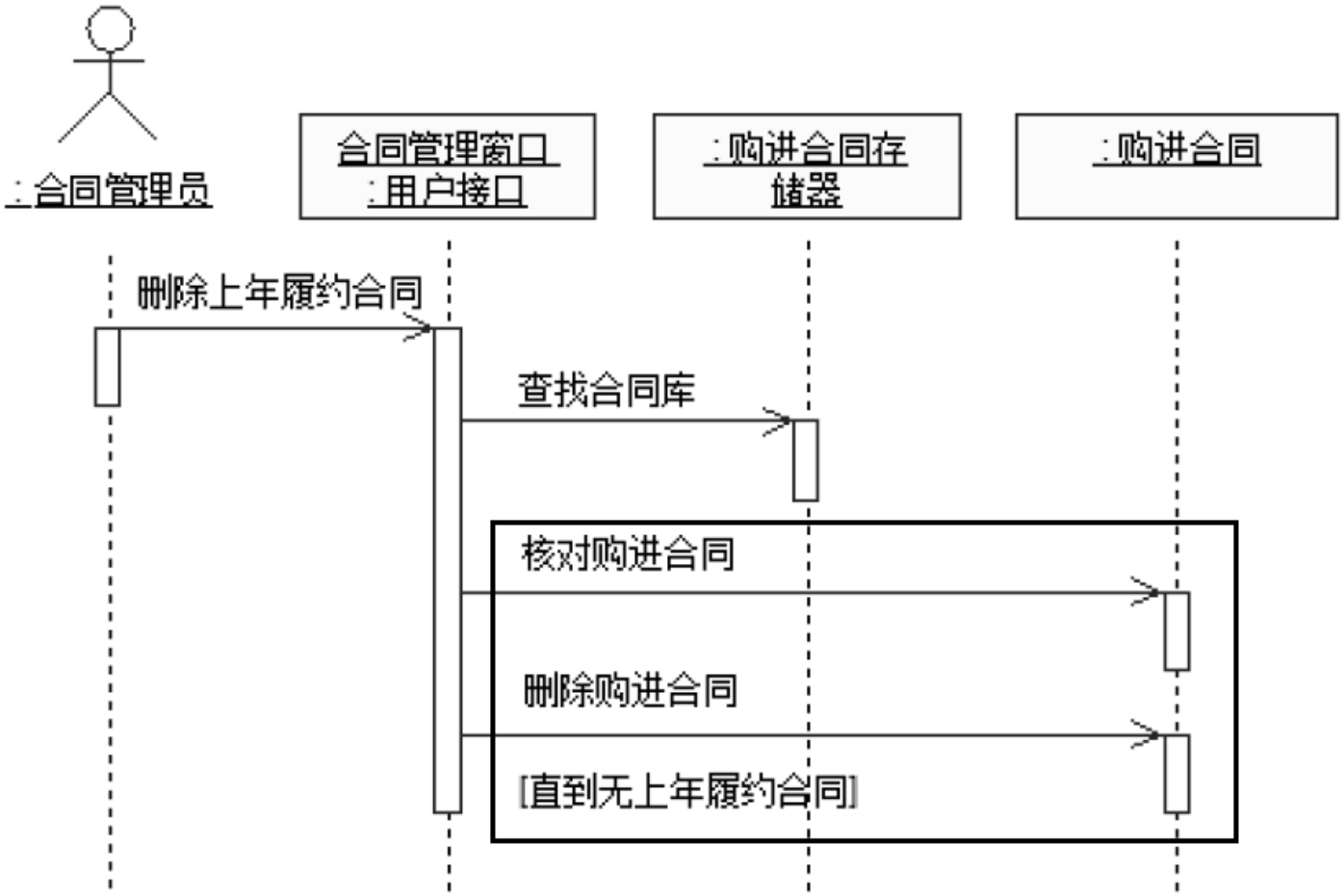


- 一个对象向另一个对象连续多次发送同一组消息，称为消息的循环
- 循环标记用一个矩形框与其包含的一组消息表示。
- 用[ ]围起的表示停止（或继续）循环的条件，标识在矩形框的底线内侧边上。
- 可以用一条在发送方竖立的虚线代替方框。

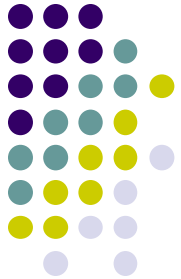
循环表示法1


:合同管理员
合同管理窗口
:用户接口
:购进合同存储器
:购进合同
删除上年履约合同
查找合同库
核对购进合同
删除购进合同
[直到无上年履约合同]

循环表示法2


:合同管理员
合同管理窗口
:用户接口
:购进合同存
储器
:购进合同
删除上年履约合同
查找合同库
核对购进合同
删除购进合同
[直到无上年履约合同]

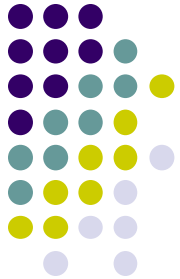
协作图（合作图）



- 协作图**(Collaboration Diagram)**用于描述相互合作的对象间的交互关系和链接关系。

- 虽然顺序图和合作图都用来描述对象间的交互关系，但侧重点不一样。顺序图着重体现交互的时间顺序，合作图则着重体现交互对象间的静态链接关系。

more



- 在协作图的对象框中，可以在{ }内填写文字用来表示该对象的创建或消亡。
  - 对象创建{new}，表示该对象在协作期被创建；
  - 对象消亡{destroyed}，表示该对象在协作期消亡；
  - 对象创建并消亡{transient}，表示该对象在创作期被创建并消亡

- 对象间的链接关系类似于类图中的联系(但无多重性标志)。通过在对象间的链接上标志带有消息串的消息(简单、异步或同步消息)来表达对象间的消息传递。

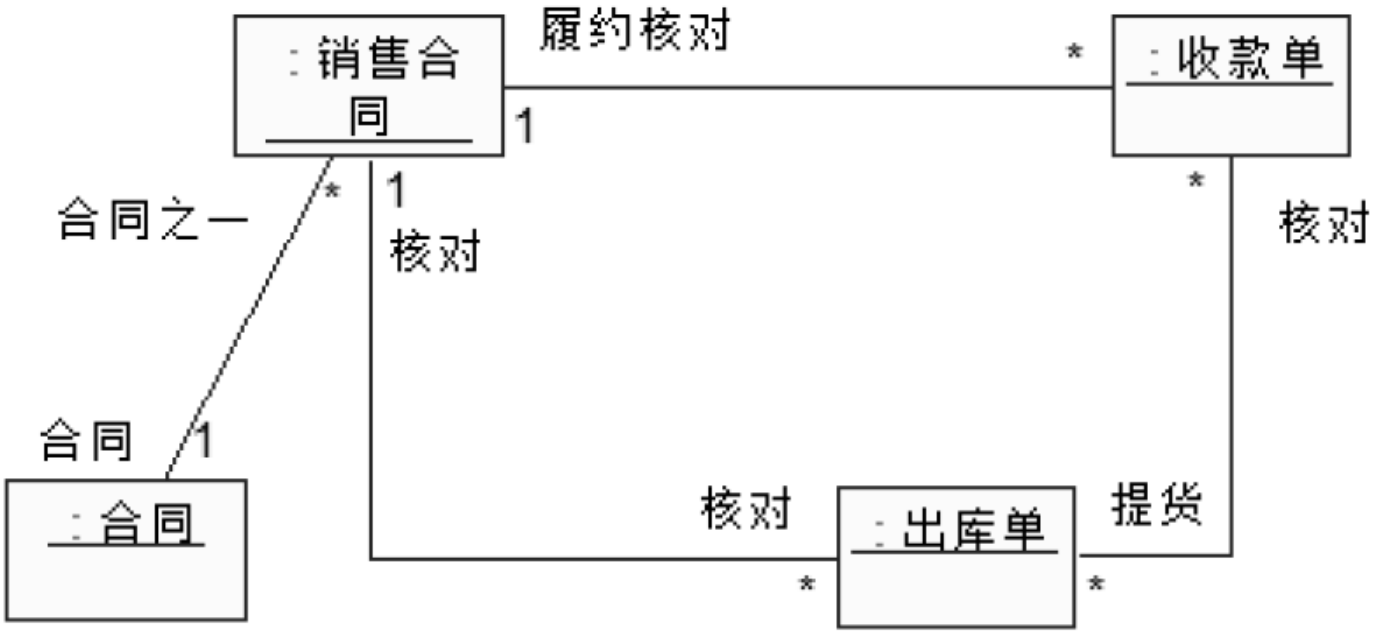
协作图的成分

- 对象
- 消息
- 链接

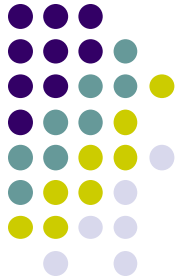
链接用于表示对象间的各种关系，包括组成关系的链接(Composition Link)、聚集关系的链接(Aggregation Link)、限定关系的链接(Qualified Link)以及导航链接(Navigation Link)。

各种链接关系与类图(对象图)中的定义相同，在链接的端点位置可以显示对象的角色名和模板信息。

Example：带有链接角色的销售子系统的协作图



explanation



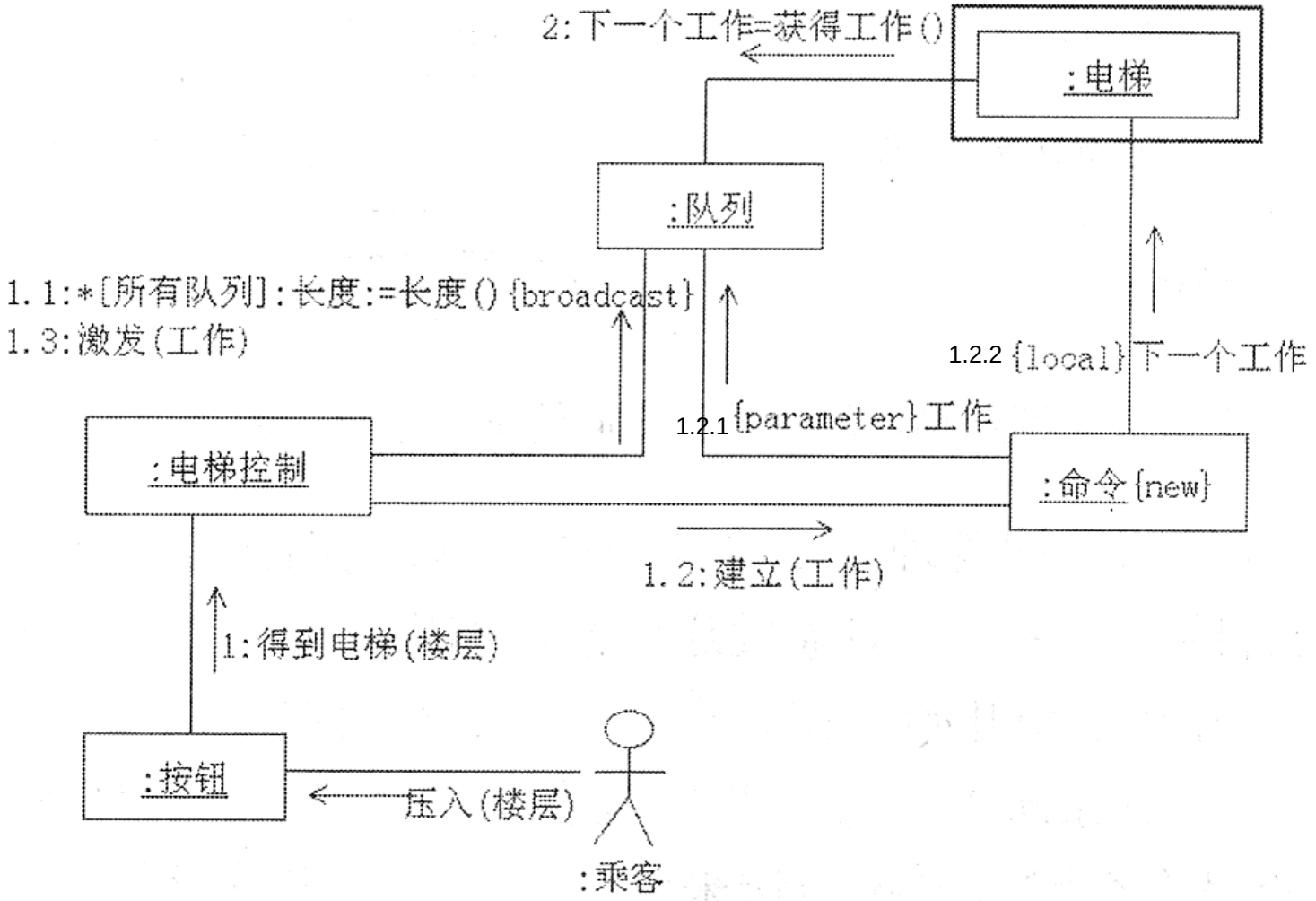
- 在图中，有 4 个对象：合同、销售合同、收款单和出库单。
- 这些对象之间用实线连接，表示它们之间有关联，关联角色和多重性标志在关联的两端标出。
- “：合同”对象和“：销售合同”对象之间的关联角色表明销售合同是合同之一；它们之间的多重性是一对多关系。
- “：销售合同”对象与“：收款单”对象之间的关联角色表明销售合同与收款单之间进行“核对”；它们之间的多重性是一对多关系。
- “：销售合同”对象与“：出库单”对象之间的关联角色表明销售合同与出库单之间进行“履约核对”；它们之间的多重性也是一对多关系。
- “：收款单”对象与“：出库单”对象之间的关联角色表明收款单与出库单之间也进行“核对”，核对正确则“提货”；它们之间的多重性是多对多关系。

协作图的成分

- 对象
- 消息（流）
- 链接

在协作图的链接线上，可以用带有消息串的消息来描述对象间的交互。消息的箭头指明消息的流动方向。消息串说明要发送的消息、消息的参数、消息的返回值以及消息的序列号等信息。

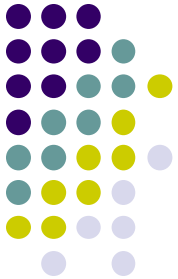
Example:一个电梯按钮的协作图



在该协作图中各个对象之间发送的都是同步消息。协作从一个乘客要求乘电梯时开始，电梯按钮的协作图工作过程如下：

（1）“:乘客”：一个电梯“:乘客”压入按钮，向“:按钮”对象发送消息，希望得到电梯。

（2）“:按钮”对象：接收到“:乘客”发送的消息后，向“:电梯控制”对象发送序号为 1 的消息，调用“:电梯控制”对象的操作“:得到电梯（楼层）”，即要求电梯来到乘客所在的楼层。



（3）“:电梯控制”对象：接收到消息后，为完成该操作，向不同对象发出一系列嵌套消息：

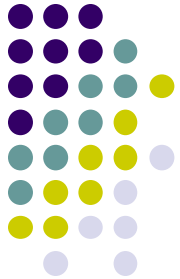
1）首先向“:队列”对象发送序号为 1 的具有广播性的嵌套消息：1.1：*［所有队列］＝长度()，即对存储在队列中的所有工作队列的长度进行循环检查，找出其中最短的，即离乘客所在楼层最近且同方向的工作；

2）再发送序号为 2 的嵌套消息：1.2:建立（工作）；建立一个工作命令对象，在该对象框的对象名后面标明｛new｝，说明该对象被创建，并将该带参数的消息“{ Parameter｝工作”命令置于最短的队列中；同时，“命令”对象向“电梯”对象发送一个局部消息：进行下一步工作；

3）最后向“:队列”对象发送序号为 3 的嵌套消息： 1.3:激发（工作）。

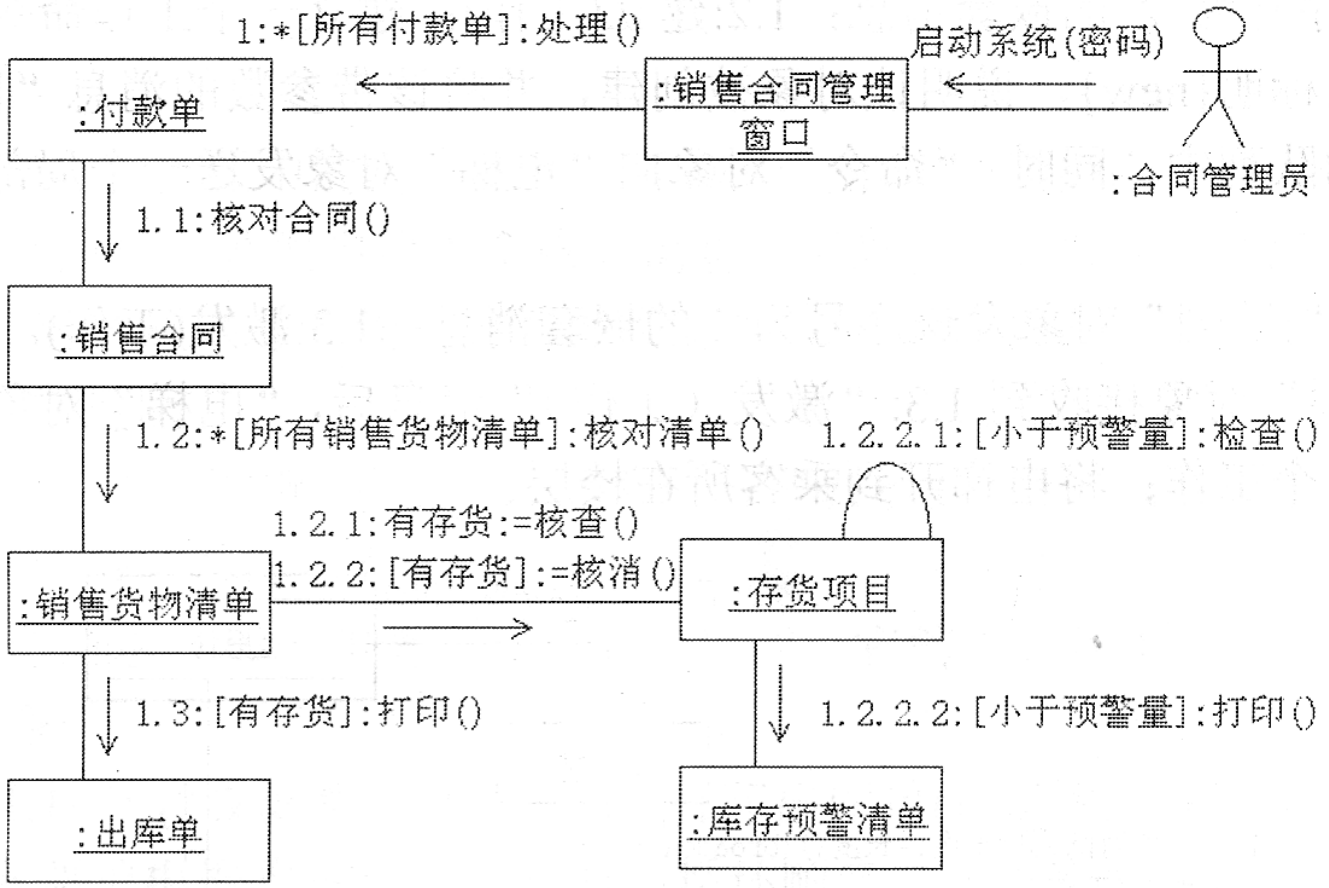
（4）“:队列”对象接收到 1.3: “激发（工作）”消息后，“电梯”对象同时运转，并从队列中选出下一个工作；将电梯开到乘客所在楼层。

协作图中消息的层次关系

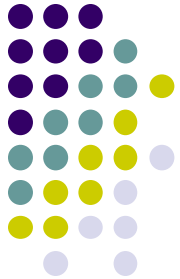


- 协作图中，对象之间传递的消息必须表明序号，用以说明消息传送的先后顺序。
  - 嵌套消息表示法
  - 消息顺序表示法：缺乏层次感

Example：嵌套表示的处理付款单协作图



explanation



- 该图只是描述销售合同管理系统中处理付款单这项工作的协作图。

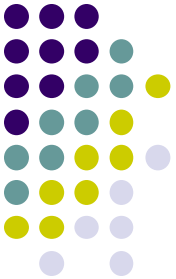
- 在协作图中，合同管理员向“：销售合同管理窗口”对象发送消息，调用其操作“启动系统（密码）”，检验密码正确后，销售合同管理系统开始工作。合同管理员在窗口下选择“处理付款单”功能，“：销售合同管理窗口”对象向“：付款单”对象发送序号为1的循环处理消息“1：*所有付款单]：处理（）”，检查是否有财务系统传送来的付款单，如果有付款单，依次循环对付款单逐一处理，直到所有付款单处理完毕。

Continued…

- 在循环处理过程中所有对象发出的消息都是嵌套消息，即“：付款单”对象及后面所有对象发出的消息都是“消息1”的嵌套消息，通过各条消息的序号排列，可以看出它们具有明显的层次隶属关系：

  1: * [所有付款单]：处理（）
  1.1：核对合同（）
  1.2：*[所有销售货物清单］：核对清单（）
  1.2.1：有存货：＝核查（）
  1.2.1.1：[小于预警量］：检查（）
  1.2.2.2：[小于预警量]：打印（）
  1.3：[有存货]：打印（）

协作图中自调用与回调

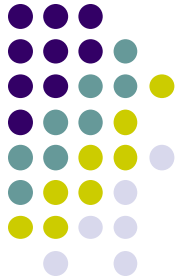


- 一个对象可以调用自己的操作，称为消息的自我调用或递归调用。

- 在协作图中，消息的自我调用的表示方法：

  连接线从消息的发送者对象返回到对象本身，也可以在连接线旁标记构造型<<self>>

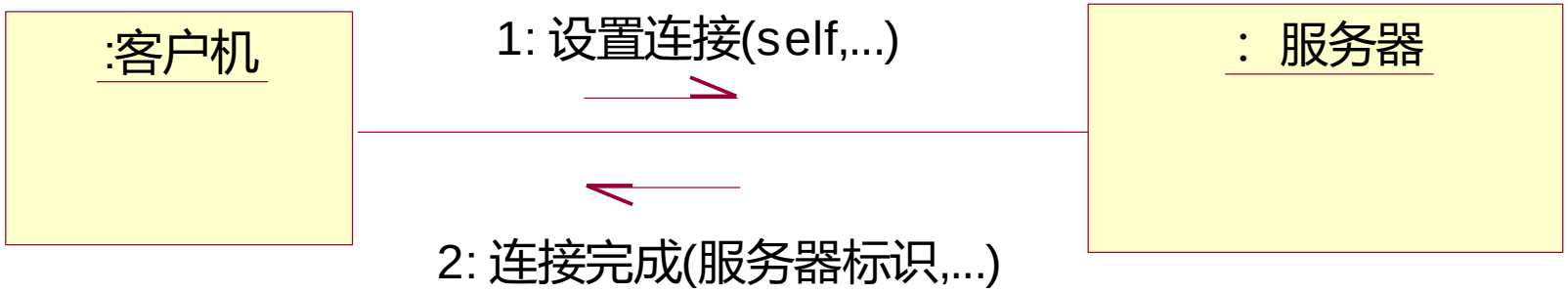
回调



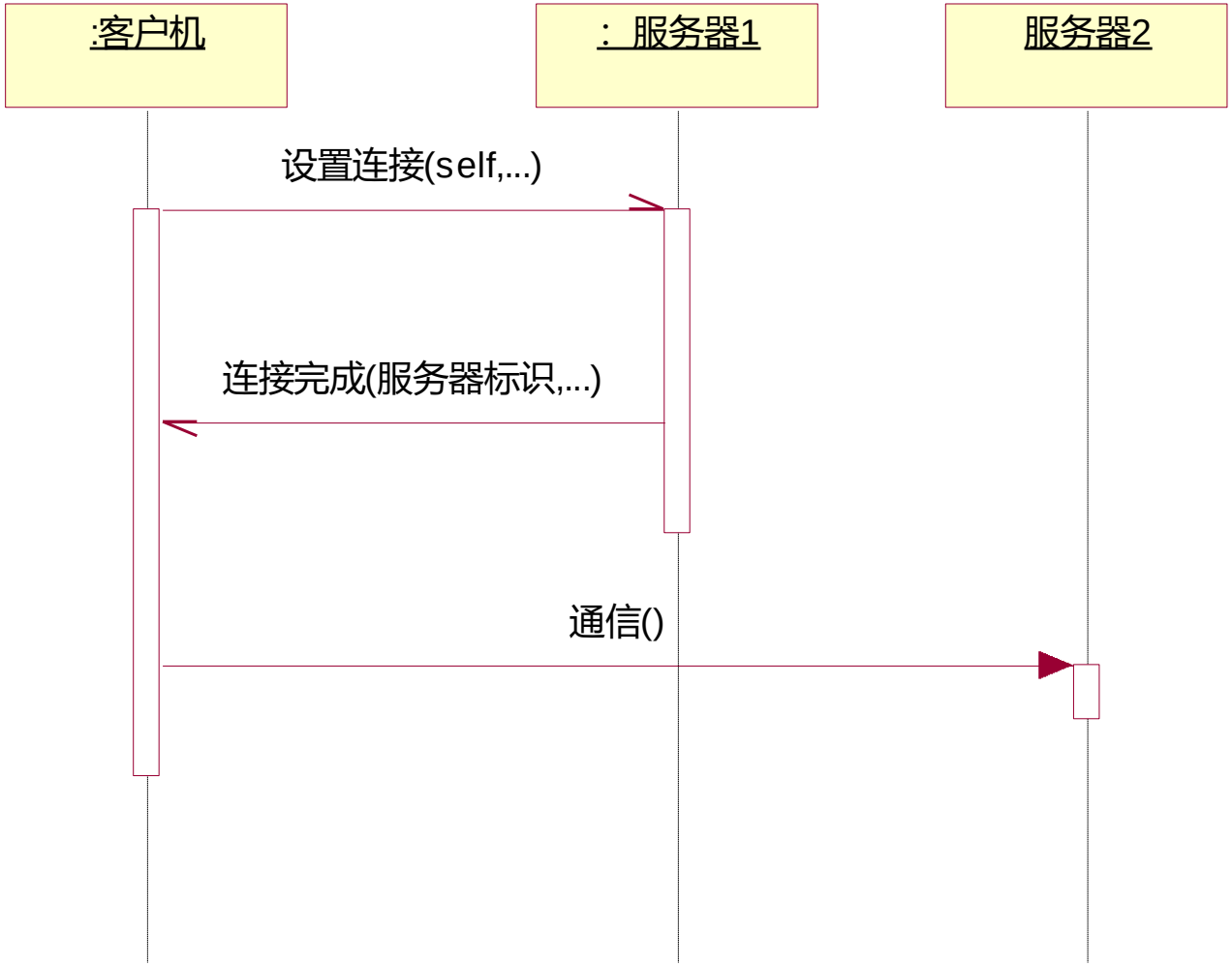
- 对象之间也有消息的回调，即消息的接收对象也可以向消息的发送对象发送消息。

- 消息回调的标记是在对象连接线的上下方各有一条方向相反的消息，如果连接线上方的消息为正常消息传递方向，则其下方的反方向传送消息称之为回调。回调都是异步消息，可以并发运行。

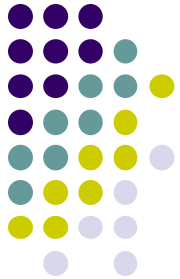
Example： 回调的协作图



回调的时序图



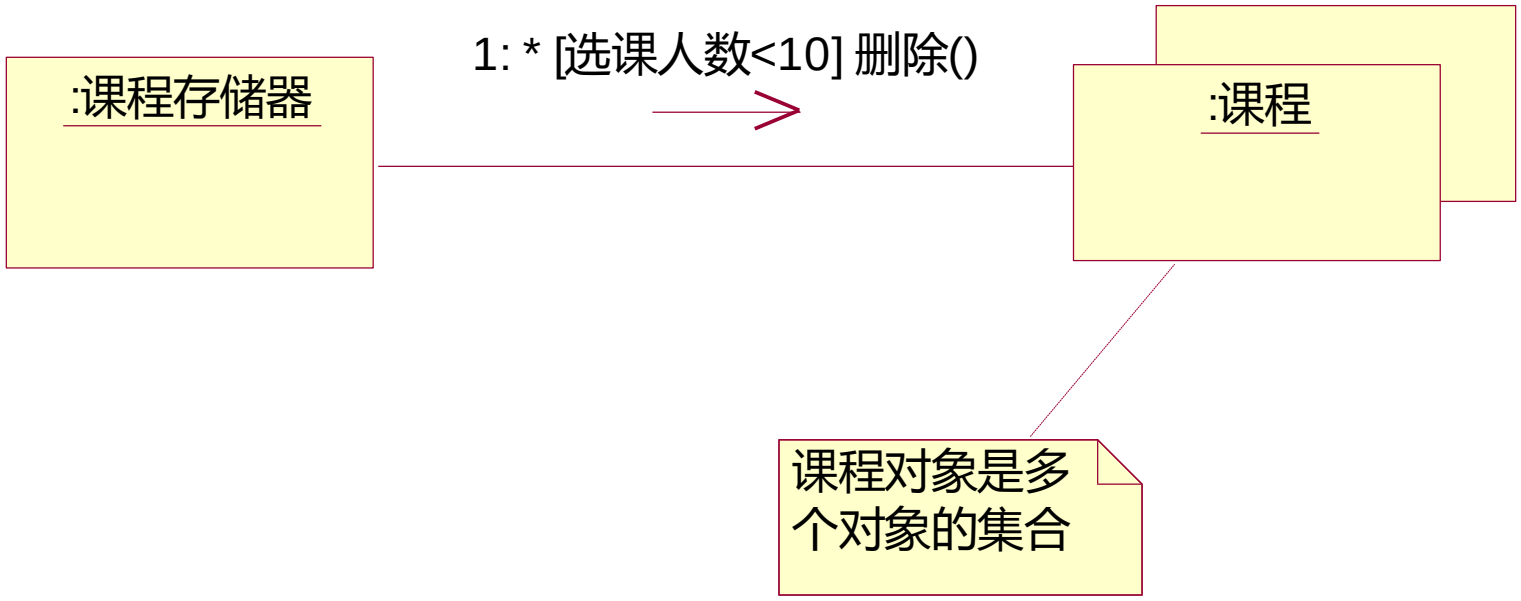
协作图中的重复消息

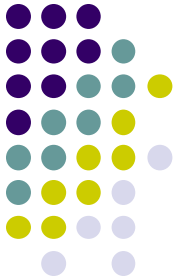


- 一个对象存储器（容器）中对象之间的动态交互关系可以用协作图来描述。

- 在协作图中，如果对象存储器要在多个对象中查找满足某个条件的对象，就要向其管理的对象发送一个可重复循环执行的消息。这个时候，接收方就是多个对象，可以用重叠的对象框表示。

Example： 一个重复消息发送给多个对象的协作图

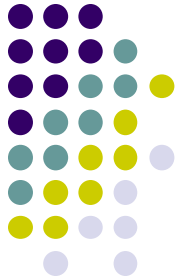




动态建模的几个基本概念

- 进程：进程是一个动作流；能够与其它进程并发执行。
- 线程：线程是进程内部的一个动作流；能够与其它线程并发执行。

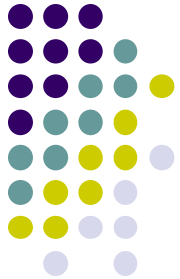
Continued…



- **主动对象：**
  - 一个拥有进程或线程的对象；
  - 能初始化控制活动；
  - 主动对象一旦被创建，无须由其它对象发来消息触发就能自动执行动作；
  - 主动对象提供主动服务
  - 一个系统可以有多个主动对象，各自独立并发运行

- **被动对象：**
  - 必须由其它对象发来的消息进行触发，才执行动作的对象
  - 系统中绝大多数是被动对象

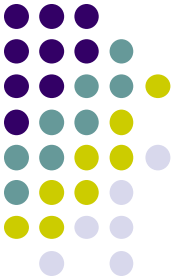
Continued…



- 主动对象类
  - 主动对象类是主动对象的抽象
  - 主动对象是主动对象类的实例
  - 主动对象用约束{active}来说明或用粗线条的对象框表示

- UML中，可以把一个独立的单线程模型化为一个主动对象，而对于多线程模型可以用协作图来表示

时序图中对象排列的原则

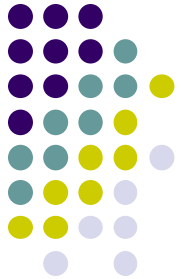


- 对象横向排列在时序图的顶部
- 最重要的对象在最左边（例如对全局进行初始的对象）
- 交互密切的对象尽可能相邻
- 交互中创建的对象，应放置在其创建的时间点上
- 每个对象有一个下垂的生命线

消息传递

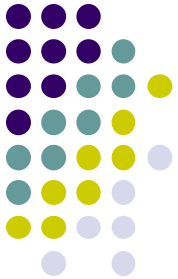
- 消息传递的方向：按时间顺序从上到下在对象的生命线之间传递
- 在消息箭头上标出消息标签的内容、约束或构造型
- 发送和接收对象的生命线必须处在激活期
- 交互中对象的创建和销毁必须绘出构造型和标记
- 区别同步和异步消息的图标标识符
- 标出消息的循环结构和出口条件
- 时序图从初始化开始，以其返回消息终止

协作图中对象排列的原则



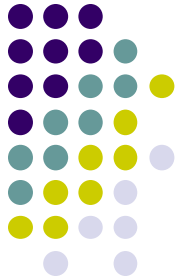
- 最重要的对象应在图的中央
- 与其有直接交互的对象放置在临近
- 对象初始化
- 选择初始对象

链接与消息传递



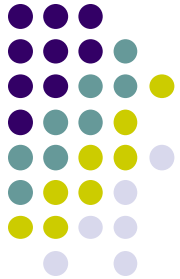
- 标明对象之间的链接
- 在链接上标上消息的序号
- 在消息箭线上标出消息标签的内容、约束或构造型
- 区别同步消息和异步消息的图标表示符
- 协作图从初始对象开始，到其终止对象结束

时序图与协作图



- 时序图和协作图常常被合称为交互图（**interaction diagram**），
- 它们描述的主要元素都是两个，即消息和对象角色。
- 实际上，这两种图的语法语义极为相似，在**Rational Rose**中甚至提供了在两种图之间进行切换的功能。

相同点



- 规定责任。两种图都直观地规定了发送对象和接收对象的责任。将对象确定为接收对象，意味着为此对象添加一个接口。而消息描述成为接收对象的操作特征标记，由发送对象触发该操作。

- 支持消息。两种图都支持所有的消息类型。

- 衡量工具。两种图还是衡量耦合性的工具。如果查看对象的交互图，就可以看见两个对象之间消息的数量以及类型，从而简化或减少消息的交互，以提高系统的设计性能。

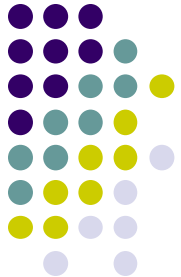
区别

（1）协作图的重点是将对象的交互映射到它们之间的链上，即协作图先以对象图的方式绘制各个参与对象，然后将消息和链平行放置。但是时序图却不把链表示出来。在时序图的对象之间，尽管没有相应的链存在，但也可以随意绘制消息。

（2）时序图可以描述对象的创建和撤销的情况。新创建的对象可以被放在对象生命线上对应的时间点，而在生命线结束的地方放置一个大写的X以表示该对象在系统中不能再继续使用。而在协作图中，对象要么存在要么不存在，除了通过消息描述或约束，没有其他的方法可以表示对象的创建或结束。但是由于协作图所表现的结构被置于静止的对象图中，所以很难判断约束什么时候有效。

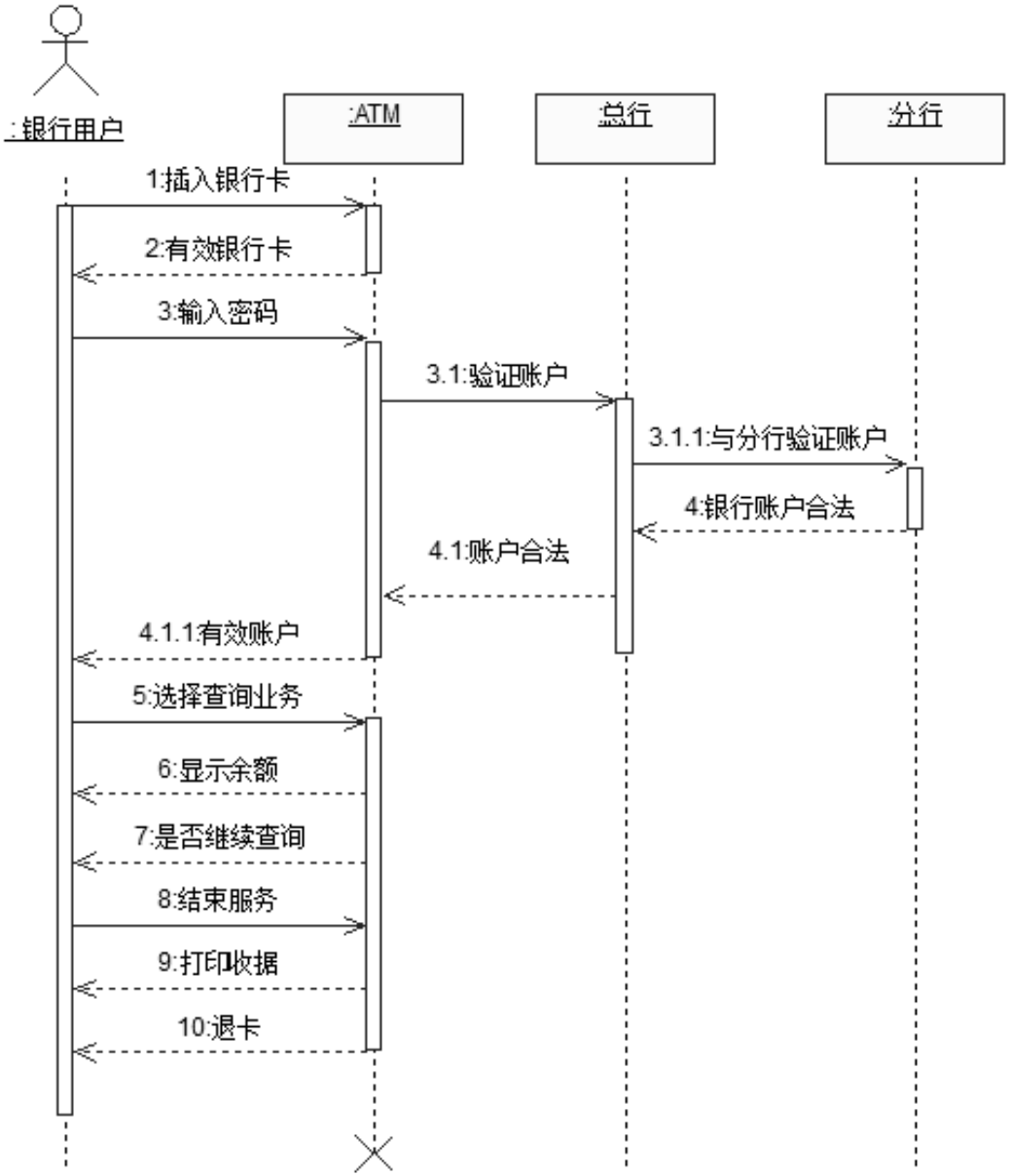
（3）时序图还可以表现对象的激活和去激活情况，但对于协作图来说，由于没有对时间的描述，所以除了通过对消息进行解释，它无法清晰地表示对象的激活和去激活情况。

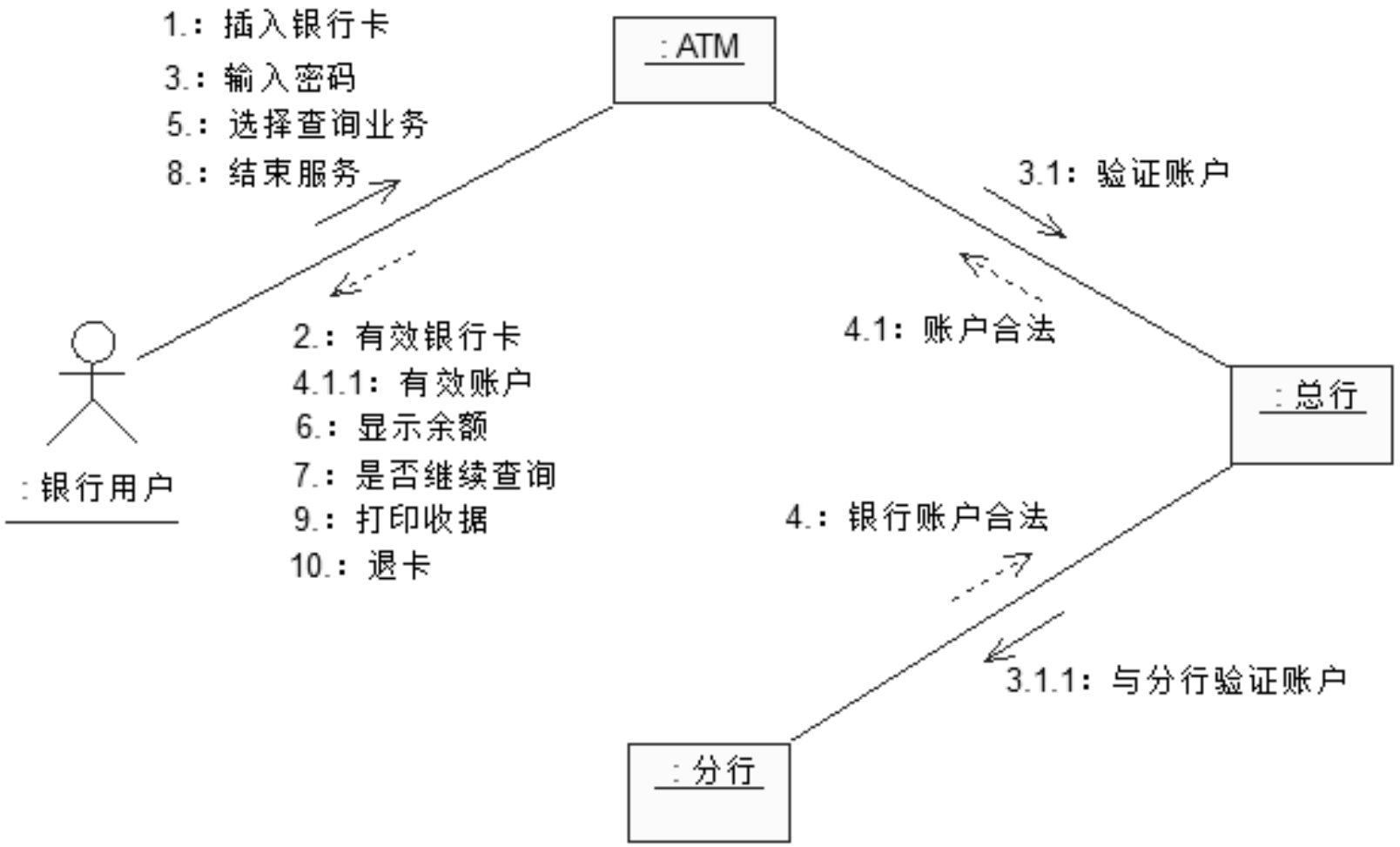
总结



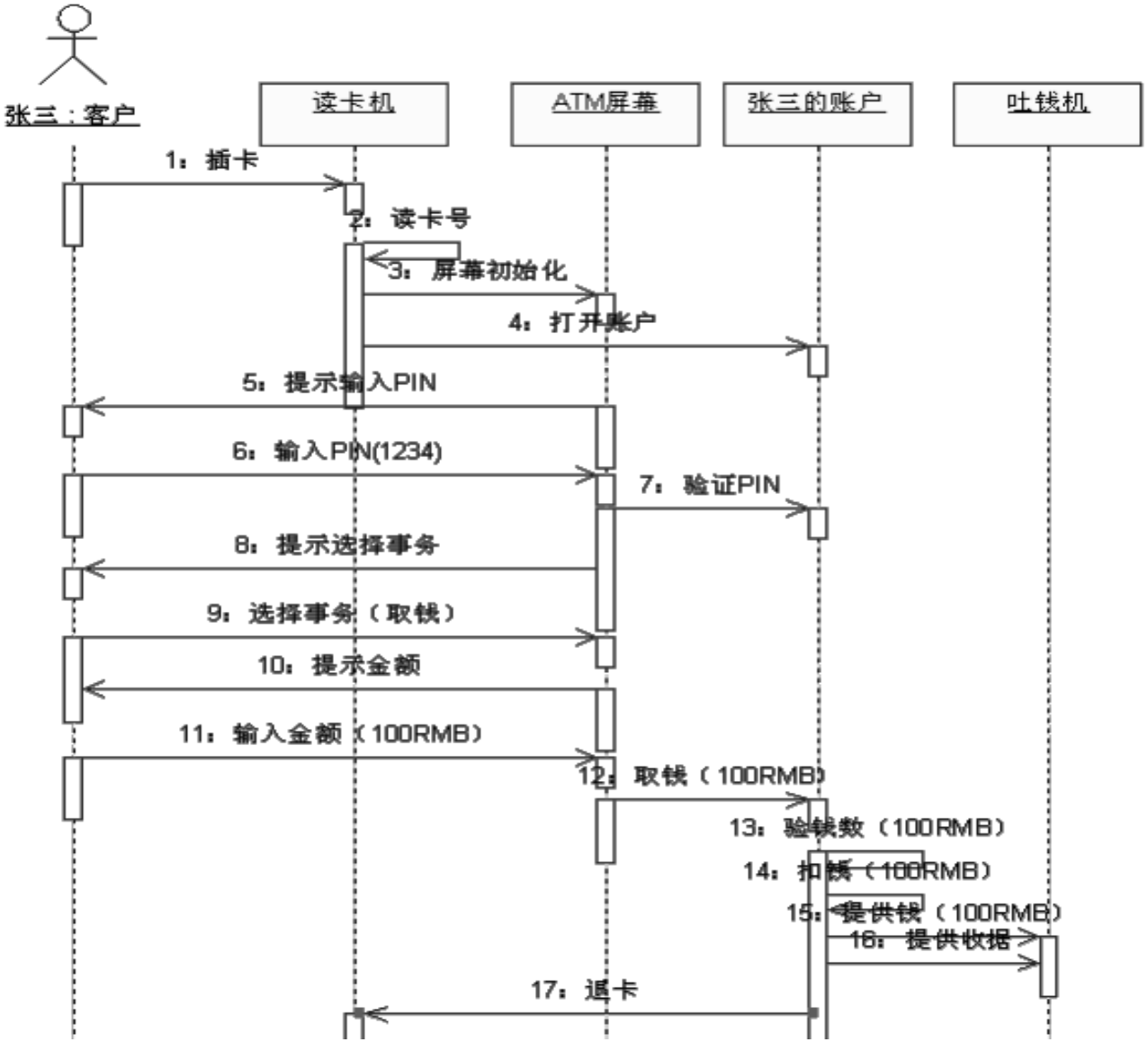
- 时序图与协作图都表示对象之间的交互作用，只是它们的侧重点有所不同。
- 时序图描述了交互过程中的时间顺序，但没有明确地表达对象之间的关系；
- 协作图描述了对象之间的关系，但时间顺序必须从顺序号获得。
- 两种图的语义是等价的，可以从一种形式的图转换成另一种形式的图，而不丢失任何信息。

时序图与协作图的转换：ATM系统查询余额

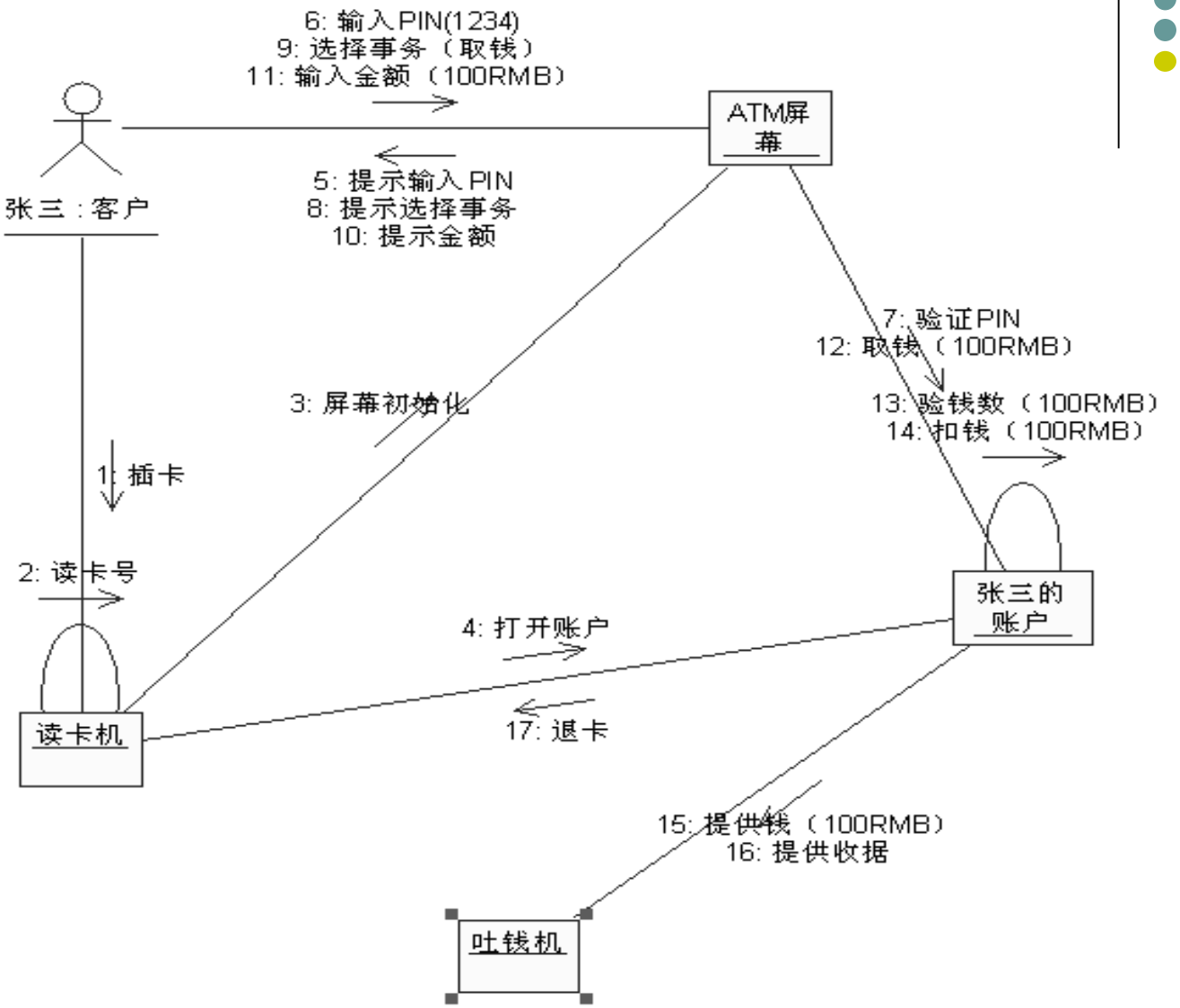



1.：插入银行卡
3.：输入密码
5.：选择查询业务
8.：结束服务
:ATM
3.1：验证账户
4.1：账户合法
2.：有效银行卡
4.1.1：有效账户
6.：显示余额
7.：是否继续查询
9.：打印收据
10.：退卡
:银行用户
:总行
4.：银行账户合法
3.1.1：与分行验证账户
:分行

ATM取款（100元）的时序图

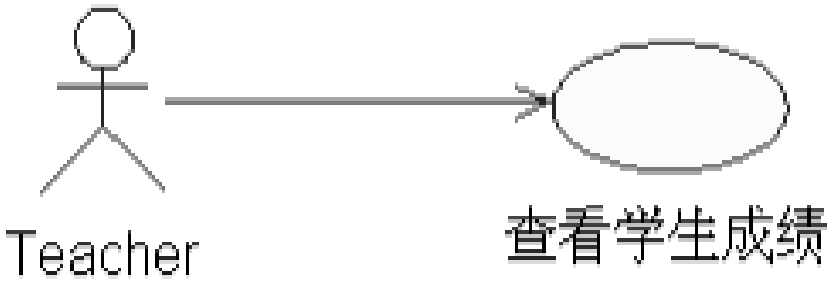


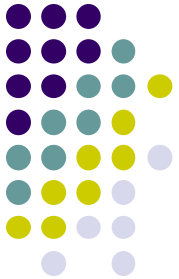
ATM取款（100元）的协作图



More example

- 以“教师查看学生成绩”为例，介绍如何去创建系统的时序图



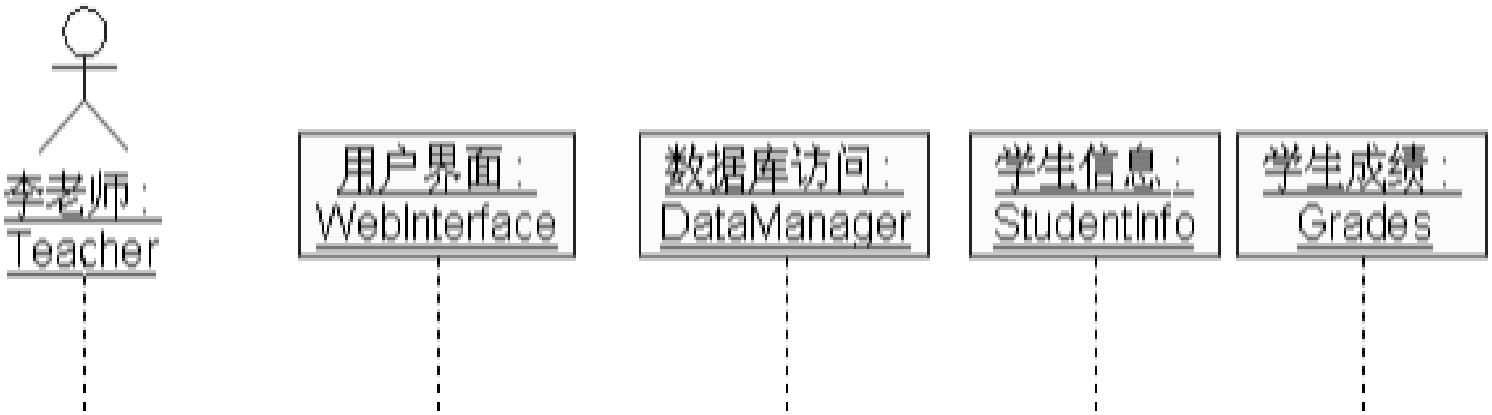


1.确定工作流程

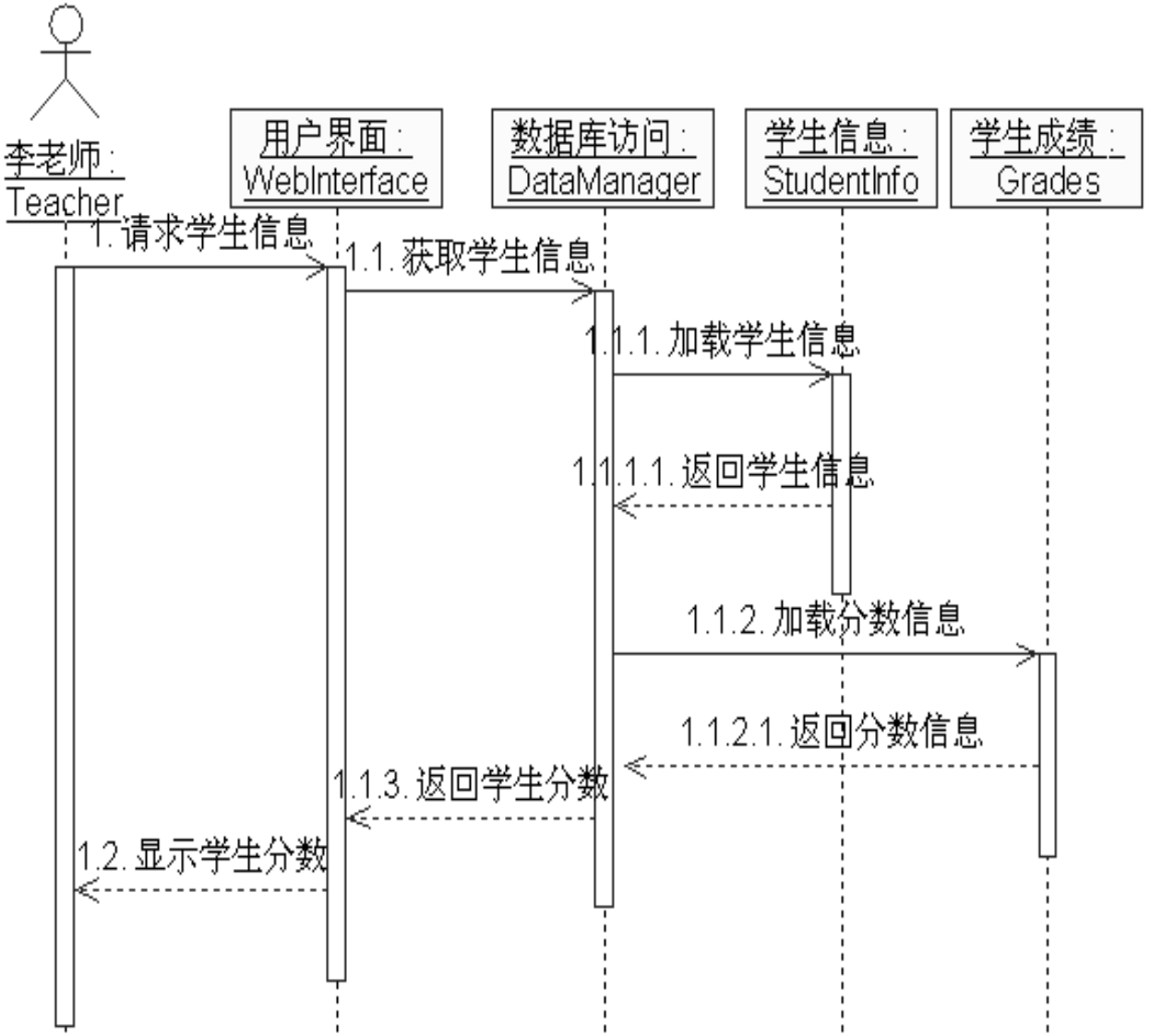
可以通过更加具体的描述来确定工作流程，基本的工作流程如下：

① 老师希望通过系统查询某名学生的学科成绩。
② 老师通过用户界面录入学生的学号。
③ 用户界面根据学生的学号向数据库访问层请求学生信息。
④ 数据库访问层根据学生的学号加载学生信息。
⑤ 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。
⑥ 数据库访问层将学生信息和分数信息提供给用户界面。
⑦ 用户界面将学生信息和分数信息显示出来。

- 建模时序图的下一步是从左到右布置在该工作流程中所有的参与者和对象

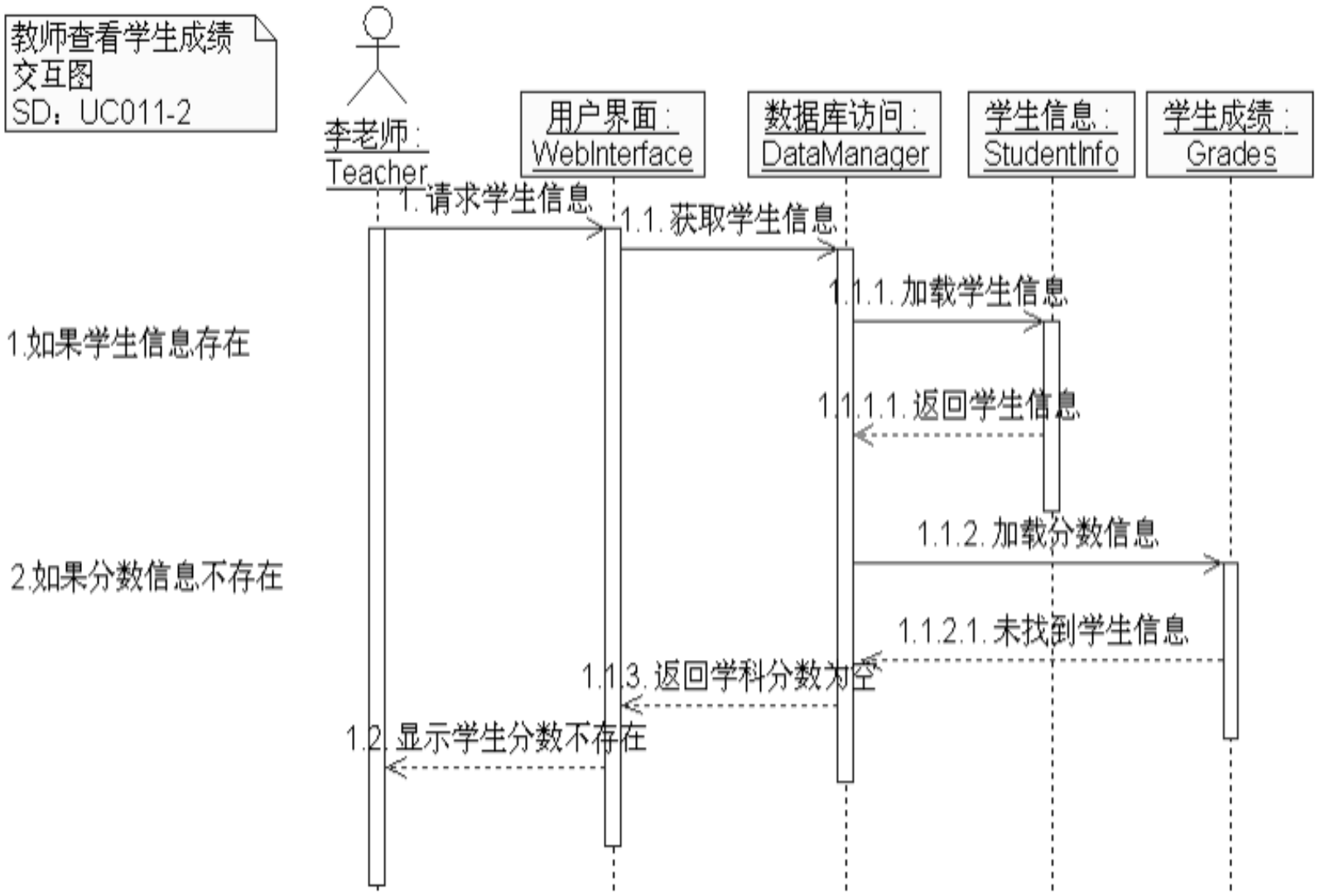


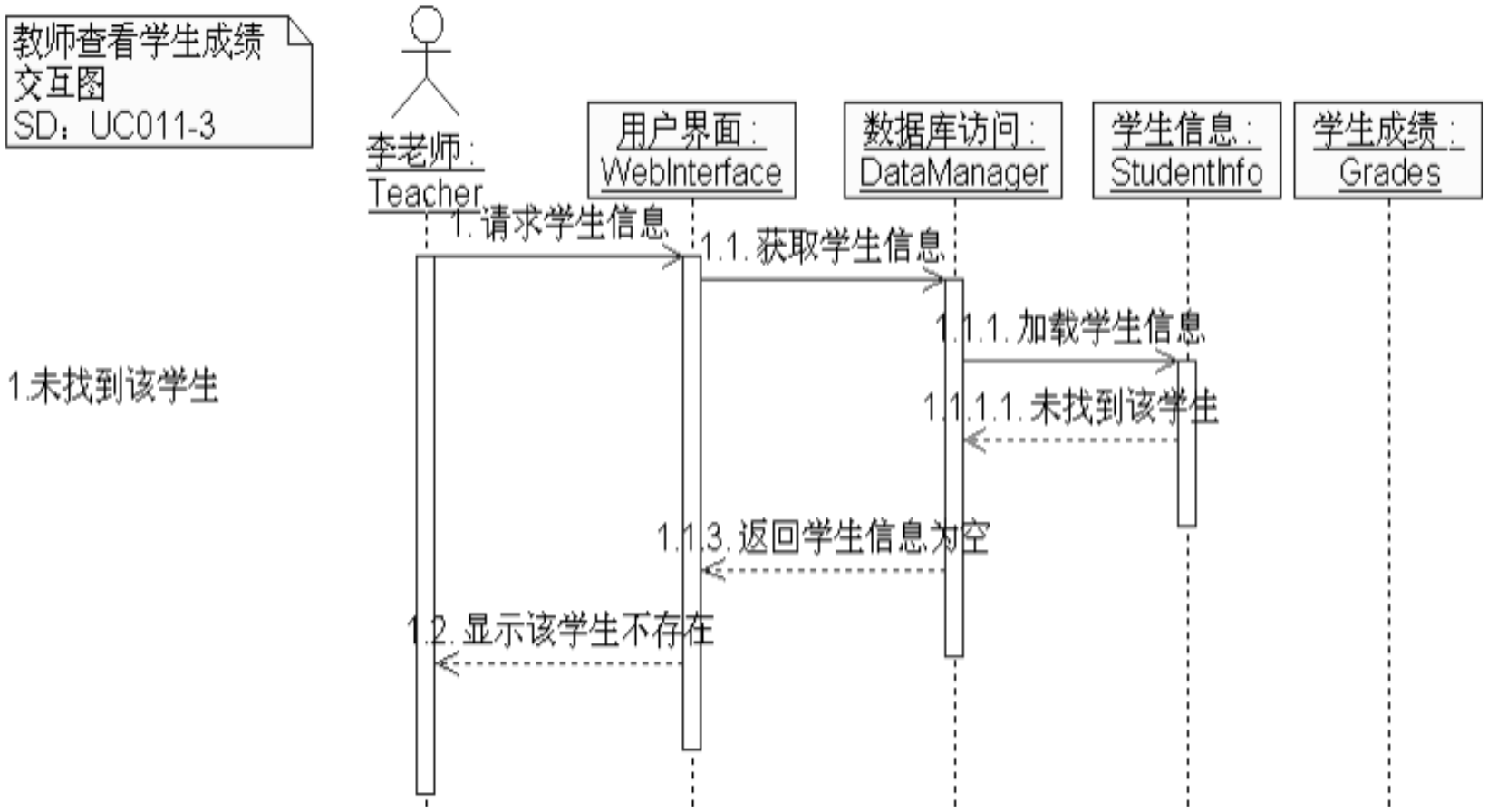
教师查看学生成绩
交互图
SD：UC011-1

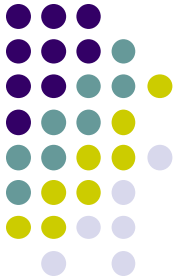


1.如果学生信息存在

2.如果分数信息存在


教师查看学生成绩
交互图
SD：UC011-2
李老师：
Teacher
用户界面：
WebInterface
数据库访问：
DataManager
学生信息：
StudentInfo
学生成绩：
Grades
1.请求学生信息
1.1.获取学生信息
1.1.1.加载学生信息
1.1.1.1.返回学生信息
1.1.2.加载分数信息
1.1.2.1.未找到学生信息
1.1.3.返回学科分数为空
1.2.显示学生分数不存在
1.如果学生信息存在
2.如果分数信息不存在


教师查看学生成绩
交互图
SD：UC011-3
李老师：
Teacher
用户界面：
WebInterface
数据库访问：
DataManager
学生信息：
StudentInfo
学生成绩：
Grades
1.请求学生信息
1.1.获取学生信息
1.1.1.加载学生信息
1.1.1.1.未找到该学生
1.1.3.返回学生信息为空
1.2.显示该学生不存在
1.未找到该学生



在这些基本的工作流程中还存在分支，可使用备选过程来描述。

备选过程A：

该名学生没有学科成绩。

① 数据访问层返回学科成绩为空。

② 系统提示老师没有该学生的成绩。

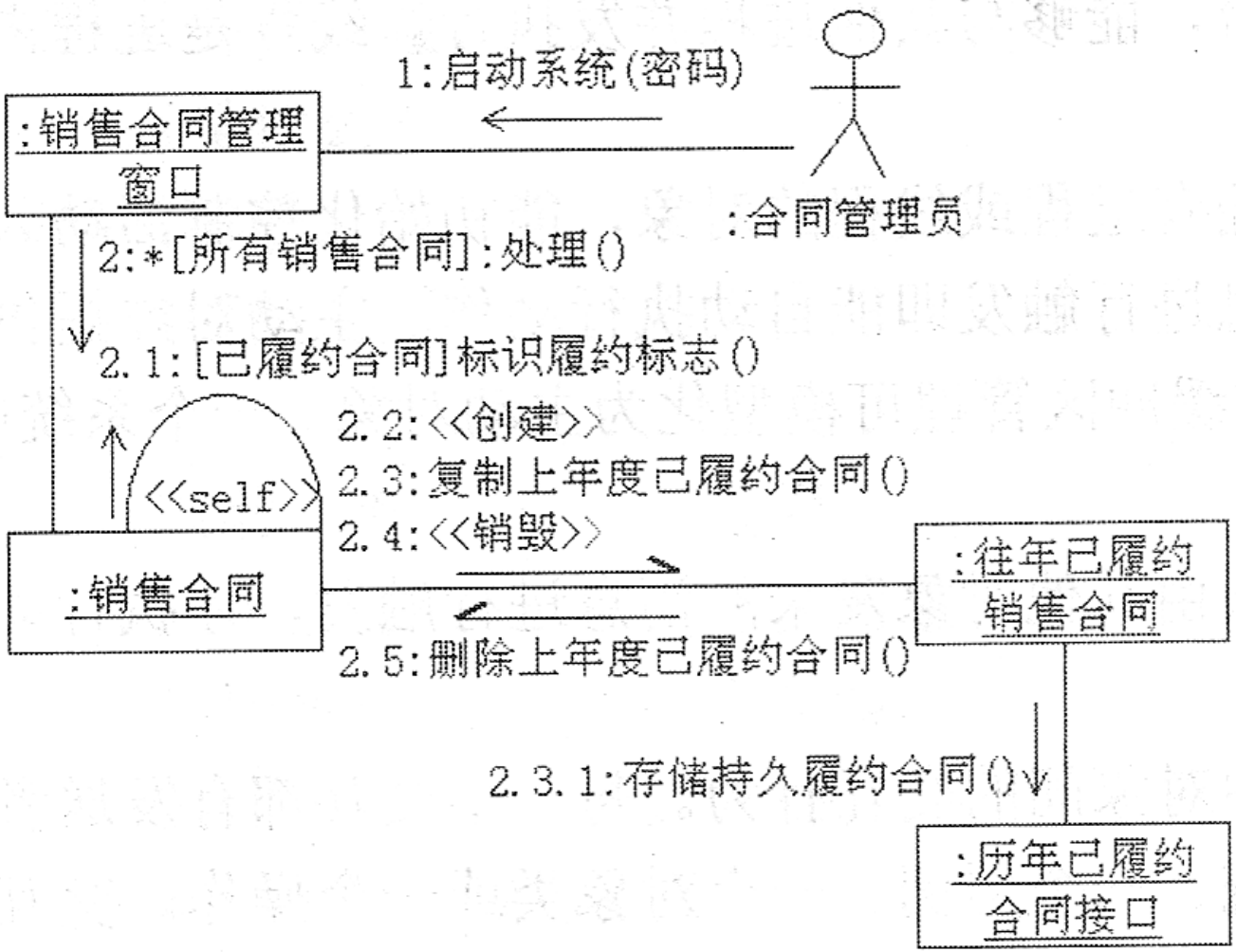
备选过程B：

系统没有该学生的信息。

① 数据访问层返回学生信息为空。

② 系统提示老师该学生不存在。

练习:履约合同检查的协作图　时序图?



review

- 交互图的作用和特点是什么？
- 协作图的作用和特点是什么？
- 绘制协作图的步骤有哪些？
- 时序图的作用和特点是什么？
- 绘制时序图的步骤有哪些？