用例图

软件需求获取的困难

- 需求：饮料问题

我要一瓶饮料…

差不多，但我要无糖饮料…

很好，不过我要绿茶的…

啊，没有大瓶的…

难捕获，易变！

大瓶的无糖绿茶饮料

什么是用例？

- 用例是软件工程或系统工程中对系统如何反应外界请求的描述，是一种通过用户的使用场景来获取需求的技术。
- 每个用例提供了一个或多个场景，每个场景说明了系统是如何和最终用户或其它系统进行互动，也就是谁可以用系统做什么。

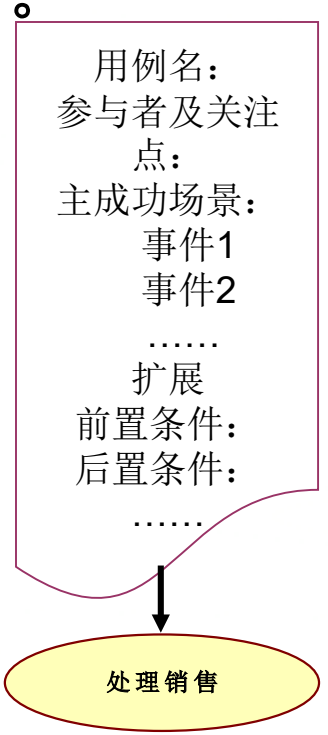
什么是用例？

Ivar Jacobson
UML三友之一，模块和模块架构、用例、现代业务工程、RUP等业界主流方法和技术的创始人。Ivar Jacobson International董事会主席。

- 用例的由来
  - 1967年Ivar Jacobson定义爱立信AXE系统的架构时开始书写使用场景（usage scenarios）。
  - 1986年他创造了术语“用例”（use case）。
  - 1995年他开始制定UML，使用“用例图”。
  - 1997年UML制定完成，成为工业标准。

什么是用例？

- 用例的定义
  - 定义1：用例是对一个参与者（actor）使用系统的一项功能时所进行的交互过程的一个文字描述序列。
  - 定义2：用例是系统、子系统或类和外部的参与者（actor）交互的动作序列说明，包括可选的动作序列和会出现异常的动作序列。



- 用例的本质
  - 用例是对一组动作的描述，系统通过执行这些动作将对用例的参与者产生可以看到的结果，用来描述参与者可以感受到的系统服务或功能。用例分析可以认为是对系统功能的分解。
  - 用例分析做的好坏，直接影响系统开发质量。

- 采用用例进行需求获取
  - 从使用系统的角度描述系统中的信息。
  - 描述用户提出的一些可见需求。
  - 对系统的行为进行动态描述。
  - 表示功能需求，非功能需求用补充文档描述。

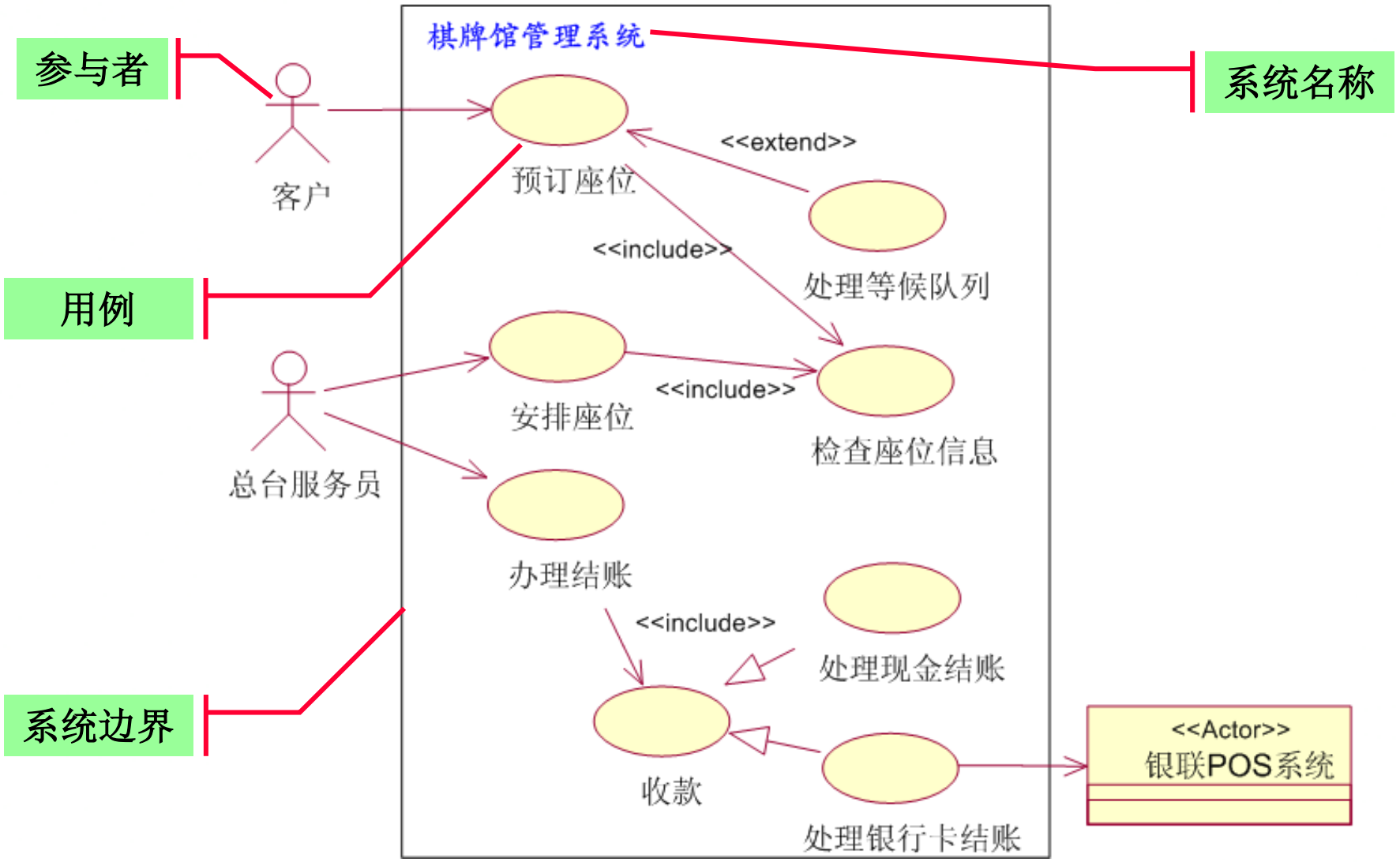
用例图

- 用例图是对用例的一种可视化的描述，描述系统功能的静态视图，以用例（Use Case）、参与者（Actor）为基本元素，从用户的角度描述系统功能。

  用例图的基本元素包括：

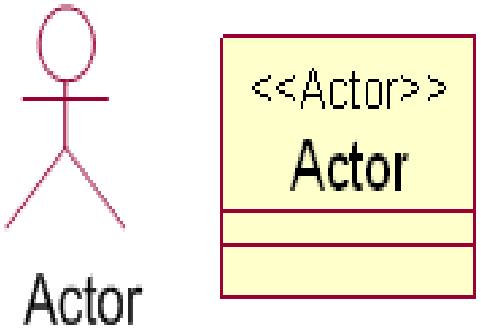
  – 参与者
  – 用例
  – 系统边界
  – 关系

用例图示例


参与者
系统名称
用例
系统边界
棋牌馆管理系统
客户
总台服务员
预订座位
<<extend>>
处理等候队列
<<include>>
安排座位
<<include>>
检查座位信息
办理结账
<<include>>
处理现金结账
收款
处理银行卡结账
<<Actor>>
银联POS系统

用例图——参与者

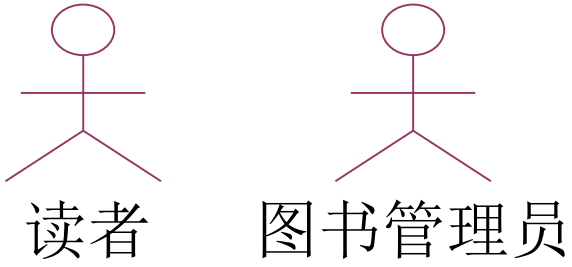
- 参与者
  - 参与者（也称为角色或活动者，**Actor**）是系统外部的一个人或者物，它以某种方式参与了系统的执行过程。
  - 参与者不是特指人，是指系统以外的，在使用系统或与系统交互中所扮演的角色。



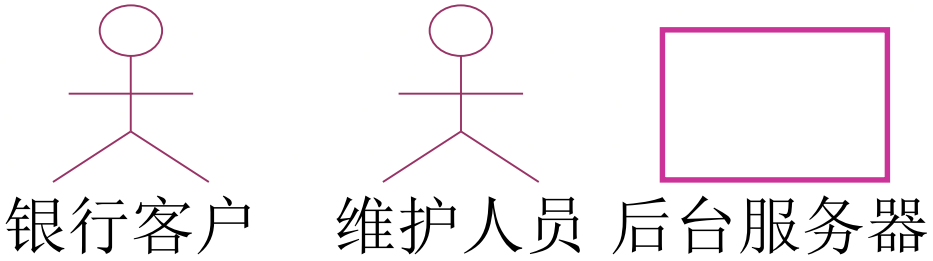
- 参与者的种类
  - 系统用户
  - 与所建造的系统交互的外部系统
  - 一些设备
  - 时钟：当系统需要定时触发时，时钟就是参与者。

- 怎样识别参与者？
  - 谁使用系统的主要功能？
  - 谁需要系统的支持以完成日常工作任务？
  - 谁从系统获取信息？
  - 谁负责维护和管理系统以保证其正常工作？
  - 系统需要使用哪些外部硬件设备？（打印机、扫描仪）
  - 系统需要和哪些外部系统交互？（跨行转账的外部银行系统、时间到了定时启动系统某功能）

- 例1：对一个图书管理系统来说，有哪些参与者？
  - 读者
  - 图书管理员



- 例2：对ATM系统来说，有哪些参与者？
  - 银行客户
  - ATM维护人员
  - 后台服务器

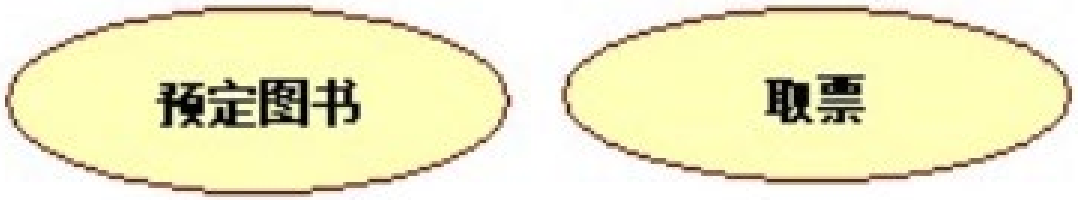


– 客户给销售员发来传真订货， 销售员下班前将当日订货单汇总输入系统。谁是系统的参与者？

销售员

用例图——用例

- 用例
  - 用例是用例图最重要的元素，是对业务工作的描述，或者说是对系统功能的陈述。
  - 在用例图中使用一个水平的椭圆来表示用例。
  - 例如：“预定图书”,“取票”等都是用例的实例。用例名一般为动宾结构 或者主谓结构。



- 识别用例
  - 主要分析各参与者目标，需要系统提供什么样的服务，或者说参与者是如何使用系统的。
  - 识别用例可以从以下问题入手：
    - 参与者使用该系统执行什么任务？
    - 参与者是否会在系统中创建、修改、删除、访问、存储数据？如果是的话，参与者又是如何来完成这些操作的？
    - 参与者是否会将外部的某些事件通知给该系统？
    - 系统是否会将内部的某些事件通知给参与者？

- 例1：对图书馆管理系统来说，有哪些参与者和用例？
  - 图书管理员
    - 管理读者信息
    - 管理图书信息
    - 登记借书
    - 登记还书
  - 普通读者：
    - 预订图书
    - 取消预订
    - 查询浏览图书信息
- 例2：对ATM系统来说，有哪些参与者和用例？
  - 银行客户
    - 查询
    - 取款
    - 转账
  - ATM维护人员
    - 维护系统
  - 后台服务器
    - 周期性操作

- 识别用例的注意事项：
  - 用例必须是由某一个参与者触发而产生的活动，即每个用例至少应该涉及一个参与者。
  - 如果存在与参与者不进行交互的用例，需要将其并入其他用例，或者是检查该用例相对应的参与者是否被遗漏。
  - 反之，每个参与者也必须至少涉及到一个用例，如果发现有不与任何用例相关联的参与者存在：
    - 仔细考虑该参与者是如何与系统发生对话的；
    - 由参与者确定一个新的用例；
    - 该参与者是一个多余的模型元素，应该将其删除。

用例图——边界

- 系统边界
  - 一个系统所包含的系统成分与系统外事物的分界线。系统的目标和范围。

例：学生成绩管理系统

•目标：

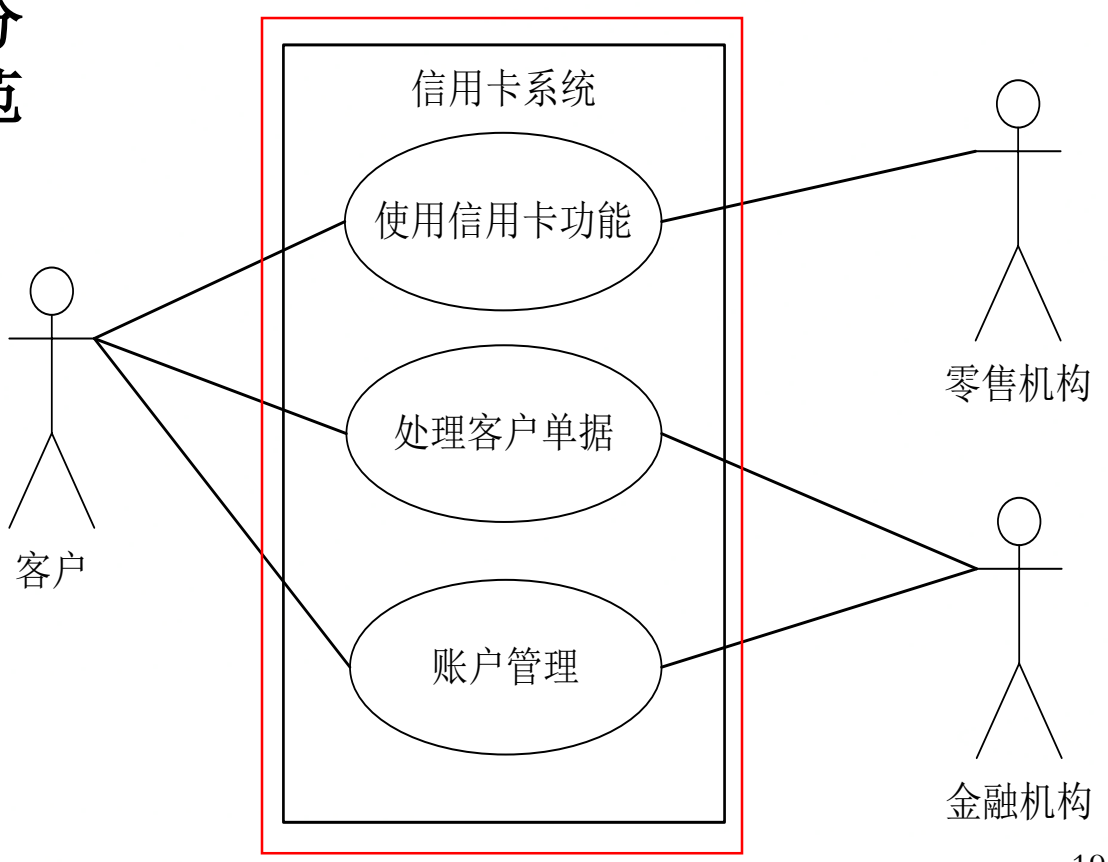
大学？中小学？

•范围：

单机、网络？

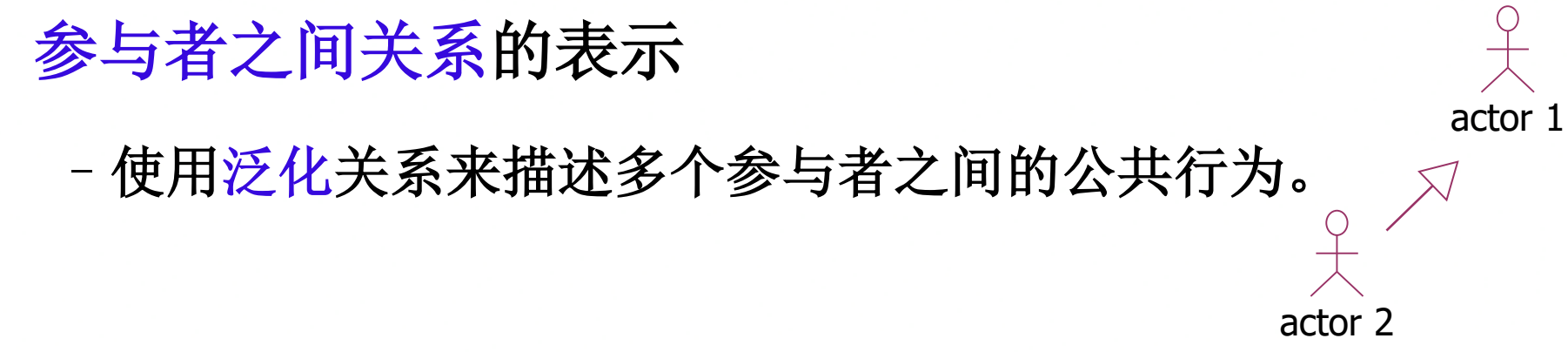
学籍？课程？

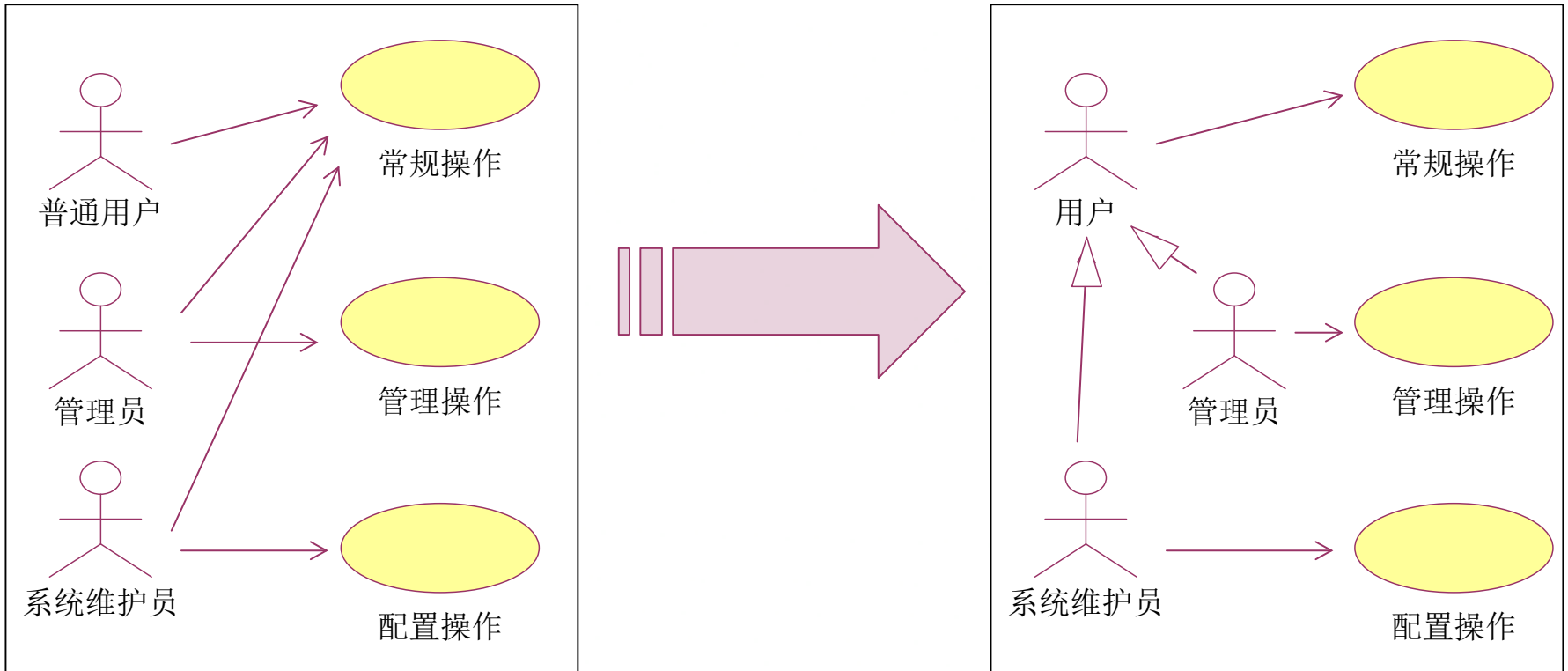
- 系统边界示例



用例图——参与者之间的关系

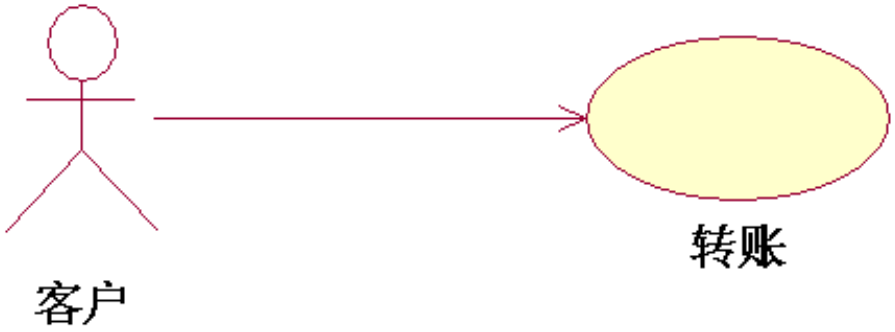
- 参与者之间关系的表示
  - 使用泛化关系来描述多个参与者之间的公共行为。





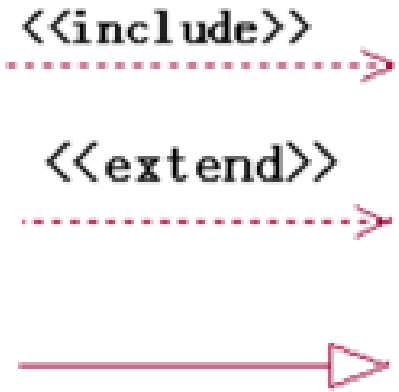
用例图——参与者与用例的关系

- 关联（accociation）关系：参与者与用例
  - 每个用例都由参与者启动（每个用例必须和一个参与者关联，有一个参与者来参与），除包含和扩展用例。
  - 无论用例和参与者是否存在双向数据交流（无论是参与者提供信息给系统，还是从系统获取信息），关联总是由参与者指向用例，只用单向箭头。



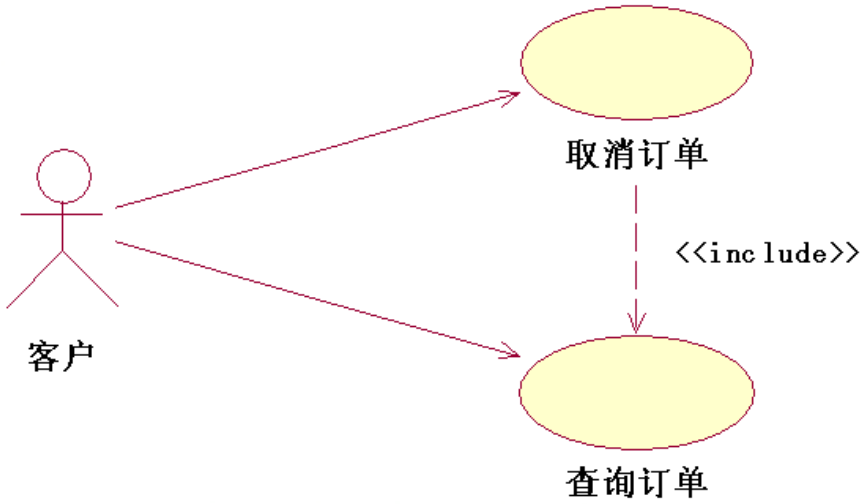
用例图——用例与用例的关系

- 用例与用例的关系
  - 用例与参与者有关联关系外，用例之间也存在着一定的关系，如下：
    - 包含（include）
    - 扩展（extend）
    - 泛化（generalization）

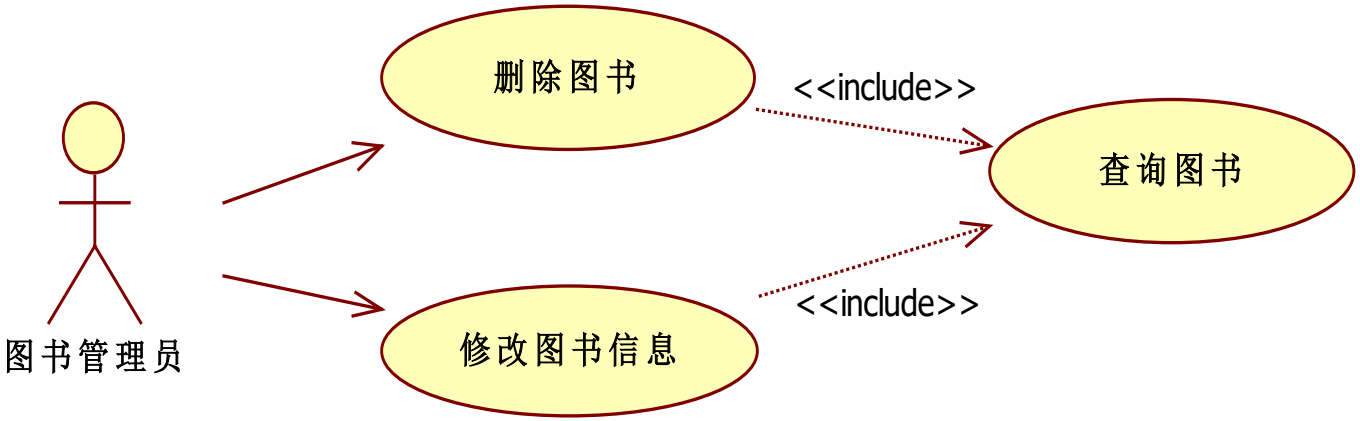


- 包含（include）关系
  - 是两个用例之间的关系，其中一个用例（称为基本用例）的行为包含了另一个用例（称为包含用例）的行为。也就是说基本用例会用到包含用例。
  - 使用包含用例的目的是封装一组跨越多个用例的相似动作，以便多个基用例复用。

类似于过程设计语言中将程序的某一段算法封装成一个子过程，然后从主程序中调用这一过程。

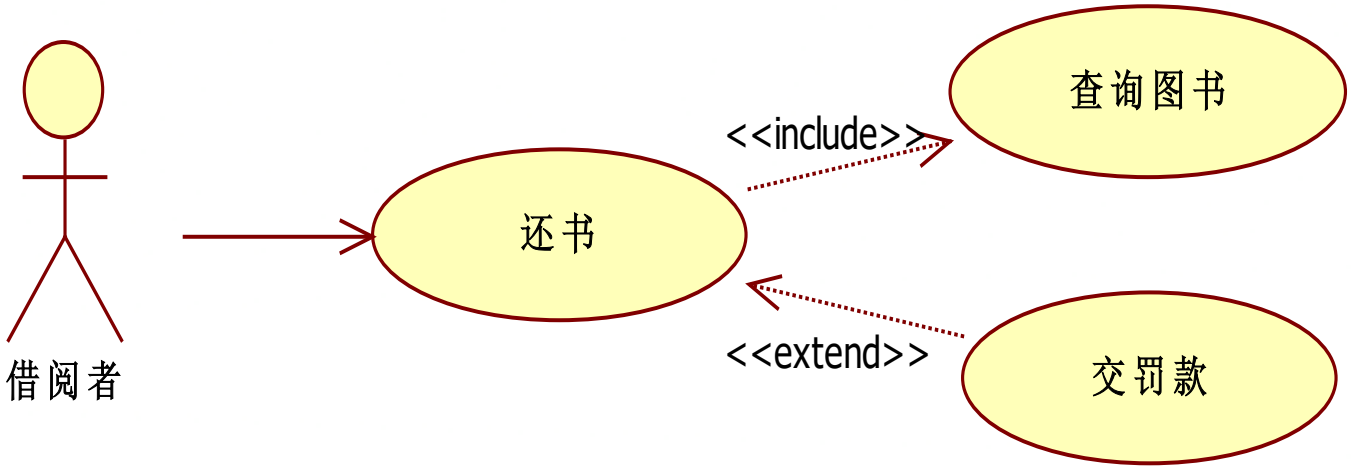


- 包含关系示例
  - 图书管理信息系统中有删除图书、修改图书信息用例。

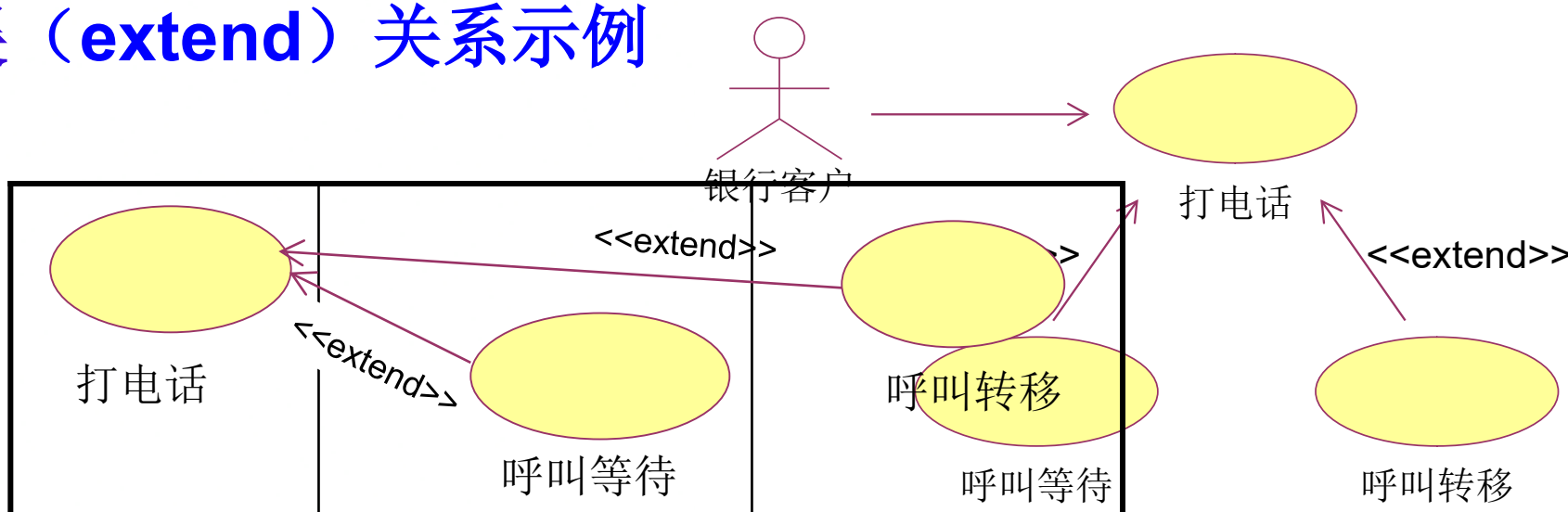


注：执行基本用例时，每次都必须调用被包含的用例，且被包含用例也可以单独执行。

- 扩展（extend）关系
  - 将基用例中一段相对独立并且可选的动作，用扩展用例加以封装，再让它从基用例中声明的扩展点上进行扩展，使基用例行为更简练，目标更集中。



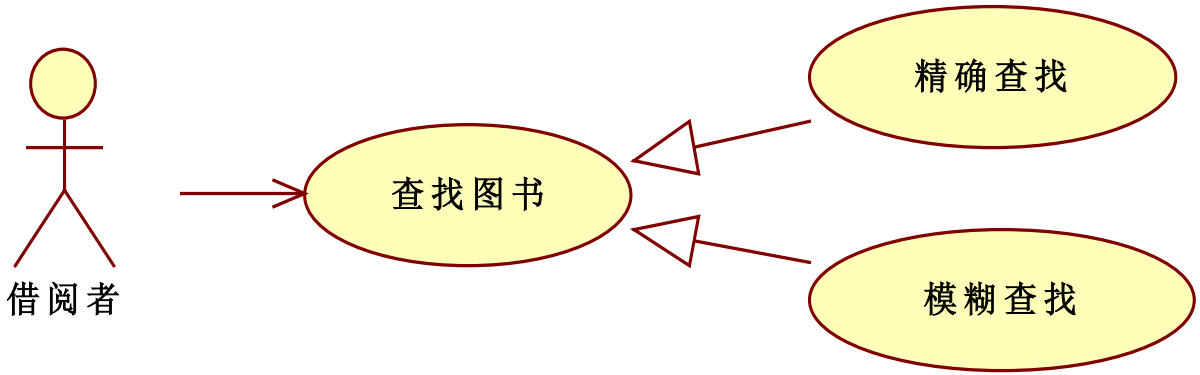
- 扩展（extend）关系示例



常规流：	扩展流：	扩展流：
1 拨号 2 建立通话链路 3 通话 4 挂机	1a 如果应答方正忙，用铃声提示应答方并保持拨号呼叫	1b 如果应答方无应答，进行呼叫转移

实际上相当于第一个用例的“扩展流”

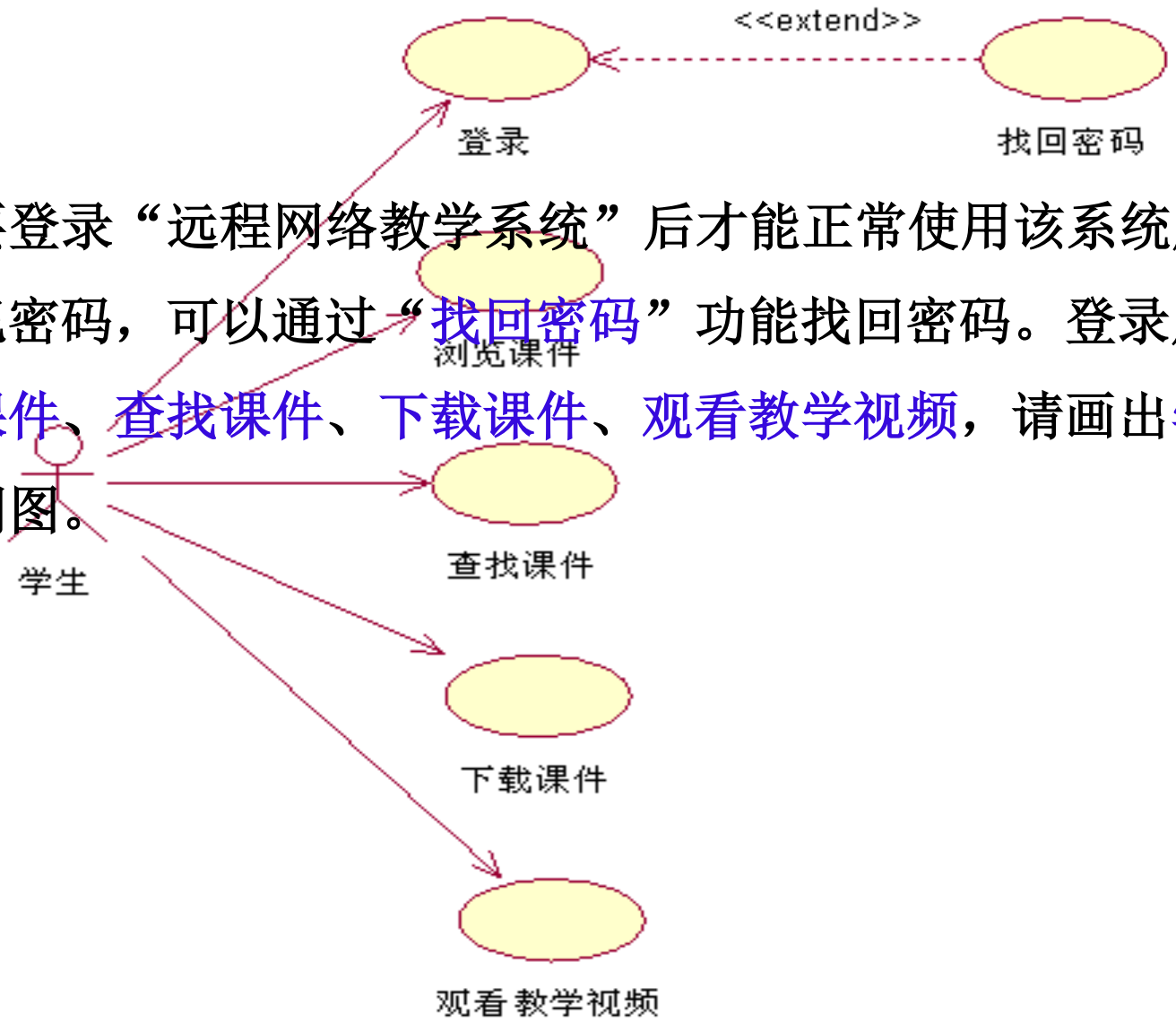
- 泛化（generalization）关系
  - 是一般与特殊的关系。当多个用例共同拥有一种类似的结构和行为的时候，可以将它们的共性抽象成为父用例，其它的用例作为泛化关系中的子用例。
  - 在用例的泛化关系中，子用例是父用例的一种特殊形式，子用例继承了父用例所有的结构、行为和关系。

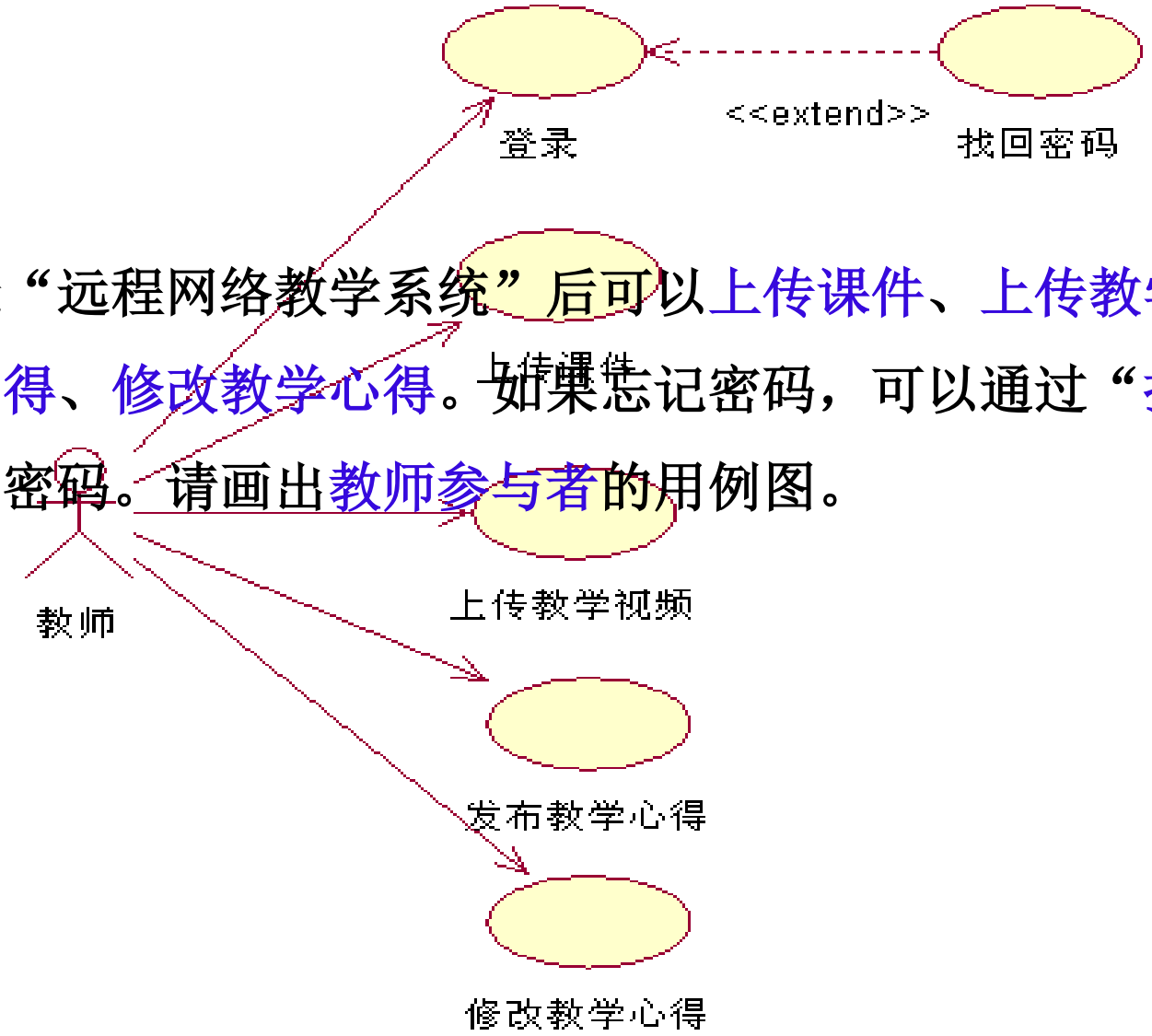


用例图案例——远程网络教学系统

- “远程网络教学系统”的功能需求包括：
  - 学生登录网站后，可以浏览课件、查找课件、下载课件、观看教学视频。
  - 教师登录网站后，可以上传课件、上传教学视频、发布教学心得、查看教学心得、修改教学心得。
  - 系统管理员负责对网站页面的维护，审核不法课件和不法教学信息，批准用户注册。

学生需要登录“远程网络教学系统”后才能正常使用该系统所有功能。如果忘记密码，可以通过“找回密码”功能找回密码。登录后学生可以浏览课件、查找课件、下载课件、观看教学视频，请画出学生参与者的用例图。





教师登录“远程网络教学系统”后可以上传课件、上传教学视频、发布教学心得、修改教学心得。如果忘记密码，可以通过“找回密码”功能找回密码。请画出教师参与者的用例图。

问题

问题：

1. 右图中的参与者有？

答案：1，4

2. 右图中的用例有？

答案：2，3，5，6

3. 2和3之间是什么关系？5和6呢？

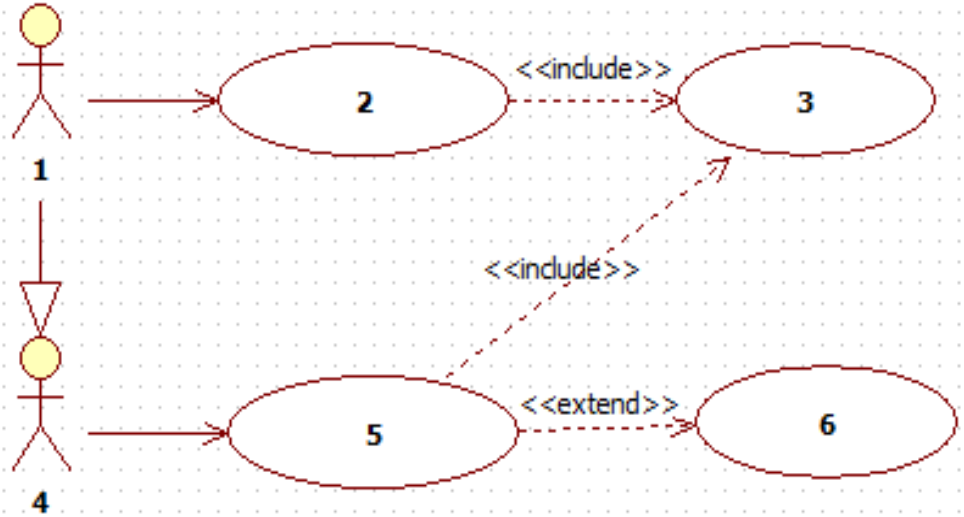
答案： 包含，扩展

4. 5缺少了3仍然是个完整的用例？

答案：不是

5. 4能够参与2吗？1能够参与5吗？

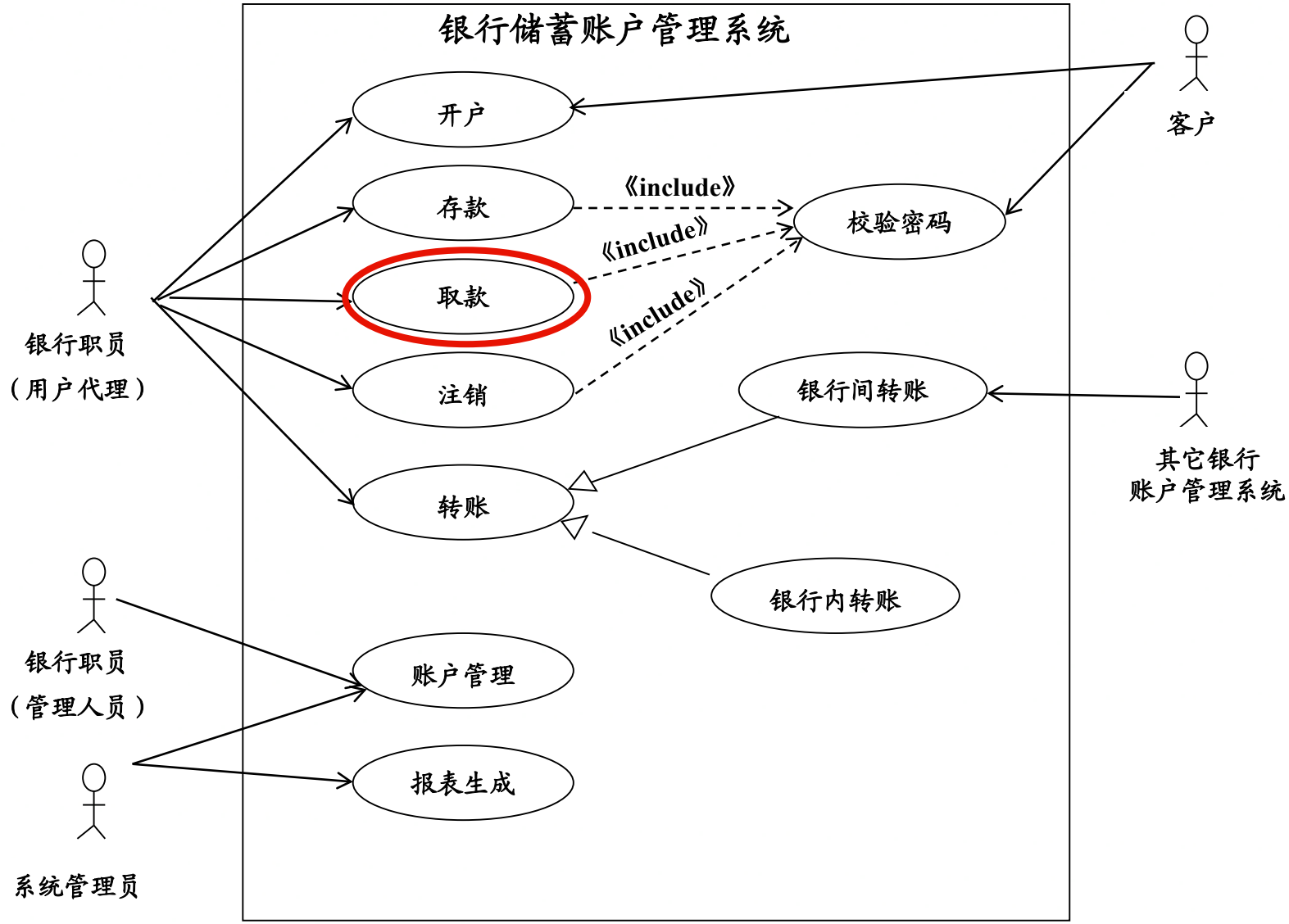
答案： 不可以，可以



用例描述

- 用例描述
  - 当用例图不能提供用例所具有的全部信息，需要使用文字描述那些不能反映在图形上的信息。用例描述是关于参与者和系统如何交互的规格说明。
  - 在编写用例描述的时候，应该只注重外部能力，不涉及内部细节。

用例描述


银行储蓄账户管理系统
开户
存款
取款
注销
转账
账户管理
报表生成
校验密码
银行间转账
银行内转账
《include》
《include》
《include》
客户
银行职员
（用户代理）
银行职员
（管理人员）
系统管理员
其它银行
账户管理系统

用例描述-取款用例描述

用例名称：取款
参与的执行者：银行职员（客户代理）
前置条件：一个合法的银行职员（客户代理）已登录到该系统
基本事件流：
1. 当选择取款功能时用例开始
2. 当输入客户信息（姓名、账号等）后
3. 输入并校验密码
4. 输入取款金额
5. 打印取款单，交客户签字
6. 建立取款事件记录，更新账户信息
7. 打印存折，用例结束

可选事件流：
2a：如果客户信息与账户不一致，显示错误信息，可以重新输入或结束用例
2b：如果该账户被冻结（如因挂失而冻结），显示冻结信息并结束用例
3a：校验密码时，如发现密码不一致，则重新输入密码，或用例结束
4a：如果该账户的余款小于取款金额，显示错误信息，要求重新输入
5a：客户签字之前的任何时刻，客户可以取消本次取款，用例结束

后置条件：如果取款成功，客户账户中的余额被更新（减少），否则余额不变。

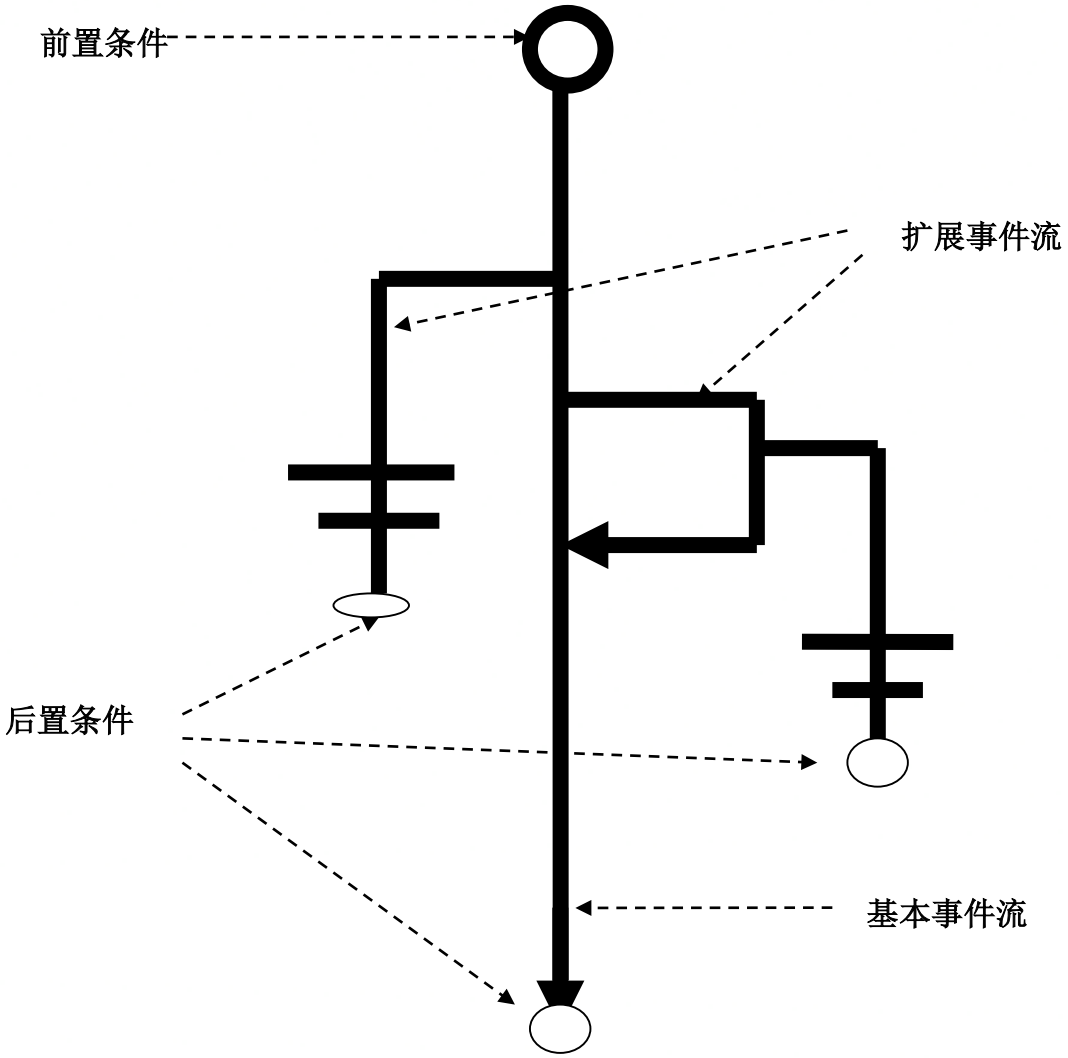
用例描述

- 用例图只能描述了系统的大概功能，是一种视图；用例描述才能表示系统活动的细节。对于用例的描述，可采用自然语言也可以采用用户自己定义的语言以及结构化语言。
- 描述用例时，应着重描述在外部用户看来会有什么样的行为，而不管该行为在系统内部如何实现的。用例描述的是一个系统做什么（what）的信息，并不说明怎么做（how），怎么做是设计模型的事。
- 用例驱动的开发方法要在概念级别上使用类图、交互图、状态图、活动图来做进一步描述。

- 用例描述有两种格式：一种是纯文本格式，另一种是表格形式。
- 在描述用例时并不要求将模板中每一项都写出来，可以根据需要进行取舍。
- 用例由参与者来执行，完成后将产生一个对参与者有价值的结果。
- 每个用例都需要一个简要的描述，随着分析的深入，描述也更加详细。

用例编号	[为用例制定一个唯一的编号，通常格式为UCxx]	
用例名称	[应为一个动词短语，让读者一目了然地知道用例的目标]	
用例概述	[用例的目标，一个概要性的描述]	
范围	[用例的设计范围]	
主参与者	[该用例的主Actor，在此列出名称，并简要的描述它]	
次要参与者	[该用例的次要Actor，在此列出名称，并简要的描述它]	
项目相关人利益说明	项目相关人	利益
	[项目相关人员名称]	[从该用例获取的利益]
	……	……
前置条件	[即启动该用例所应该满足的条件。]	
后置条件	[即该用例完成之后，将执行什么动作。]	
成功保证	[描述当前目标完成后，环境变化情况。]	
基本事件流	步骤	活动
	1	[在这里写出触发事件到目标完成以及清除的步骤。]
	2	……(其中可以包含子事件流，以子事件流编号来表示)
扩展事件流	1a	[1a表示是对1的扩展，其中应说明条件和活动]
	1b	……(其中可以包含子事件流，以子事件流编号来表示)
子事件流	[对多次重复的事件流可以定义为子事件流，这也是抽取被包含用例的地方。]	
规则与约束	[对该用例实现时需要考虑的业务规则、非功能需求、设计约束等]	

事件流模型



- 前置条件：指在用例启动时参与者（actor）与系统应置于什么状态，这个状态应该是系统能检测到的、可观测的。【描述启动该用例时系统具备的条件，如：登录成功】
- 后置条件：用例结束时系统应置于什么状态，这个状态也应该是系统能检测到的、可观测的。
- 基本事件流：基本事件流是对用例中常规、预期路径的描述，这是大部分事件所遇到的场景，它将体现系统的核心价值。
- 扩展事件流：主要是对一些异常情况、选择分支进行描述。

•事件流：

- 事件流描述了一个用例在执行时执行者与系统之间的交互过程。这个过程包括了多个分支，也就是说执行者在执行这个过程时可以有多个路线。
- 其中预期会成功的路线被称为基本事件流，剩下的其他路线被称为备选（扩展）事件流。

（一）基本事件流

基本事件流是对用例中常规和预期路径的描述。参与者通过这个路径来执行用例可以得到一个有价值的结果。

学生选课用例的基本流描述

1.学生选择要选修的课程。

2.系统通过财务系统检查学生是否缴费。

3.系统更新该学生所选课程。

4.系统显示学生所选的课程。

5.学生确认所选课程。

6.系统保存学生所选课程。

- 一般来说参与者和系统每一次交互过程可以分成以下4步来实现。
  - 1. 参与者向系统发送一个请求
  - 2. 系统验证请求和相关数据
  - 3. 系统进行数据处理，并改变它的内部状态
  - 4. 系统回传结果

- （二）备选（扩展）事件流

在用例的执行过程中，不一定都按照基本流的路径执行，在一些特殊情况下会改变执行其他的路径，这种路径称为备选（扩展）事件流

扩展事件流：

2.a 如果学生没有缴费，给出提示结束

5.a 如果学生没有确认，给出提示结束。

常用的扩展事件流是异常情况，根据经验数据有大约**60%-80%**的文本是用来描述异常情况的。

- 一个用例的扩展事件流可能很多，一般应该列出主要的扩展事件流。可以从以下两个方面考虑：

1.沿着基本事件流逐条寻找，在每个点上考虑：

这个点是否可以执行其他活动？

这个点上可能会出现那些异常？

是否有什么随时可能发生的行为？

2.从以下几个方面发现扩展事件流

参与者退出应用程序

参与者取消指定的操作

参与者请求帮助

参与者提供了异常数据

系统不可再用。

(三)事件流的循环和分支

基本事件流（循环）

1.学生输入所选课程编号。

2.系统记录输入数据

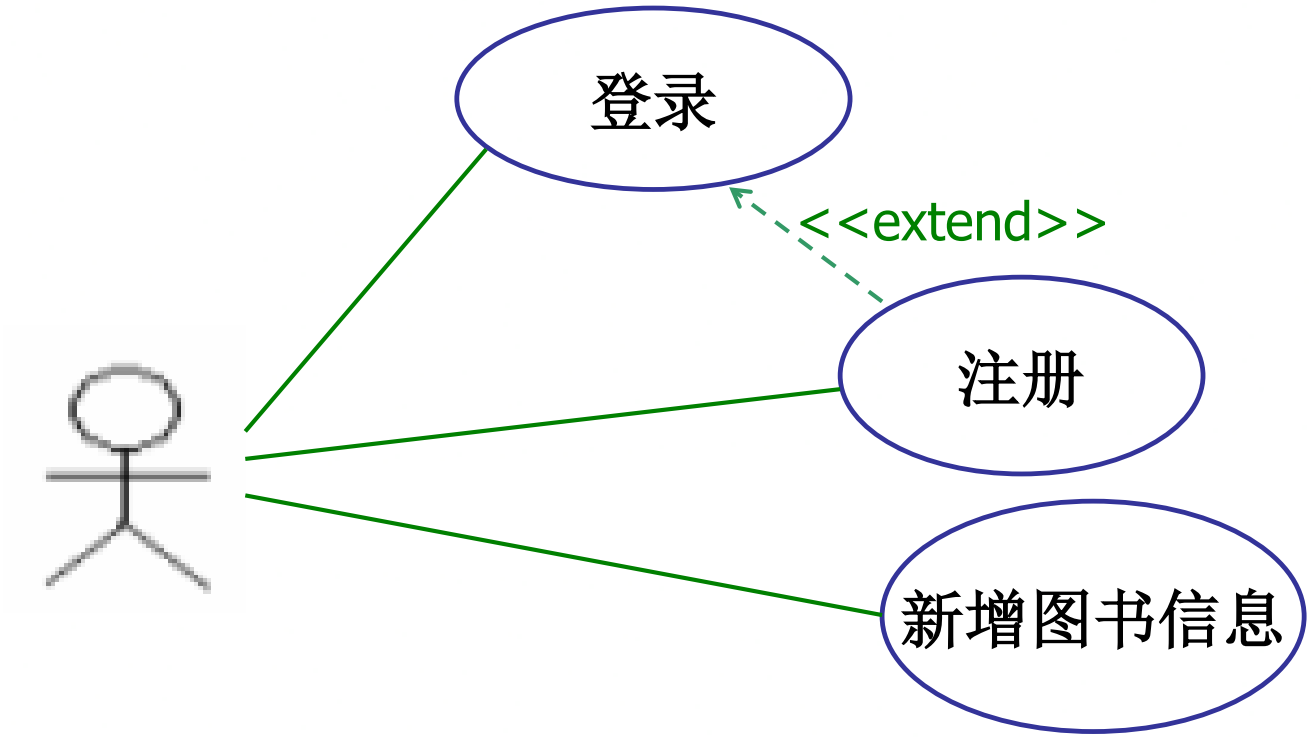
重复1-2步直到结束

3.....

如果出现分支情况，需要对事件进行分析，选择其中最容易成功的实现系统功能的分支作为基本事件流，其他分支作为扩展事件流。

在进行用例描述时，并不要求把每一项都写出来，而是根据需要进行相应的取舍：

下面是登录用例、注册用例和新增书籍信息用例的描述



用例名:登 录

简述:通过输入用户名和口令来使用系统的功能

角色:所有用户

前置条件:无

后置条件:登录成功,进入系统或退出登录或进入注册用例

基本事件流

1.输入用户名和密码

2.系统校验用户名和密码

3.用户进入系统

扩展事件流:

2a 用户名和密码不正确

if 登录次数小于三

重新登录

else

进入注册用例或退出登录

用例名:注 册

角色:所有用户未注册用户

前置条件:无

后置条件:注册成功,成为系统用户

基本事件流:

1.输入用户名

2.系统检测用户名

3.输入密码和确认密码

4.填写其他信息

5.系统检测输入密码一致

6.系统提示注册成功

扩展事件流:

1a 用户名已存在或小于6，重新输入

3a 两次输入密码不一致或小于6，重新输入

用例名称：新增书籍信息（UC01）

简要说明：录入新购书籍信息，并自动存储建档。

角色:图书管理员

前置条件：用户进入图书管理系统。

后置条件：完成新书信息的存储建档。

基本事件流：

1.图书管理员向系统发出“新增书籍信息”请求。

2.系统要求图书管理员选择要新增的书籍是计算机类还是非计算机类。

3.图书管理员做出选择后，显示相应界面，让图书管理员输入信息，并自动根据书号规则生成书号，

4.图书管理员输入书籍的相关信息，包括：书名、作者、出版社、ISBN号、开本。页数、定价。是否有CD-ROM。

5.系统确认输入的信息中，书名没有重名。

6.系统将所输入的信息存储建档。

扩展事件流：

5a）如果输入的书名有重名现象，则显示出重名的书籍，并要求图书管理员选择修改书名或取消输入

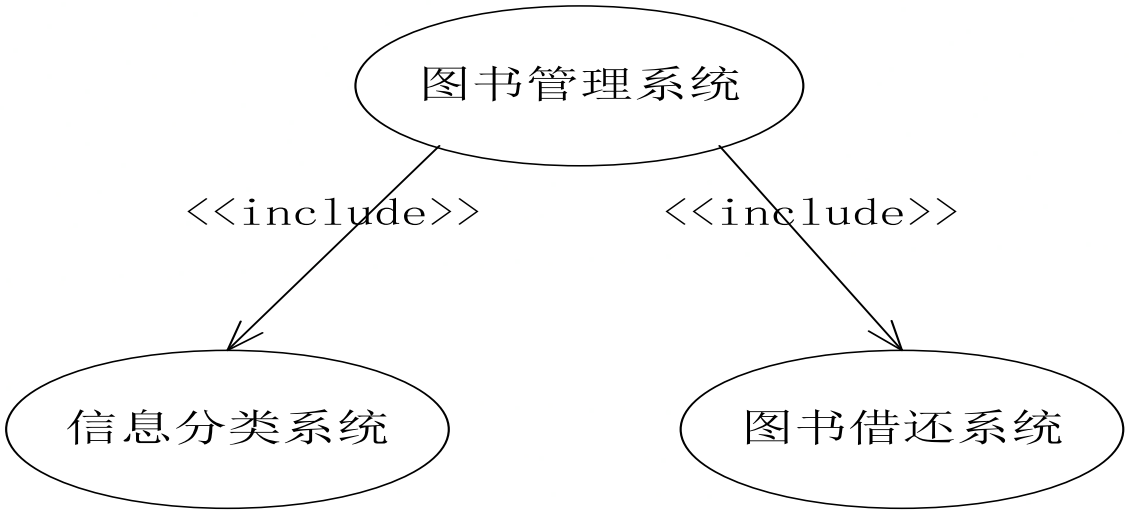
5a1）图书管理员选择取消输入，则结束用例，不做存储建档工作。

5a2)图书管理员选择修改书名后，转到5

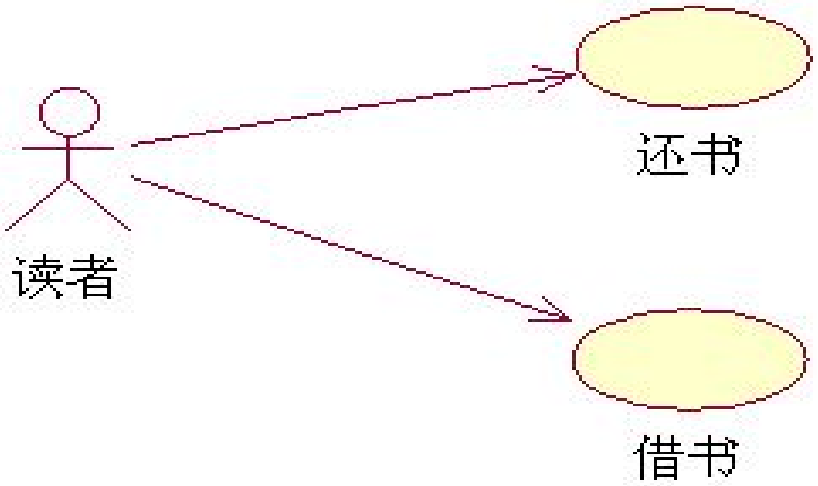
用例的粒度

- 用来描述用户目标大小的程度。分为三个层次：
  - 概述级
  - 用户目标级
  - 子功能级

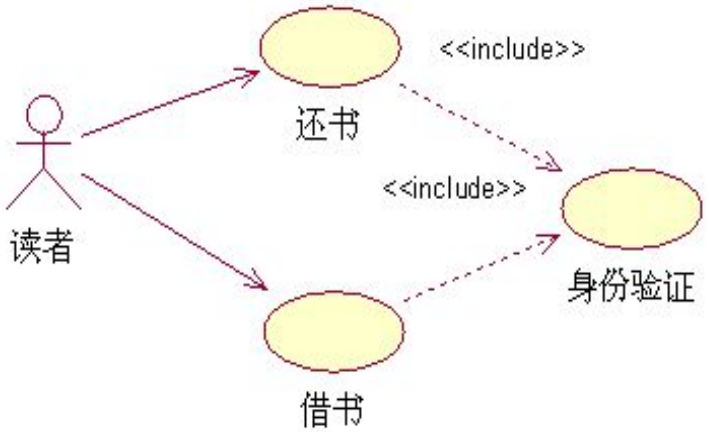
- **概述级**
  - 用来描述商业目标，可以包括多个用户目标级的大的用例。一般用于初期的需求讨论，也可以用作用户目标级用例的划分目录。例如:把整个系统看成一个用例或几个子系统集合
  - 概述级,参与者把整个系统看成一个用例。



- 用户目标级
  - 用来描述参与者或用户完成工作或使用系统的目的。一般用来描述某个人在某个时间、地点完成某项工作。
  - 目标级用例是对概述级进一步细化。



- 子功能级
  - 子功能级是比用户目标级在低一级的用例，除非是为了重用用例或其他特殊要求，一般建议在获取需求时不需要深入到这一层，否则容易出现可能取得的用例太多的情况。
  - 在进行用例分析时应侧重用户目标级的用例分析。
  - 子功能级用例是对目标级用例的进一步细化。



用例建模-步骤

- 确定系统边界
  - 系统边界：是一个系统所包含的所有系统成分和系统以外各种事物的分界线
  - 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，系统边界以外是与系统进行交互的人员、设备或外部系统
- 识别系统外部的参与者。
  - 将类似参与者组织成泛化的结构层次（参与者之间的关系）。
- 构建用例
  - 考虑每一个参与者期望的行为或需要系统提供的行为。
  - 将参与者放入到用例图中，并说明执行者与用例之间的通信路径。（参与者和用例之间的关系）
- 确定用例之间的关系。
- 描述用例

银行储蓄账户管理系统-实例分析

实现一个简化的银行储蓄账户管理系统，该系统是在银行的柜台上对客户办理活期储蓄业务。系统的需求陈述如下：

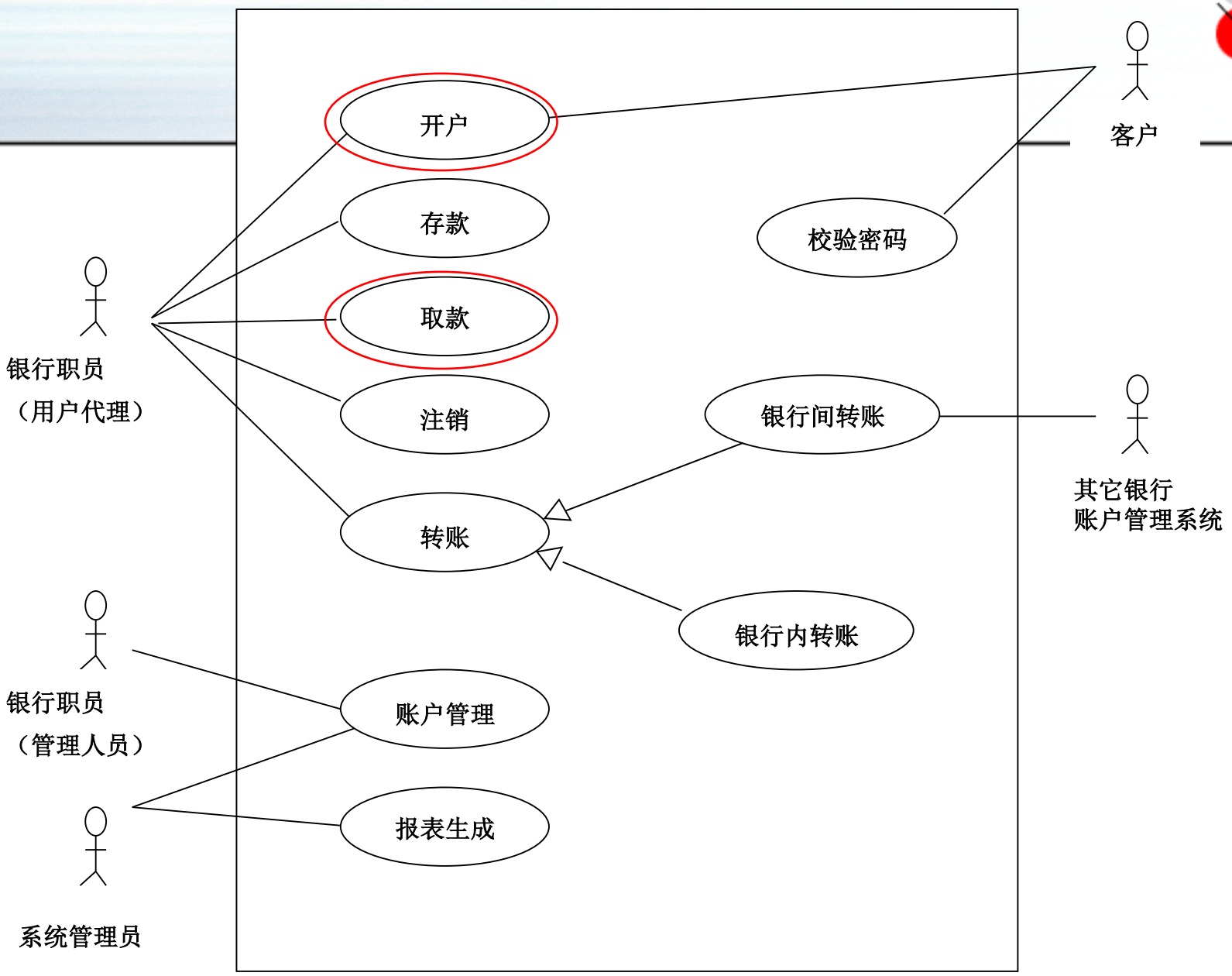
- 一个客户可以在多个银行中开设账户，一个客户也可在同一银行中开设多个不同的账户。
- 客户可以通过银行职员进行开户、存款、取款、转账、注销账户等活动。
- 其中转账指客户将自己的某个账户上的钱款转入同一银行的不同账户（称为银行内转账）或转入不同银行的账户（称为银行间转账）。
- 系统管理员负责系统的账户管理及业务报表的生成。

银行储蓄账户管理系统-实例分析

- 识别执行者
  - 客户：到银行办理储蓄业务的人，负责输入密码
  - 银行职员（客户代理）：银行工作人员，代表客户进行储蓄业务的操作
  - 银行职员（管理人员）：银行工作人员，根据客户的储蓄业务更新账户
  - 管理员：银行计算机的管理人员，负责账户的管理和业务报表的生成
  - 银行

银行储蓄账户管理系统-实例分析

- 识别用例
  - 从系统的需求陈述可知，银行职员（客户代理）需要系统提供开户、存款、取款、转账、注销账户等功能，这些功能都包含了校验密码的功能。
  - 系统管理员需要系统提供账户管理和报表生成功能。
  - 银行职员（管理人员）则参与了账户管理中的更新账户的功能。
  - 此外，转账功能可分为银行内转账和银行间转账，我们可将它们设计成三个用例，其中银行内转账用例和银行间转账用例都继承了基本转账用例。据此分析，得到该系统的用例图如下图所示。


开户
存款
取款
注销
转账
校验密码
银行间转账
银行内转账
账户管理
报表生成
客户
其它银行
账户管理系统
银行职员
（用户代理）
银行职员
（管理人员）
系统管理员

银行储蓄账户管理系统

银行储蓄账户管理系统-实例分析

开户用例描述

用例名称：开户

参与的执行者：银行职员（客户代理），客户

前置条件：一个合法的银行职员（客户代理）已登录到该系统

事件流：

1. 当选择开户功能时用况开始
2. 输入客户信息（姓名、地址、身份证号等）
3. 从账户管理系统获取新的账号
4. 请客户输入密码
5. 请客户再次输入密码
6. 如果两次密码不一致则回到第4步，否则继续
7. 在账户库中添加新账户
8. 打印存折，用况结束

后置条件：在账户库中增加了一个新账户，得到一张新存折

银行储蓄账户管理系统-实例分析

用况名称：取款
参与的执行者：银行职员（客户代理）
前置条件：一个合法的银行职员（客户代理）已登录到该系统
基本事件流：

1. 当选择取款功能时用况开始
2. 当输入客户信息（姓名、账号等）后
3. 输入并校验密码
4. 输入取款金额
5. 打印取款单，交客户签字
6. 建立取款事件记录，更新账户信息
7. 打印存折，用况结束

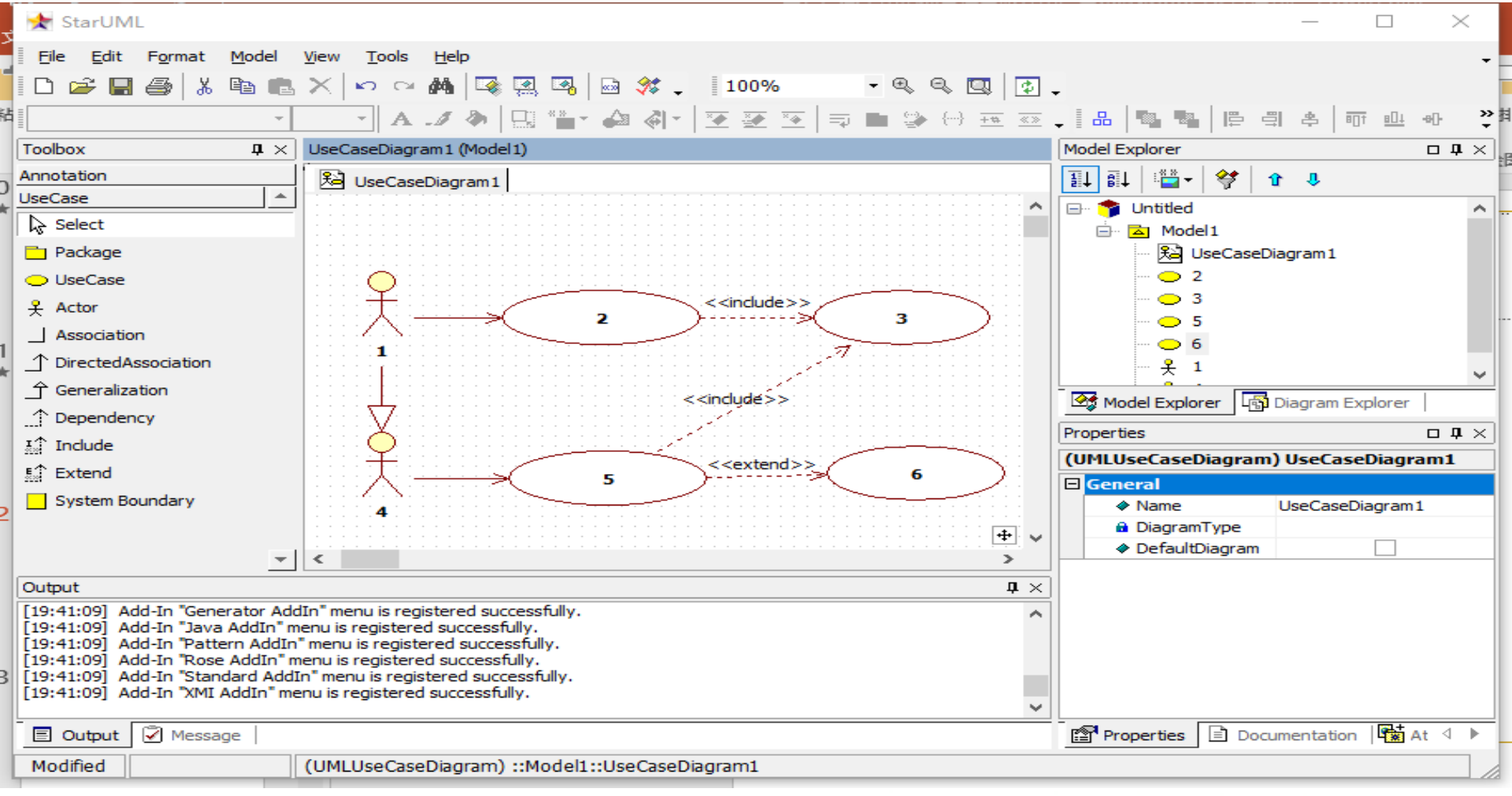
可选事件流：

2a：如果客户信息与账户不一致，显示错误信息，可以重新输入或结束用况
2b：如果该账户被冻结（如因挂失而冻结），显示冻结信息并结束用况
3a：校验密码时，如发现密码不一致，则重新输入密码，或用况结束
4a：如果该账户的余款小于取款金额，显示错误信息，要求重新输入
5a：客户签字之前的任何时刻，客户可以取消本次取款，用况结束

后置条件：如果取款成功，客户账户中的余额被更新（减少），否则余额不变。

用例图建模工具

- **StarUML**（功能较多，包括代码转换，**XML**格式保存）



思考：用例图的缺点

- 用例图的不足之处：
  - 不能表达非功能需求。可靠性、性能等无法表达。
  - 对不懂UML的客户或程序员来说难以理解。理解那些椭圆以及类似伪码的事件流并非易事。
  - 粗粒度。不涉及设计实现细节，只是一个功能划分，需要用其他工具进行辅助说明。
  - 图形符号表达能力有限。

用例图实践-建立用例图的步骤

- 1、识别参与者思路--为了识别用例首先要识别出参与者
- 谁是系统的主要用户
- 谁向系统提供信息
- 谁改变系统的数据
- 谁从系统获取信息
- 谁需要系统的支持以完成日常工作任务
- 谁负责日常维护、管理并保证系统正常运行
- 系统需要操纵哪些硬设备
- 系统需要和那些外部系统交互
- 谁（或什么）对系统运行产生的结果（值）感兴趣
- 时间、气温等内部外部条件

- 2、识别用例
  - 活动者希望系统执行什么任务？
  - 活动者在系统中访问哪些信息？（创建、存储、修改、删除等）
  - 需要将外界的哪些信息提供给系统？
  - 需要将系统的哪个事件告诉活动者？
  - 如何维护系统？