

# 第9章 用例图

(Usecase diagram & Activity diagram)

# 学习目标

◆ 学习完本章节，要求达到以下状态：

- 能够说明用例图和活动图的使用场合和表示方法
- 能够从用例图中读懂用户对系统的需求
- 能够通过用例图来表达用户的需求
- 能够读懂某个用例的用例描述
- 能够用活动图来表达某个用例的用例描述中的事件流

- **用例**是代表系统中各个项目相关人员之间就系统的行为所达成的契约。

例：在一个银行业务系统中，可能会有以下一些**用例**：

- 浏览帐户余额
- 列出交易内容
- 划拨资金
- 支付帐款
- 登录
- 退出系统
- 编辑配置文件
- 买进证券
- 卖出证券

## use case说明:

- **Use case**从使用系统的角度描述系统中的信息，即站在系统外部察看系统功能，并不考虑系统内部对该功能的具体实现方式。
- 使用**use case**可以促进与用户沟通，理解正确的需求，同时也可以用来划分系统与外部实体的界限，是OO系统设计的起点，是类、对象、操作的来源。

- 用例的一些特点：
  - （1）用例描述了用户提出的一些可见的需求；
  - （2）用例可大可小；
  - （3）用例对应一个具体的用户目标。
- 理论上可以把一个软件系统的所有Use Case画出来，但实际运用时只需把重要的、交互过程复杂的那些画出来。
- 需求有两种基本形式：**功能性**和**非功能性的**。  
不是所有的需求都要用use case表示出来。
- 问题：一个系统的需求包括哪些内容？
  - 可以参照需求大纲

- Use Case技术很容易使用，但也很容易误用。
- 正确使用Use Case 技术来做好Domain Modeling，以确保定义出正确的需求(right requirements)，然后开发出正确的系统(right system)，是保证OO软件开发成功的基础。

## 争论：

- 本质上，Use Case分析是一种**功能分解 (functional decomposition)**的技术，没有以对象观点为中心，因而有人认为Use Case 分析只是OOA/OOD的先导性工作，并非OOA/OOD过程的一部分；但也有人视其为OOA/OOD的一环。
- 不管怎样，Use Case是UML的一部分，Use Case是OO开发过程中的第一步。

# 用例图的概要

- ◆ 站在用户的角度描述用户对系统的需求。
- ◆ 在系统开发的初期阶段把握用户需求时使用，便于用户和系统开发者之间进行意向沟通。
- ◆ 从动态的侧面来描述系统提供给用户可使用的功能。
- ◆ 在分析设计阶段，以用例图为基础，来做成其他UML图形（顺序图，通信图，类图等）。
- ◆ 用例图由以下元素组成：
  - 参与者（actor）
  - 用例（usecase）
  - 关联
  - 系统边界



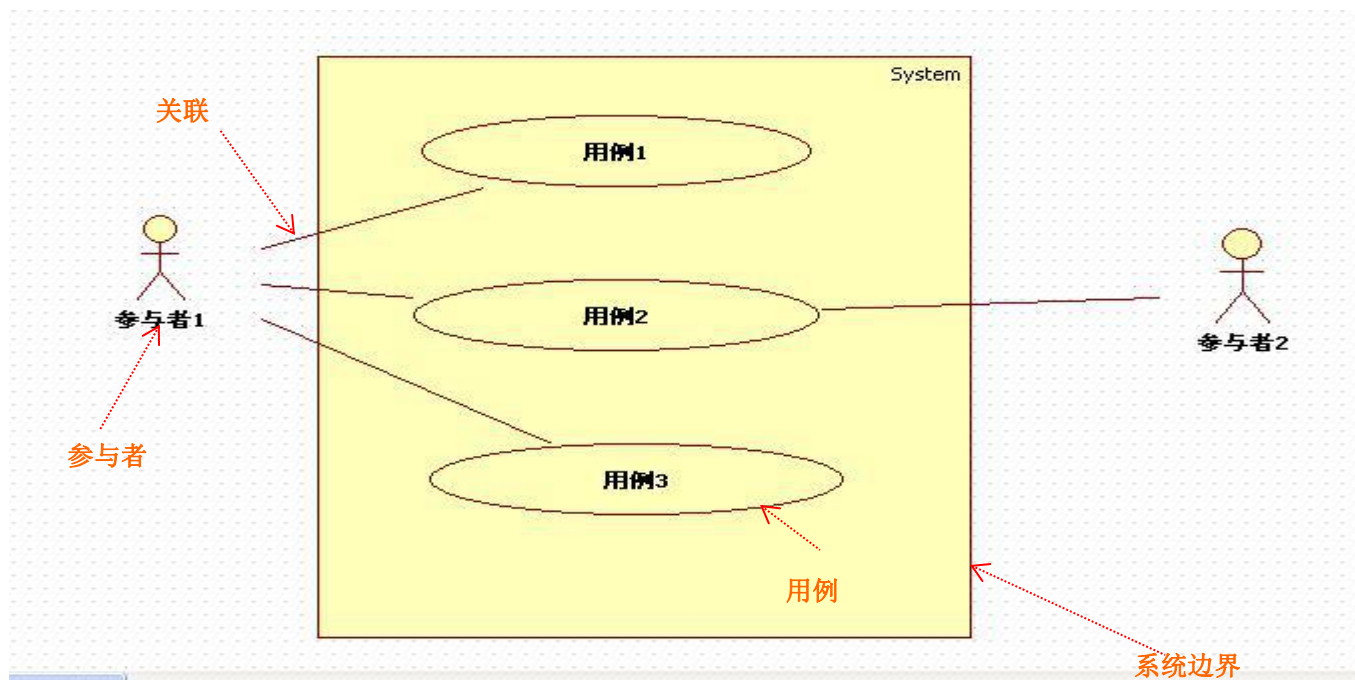
# 用例图的目的

- ◆ 在系统开发的初期阶段，基于以下目的做成用例图：
  - 明确开发系统的主要功能。
  - 明确开发对象范围。
  - 明确开发对象和外界的关系。

# 用例图的组成元素

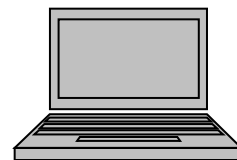
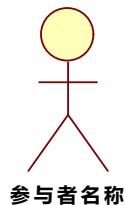
◆ 用例图由以下元素组成：

- 参与者（actor）
- 用例（usecase）
- 关联
- 系统边界



# 参与者

- ◆ 参与者是与对象系统进行信息交换的人，物，或者其他系统。
- ◆ 参与者存在于对象系统之外，既可主动与对象系统进行信息交换，也可以被动接受来自于对象系统的信息。。
- ◆ 参与者的表示方法

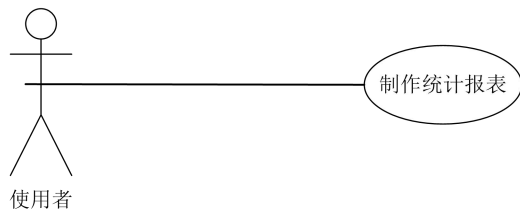


客户管理系统

- ◆ 可以成为参与者的有：
  - 系统使用者
  - 外部硬件
  - 现有的系统

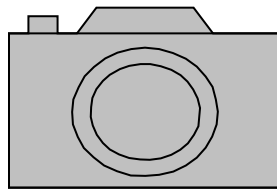
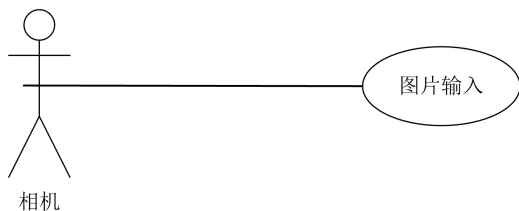
# 参与者的例子

## ◆ 参与者是使用者的例子



注：参与者是一个角色，并不是某个个别人。

## ◆ 参与者是外部硬件的例子

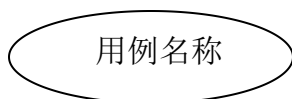


## ◆ 参与者是现有系统的例子



# 用例

- ◆ 用例是站在使用者的立场上看到的系统所提供的功能。
  - 是系统中的功能。
  - 一个用例表示一个功能，集中所有的用例，可完整描述该如何使用该系统。
  - 可以通过关联线与参与者连接，一个用例至少要与一个参与者相关联。
  - 给用例起名要站在使用者的立场上考虑
  - 可以用系统边界把用例框起来以区分系统内外。
- ◆ 用例的表示方法



用例名称

# 用例例子

参与者与系统的不同的交互都可以被量化为用例。  
自动售货机例子。

- ◆ **购买一瓶饮料。**在客户选择并支付以后，自动售货机弹出饮料。
- ◆ **进行定期维修。**维修技师对自动售货机进行定期服务，以让它保持良好的工作状态。
- ◆ **维修。**维修技师要对自动售货机做一些非预期的工作，排除运行过程中出现的故障。
- ◆ **装入物品。**存货管理员给自动售货机添加物品，补给饮料供应。

# 用例特点

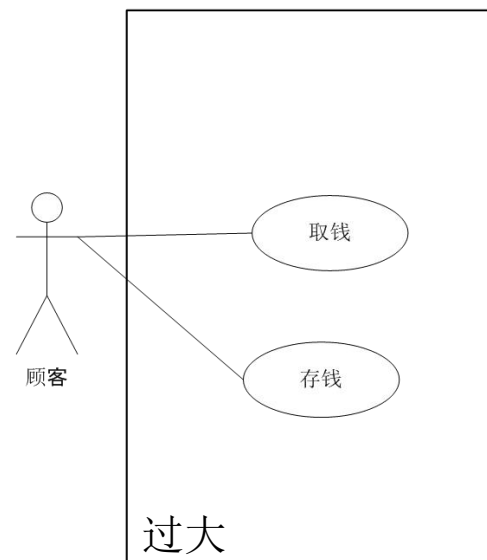
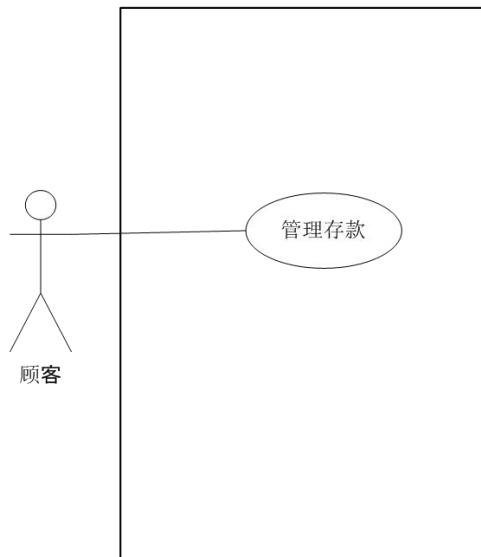
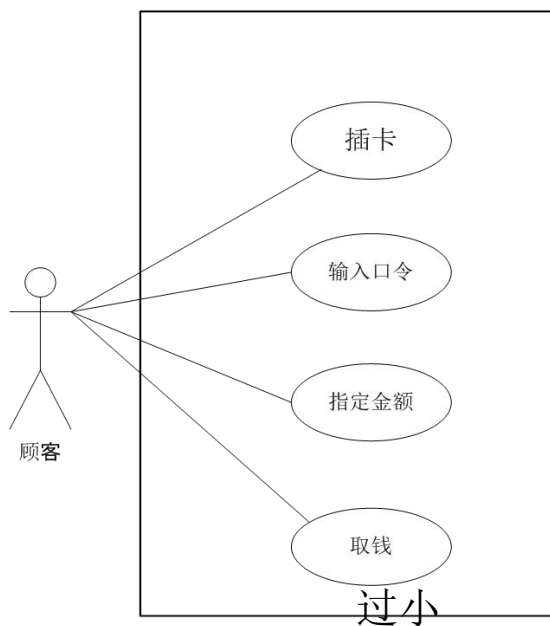
- 1 每个用例会涉及到一个或多个参与者以及系统本身。
- 2 用例涉及系统与其参与者之间的消息序列。
- 3 错误条件也是用例的一部分。

# 用例大小的设定

◆ 对系统中的用例进行分割的时候，要站在用户的立场上按如下原则进行分割：

- 一个用例结束，代表用户使用该系统的一个目的达到。
- 一个用例在执行过程中不宜发生中断。

使用ATM机系统的例子：



合适



# 用例模型准则

**1 首先确保系统边界。**

**2 确保关注参与者。** 每个参与者都应该有单一的、一致的目的。例如，计算机的拥有者会安装软件、配置数据库和发送邮件。由于这些功能有很大的不同，因此可以分为：系统管理员，数据库管理员，计算机用户3个参与者。

**3 每个用例必须给用户提供服务。** 用例应该表示成给用户提供服务的完整事务，不应该被定义的过于狭窄。

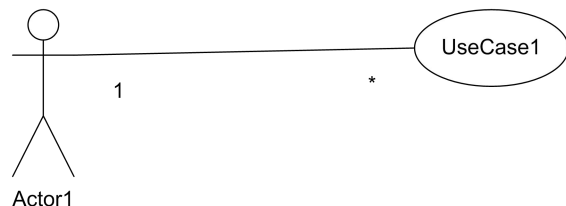
例如，电话系统中，拨打电话就不是一个好用例。它仅是打电话用例的一部分。

**4 用例是非形式化的。** 它不是一种形式化的机制，而是从以用户为中心的角度来识别和组织系统功能。

# 参与者的关系

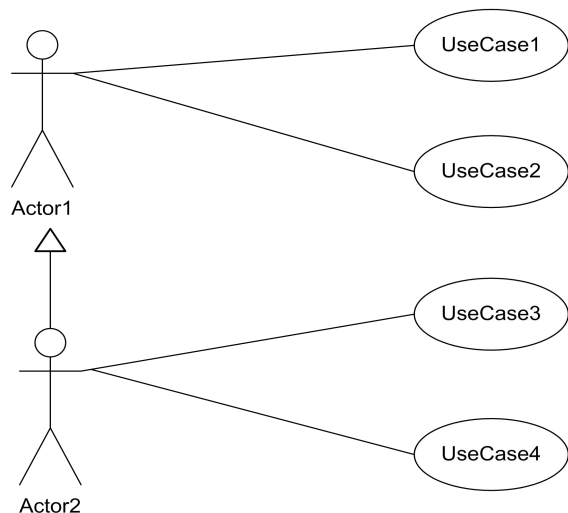
## ◆ 参与者和用例之间的关联关系

- 表示参与者和该用例之间存在着某种通信。
- 该关联也可以设定多重度。



## ◆ 参与者之间的泛化关系

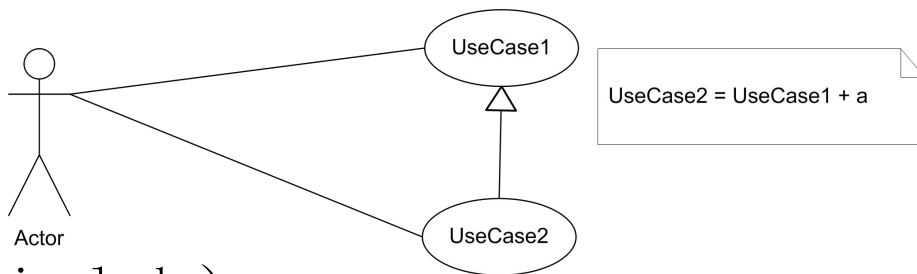
- 某一个参与者相关联的所有用例，都和另外一个参与者相关联。
- 可减少关联，简化用例图



# 用例之间的关系

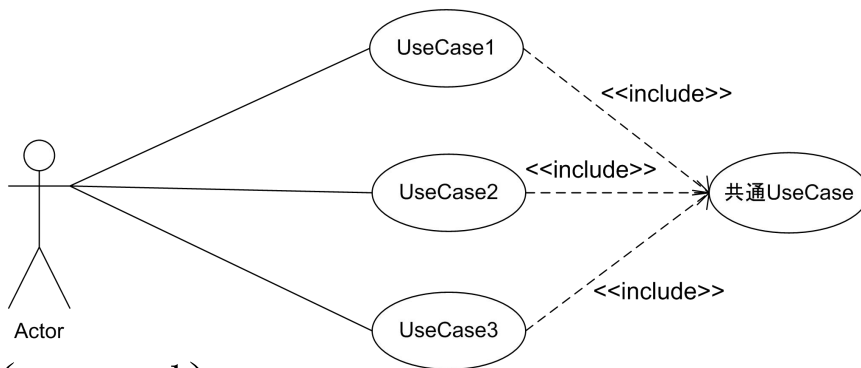
## ◆ 泛化

- 一个用例的流程默认情况下总是被另外一个用例所使用。



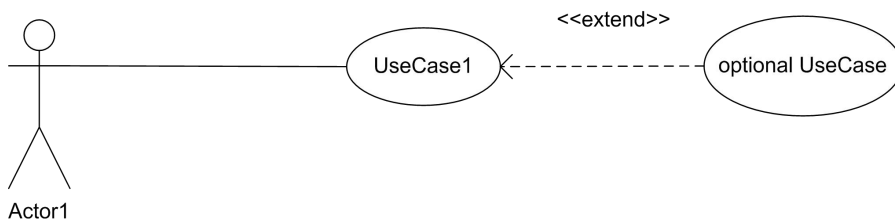
## ◆ 包含 (include)

- 多个用例的共通部分进行分离，用包含 (include) 关系进行连接。



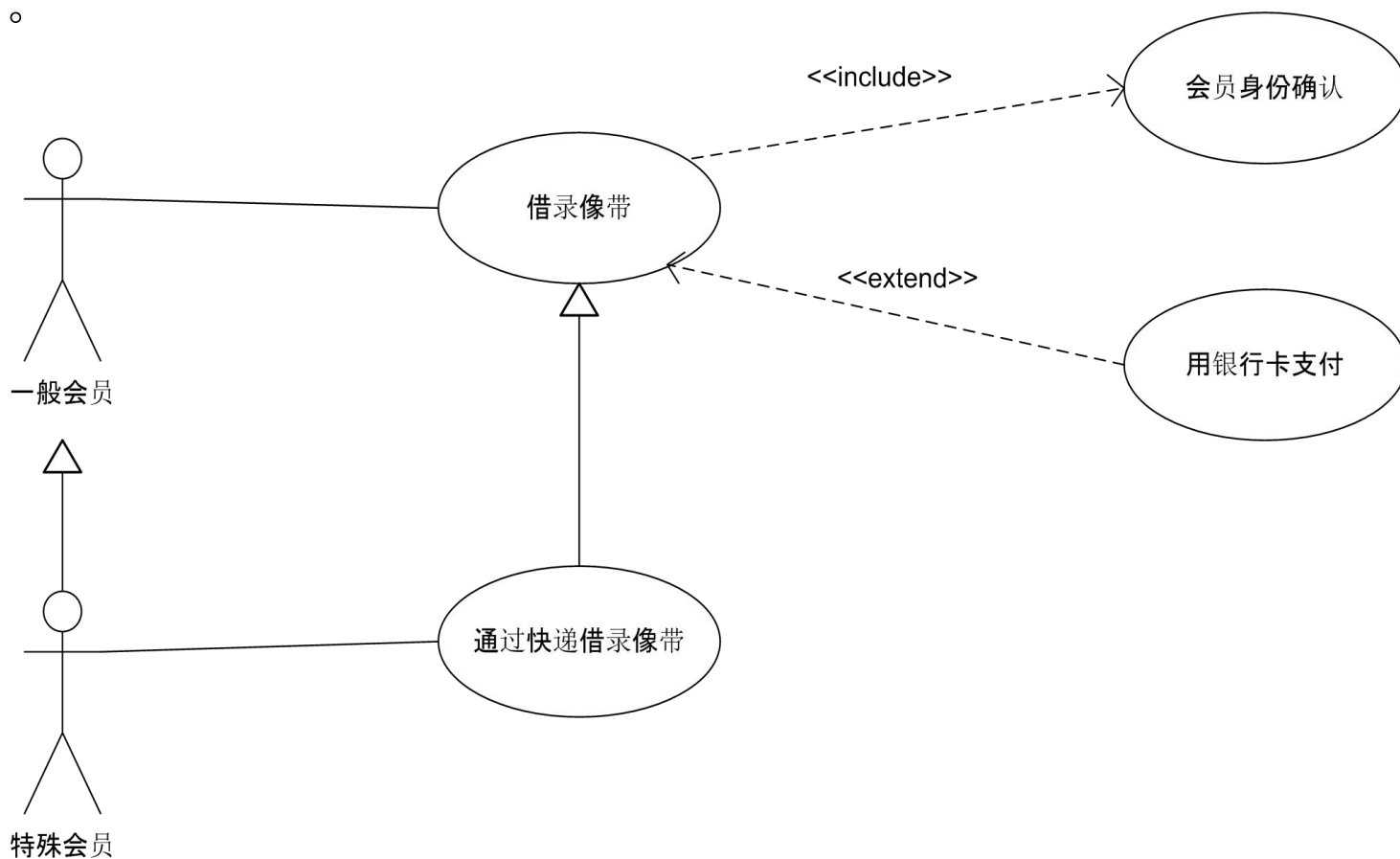
## ◆ 扩张 (extend)

- 有条件有选择地被执行的用例。



# 关系的例子

◆ 以录像带租赁业务的例子，来说明参与者之间，用例之间的各种关系。



# 用例描述

- ◆ 用例描述是对用例图的补充，用例描述和用例图合称为用例模型。  
(UseCase Model)
  - 可详细描述用例内部的业务流程。
  - 避免仅靠用例图造成的误解。
- ◆ UML中没有明确规定用例描述的标准格式，一般情况下应该包括：
  - 概要
    - 用几行文字描述用例的作用，目的等。
  - 脚本
    - 描述一个用例在某种条件下的具体执行流程。
  - 事件流
    - 描述用例在各种条件下的具体流程。

# 脚本和事件流

- ◆ 脚本用一段文字来描述用例执行时的具体流程。分为
  - 基本脚本  
正常运行能达到用例目的脚本。
  - 例外脚本  
出现异常情况不能达到用例目的的脚本。
- ◆ 事件流描述用例在各种条件下的具体执行流程，一般包括：
  - 前提条件  
为用例开始而必须满足的条件。。
  - 事后条件  
用例结束后对系统造成的影响。
  - 基本流  
正常处理时的基本流程。
  - 代替流  
出现概率低，但用例也还算是正常结束的流程。
  - 例外流  
发生错误时的流程。

例：用例模板

- **用例名称**：处理订单
- **标识符**：UC1701
- **用例描述**：当一个订单初始化或者被查询的时候这个用例开始。它处理有关订单的初始化定义和授权等问题。当订单业务员完成了同一个顾客的对话的时候，它就结束了。
- **参与者**：订单业务员
- **优先级**：1
- **状态**：通过审查
- **前置条件**：订单业务员登录进系统
- **后置条件**：下订单；库存数目减少

- **基本操作流程：**

- 1. 顾客来订购一个吉他，并且提供信用卡作为支付手段。.....
- 2. ....

- **可选操作流程：**

- 顾客来订购一个吉他，并且使用汇票的方式。.....
- 顾客来订购一个风琴，并且提供信用卡作为支付手段。.....
- 顾客使用信用卡下订单，但那张信用卡是无效的。
- 顾客来下订单，但他想要的商品没有存货。



- 被泛化的用例：无
- 被包含的用例：登录 (UC1706)。
- 被扩展的用例：无
- 修改历史记录：
  - 张三，定义基本操作流程，2003年5月4日
  - 张三，定义可选操作流程，2003年5月8日

# 用例描述的一些常见错误分析

- 在描述用例时易犯的错误包括：
  - 只描述系统的行为，没有描述actor的行为。
  - 只描述actor的行为，没有描述系统的行为。
  - 设定对用户界面的设计的要求。
  - 过于冗长。

- 例1：下面是一个用例描述的片断：

Use Case: Withdraw Cash（提取现金）

参与者：Customer

主事件流：

1. 储户插入ATM卡，并键入密码。
2. 储户按 “Withdrawal” 按钮，并键入取款数目。
3. 储户取走现金、ATM卡并拿走收据。
4. 储户离开。

上述描述中存在的问题：

- 只描述了参与者的动作序列，没有描述系统的行为。
- 改进的描述如下：

# Use Case: Withdraw Cash （提取现金）

参与者: Account Holder

主事件流:

1. 通过读卡机，储户插入ATM卡。
2. ATM系统从卡上读取银行ID、帐号、加密密码、并用主银行系统验证银行ID和帐号。
3. 储户键入密码，ATM系统根据上面读出的卡上加密码，对密码进行验证。
4. 储户按“FASTCASH”按钮，并键入取款数量，取款数量应该是5美元的倍数。
5. ATM系统通知主银行系统，传递储户帐号和取款数量，并接收返回的确认信息和储户帐户余额。
6. ATM系统输出现金，ATM卡和显示帐户余额的收据。
7. ATM系统记录事务到日志文件。

- 例2：下面是一个用例描述的片断：

Use Case: Withdraw Cash

参与者: Customer

主事件流：

1. ATM系统获得ATM卡和密码。
2. 设置事务类型为 "Withdrawal"。
3. ATM系统获取要提取的现金数目。
4. 验证帐户上是否有足够储蓄金额。
5. 输出现金，数据和ATM卡。
6. 系统复位。

上述描述中存在的问题：

- 只描述了ATM系统的行为，而没有描述参与者的行为。这样的描述很难理解，验证和修改。
- 改进的描述同例1。

- 例3：下面是一个用例描述的片断：

Use Case: Buy Something

参与者: Customer

主事件流：

1. 系统显示 “ID and Password” 屏幕。
  2. 顾客键入ID和密码，然后按OK按钮。
  3. 系统验证顾客ID和密码，并显示 “Personal Information” 屏幕
  4. 顾客键入姓名、街道地址、城市、州、邮政编码、电话号码，然后按OK按钮。
  5. 系统验证用户是否为老顾客。
  6. 系统显示可以卖的商品列表。
  7. 顾客在准备购买的商品图片上点击，并在图片旁边输入要购买的数量。选购商品完毕后按 “Done” 按钮。
  8. 系统通过库存辅助系统验证要购买的商品是否有足够库存。
- ...etc.

上述描述中存在的问题：

- 对用户界面的描述过于详细，对于需求文档来说，详细的UI描述对**获取需求**并无帮助。
- 改进的描述可以如下所示：

# Use Case: Buy Something

参与者: Customer

主事件流:

1. 顾客使用ID和密码进入系统。
  2. 系统验证顾客身份。
  3. 顾客提供姓名、地址、电话号码。
  4. 系统验证顾客是否为老顾客。
  5. 顾客选择要购买的商品和数量。
  6. 系统通过库存辅助系统验证要购买的商品是否有足够库存。
- ... etc.



- 例4：下面是一个用例描述的片断：

Use Case: Buy Something

参与者: user(或Customer)

主事件流：

1. 顾客使用ID和密码进入系统。
2. 系统验证顾客身份。
3. 顾客提供姓名。
4. 顾客提供地址。
5. 顾客提供电话号码。
6. 顾客选取商品。
7. 顾客确定商品的数量。
8. 系统验证顾客是否为老顾客。
9. 系统打开到库存系统的连接。
10. 系统通过库存系统请求当前库存量。
11. 库存系统返回当前库存量。
12. 系统验证购买商品的数量是否小于库存量。
- ... etc.

上述描述中存在的问题：

- 对用例的描述过于冗长，可以采用更为简洁的描述方式，如合并类似的数据项（步骤3至步骤5），提供抽象的高层描述（步骤9至12）等。
- 改进的描述可以如下所示：

Use Case: Buy Something

参与者: user(或Customer)

主事件流：

1. 顾客使用ID和密码进入系统。
  2. 系统验证顾客身份。
  3. 顾客提供个人信息(包括姓名、地址、电话号码), 选择要购买的商品及数量。
  4. 系统验证顾客是否为老顾客。
  5. 系统使用库存系统验证要购买的商品数量是否少于库存量。
- ... etc.

# Use Case图的建立步骤

- (1) 找出系统外部的参与者和外部系统，确定系统的边界和范围；
- (2) 确定每一个参与者所期望的系统行为；
- (3) 把这些系统行为命名为Use Case；
- (4) 使用泛化、包含、扩展等关系处理系统行为的公共或变更部分；
- (5) 编制每一个Use Case的脚本；
- (6) 绘制Use Case图；
- (7) 区分主事件流和异常情况的事件流，可以把表示异常情况的事件流作为单独的Use Case处理；
- (8) 细化Use Case图，解决Use Case间的重复与冲突问题。

用例分析的例子：

- 假设要开发一个工资支付系统，其中有这样的需求：
  - 每个星期五以及每个月的最后一个工作日，系统自动生成一份员工工资册。
- 分析1：考虑下面的用例图：



- 分析2: System Clock是设计时的问题。考虑改用下面的用例图:

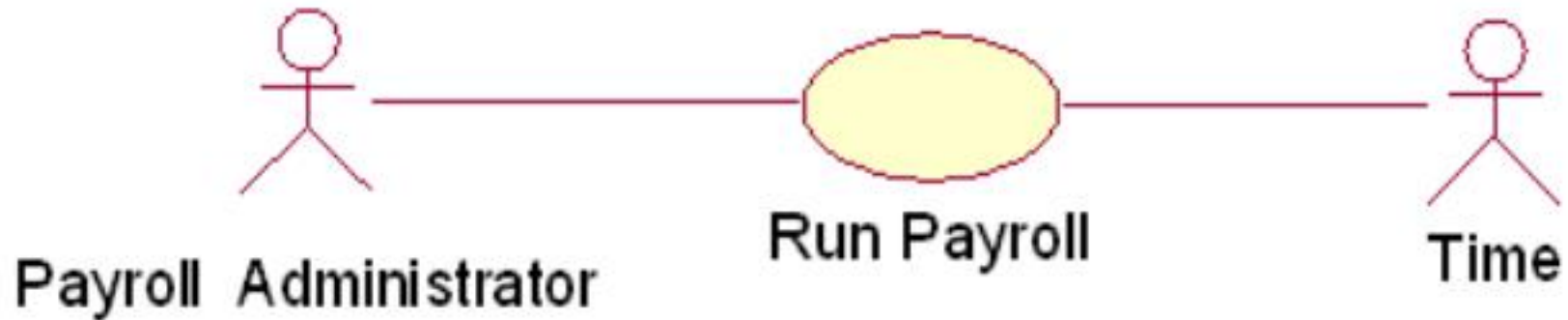


- Time是否是真正的参与者?

- 分析3：系统的目标是什么？谁真正需要这个功能？
  - 可能真正的参与者是工资册管理员(Payroll Administrator)



- 分析4：如何处理需求中的“自动生成一份员工工资册”的问题？
  - 考虑将Time作为次要参与者。



- 如果有必要在用例图中指明哪个用例是依赖于时间的，则这种方法很有效。
- 分析3也是可行的方法，用例图中不指明哪个用例是依赖于时间的，但在用例描述中说明。

# 实例分析—课程注册系统





# Problem Statement

- Wylie学院计划开发一个新的学生注册系统。
- Wylie原来有一个旧的学生注册系统，采用主机-终端型结构。新系统准备采用client-server结构。
- 新的系统允许学生利用局域网上的PC机来注册课程并查看自己的report card。
  - 根据glossary中的定义：report card包含了一个学生在一个学期内的所有课程的所有成绩。
- 新的系统允许教授决定要教哪些课程，并且登记该门课程的成绩。

- 由于经费问题，学院不想立刻更换旧系统的所有部分，计划保留旧系统中的课程目录(course catalog)数据库部分，课程目录数据库中保存了所有的课程信息。
- 课程目录数据库建立在DEC VAX 机器的Ingres 关系数据库上。
- 幸运的是，Wylie学院已购买了一个开放的SQL接口，通过这个SQL接口，可从Unix服务器上直接存取课程目录数据库。
- 旧的学生注册系统的性能相当差，所以要求新的学生注册系统在性能上要有明显改进，在存取课程目录数据库中的数据时要及时。
- 新的学生注册系统将存取课程目录数据库中的课程信息，但不会修改数据库中的课程信息。Registrar通过其它系统来维护课程信息。

- 在每个学期初，学生可以获取这个学期所开设的所有课程的目录，在课程目录中包含每门课的详细信息，如professor, department, prerequisite等。
- 每个学生在一个学期中可选四门课，每个学生还可选两门备选课，以便在主选课不能满足的情况下学生可上备选课。
- 每门课的学生人数最多为10人，最少为3人，如果学生人数少于3人，该门课将被取消。
- 在每个学期，有一个选课期，在这个时间段内，学生可以改变他们的选课计划(Schedule)，注册系统允许学生在这段时间内可以增加或删除所选课程。

- 一旦一个学生的课程注册过程结束，注册系统(registration system)将向计费系统(billing system)发送信息以便学生能交费。如果一门课程已经选满，则必须在向计费系统提交选课计划(schedule)前通知学生。
- 在学期结束的时候，学生可以通过系统查询成绩，由于学生成绩属于敏感信息，因此系统要有安全措施来防止非授权的存取。
- 教授可以存取系统来获取他们所教的课程的信息，可以了解哪些学生选了他们的课，也可以登记该门课程的学生成绩。

说明：

- 两个外部系统：
  - Billing System
  - Course Catalog Database System
- 可以参考RUP 2002的课程注册系统实例。

# Glossary

- 术语表(glossary)定义出所有需要澄清的术语，以便于增进人员之间的交流和减少由于误解所带来的开发风险。
- 在课程注册系统的glossary中，定义了下列术语的含义。

Course

Course Offering

Course Catalog

Faculty

Finance System

Grade

Professor

Report Card

Roster

Student

Schedule

Transcript

- 例： Course和Course Offering 的区别
  - Course: A class offered by the university.
  - Course Offering: A specific delivery of the course for a specific semester – you could run the same course in parallel sessions in the semester. Includes the days of the week and times it is offered.

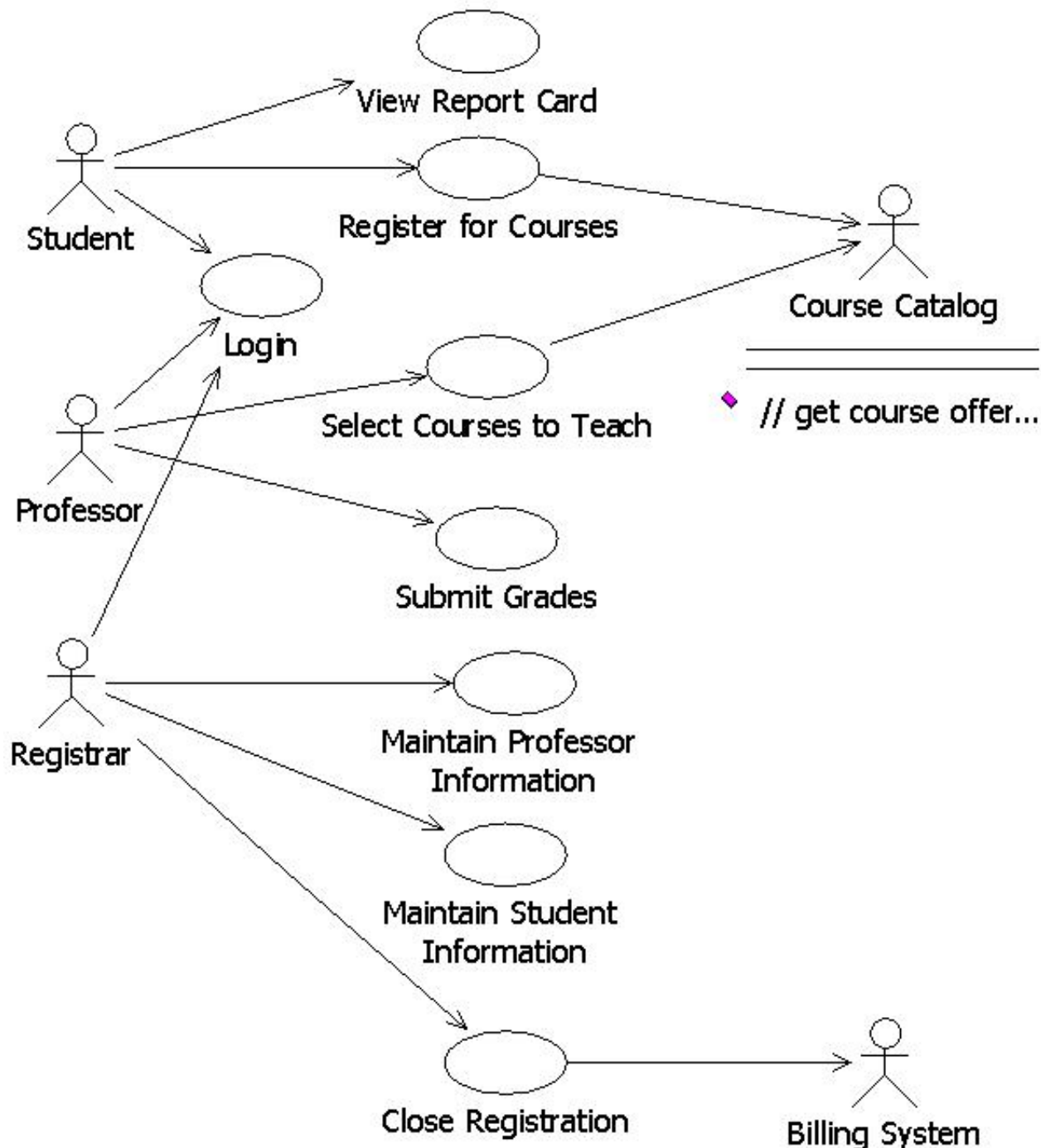
# 分析阶段模型说明

- 文件coursereg(analysis).mdl。
  - 设计阶段的模型文件为coursereg(design).mdl。  
(RUP 2002中的例子的文件名分别是  
coursereg\_analysis.mdl和coursereg\_design.mdl)
- 分析阶段的Use Case图说明



# Use Case图说明

共五个actor，  
八个use case。



说明：

- 以上只是一个可能解决方案，并不是唯一的，只要最后得到的系统是符合需求的即可。
- 某些方案可能比另一些解决方案好，但好的方案来自于experience和hard work。

问题：

新生和老生是否采用不同的actor？

说明：

- 参与者Course Catalog和Billing System为外部系统。
- 在分析阶段，参与者Course Catalog有方法：
  - // get course offerings
- 课程注册系统对Login， Register for Courses用例的分析和设计较完整，包括了Use Case Model， Analysis Model， Design Model， Process Model和Deployment Model；对Close Registration用例只进行了分析，包括了Use Case Model和Analysis Model；对于剩下的用例，只提供了Use Case Model。
- 有八个文件分别对各个Use case进行说明。

## • 注册课程(Register for Courses用例的描述)

### 1. 用例名称：注册课程

#### 1.1 简要描述

这个用例允许学生注册本学期需要学习的课程。在学期开始的课程“add/drop”阶段，学生也可以修改或删除所选择的课程。课程目录系统提供了本学期开设的所有课程列表。

#### 1.2. 事件流程

##### 1.2.1 基本流程

当学生希望注册课程，或想改变他的课程计划(Schedule)时，用例开始执行。

- 1. 系统要求学生选择要执行的操作(创建计划，修改计划，或删除计划)。
- 2. 一旦学生提供了系统要求的信息，以下子流程中的某一个将被执行。

如果学生选择的是“Create a Schedule”，则“创建计划”子流程将被执行。

如果学生选择的是“Update a Schedule”，则“修改计划”子流程将被执行。

如果学生选择的是“Delete a Schedule”，则“删除计划”子流程将被执行。

### 1.2.1.1 创建计划

1. 系统从课程目录系统中检索出有效的课程列表并显示.
2. 学生从有效课程列表中选择4门主选课和2门备选课.
3. 当学生完成选择,系统将为这个学生创建一个“计划”,  
这个计划包含了学生所选的课程.
4. 执行“提交计划”子流程

### 1.2.1.2 修改计划

1. 系统检索并显示学生当前的计划
2. 系统从课程目录系统中检索出有效的课程列表并显示.
3. 学生可以通过增加或删除课程来修改所选课程. 学生从有效课程列表中选择增加的课程, 学生也可以从当前的计划中选择任何想要删除的课程.
4. 当学生完成选择,系统将修改这个学生的“计划”.
5. 执行“提交计划”子流程

### 1.2.1.3 删除计划

1. 系统检索并显示学生当前的计划
2. 系统提示学生确认这次删除
3. 学生确认这次删除
4. 系统删除计划. 如果这个计划中包含“已注册”(enrolled in)的course offering, 则这个学生要从course offering中删除.

### 1.2.1.4 提交计划

1. 对于计划中所选的每门课程, 如果还没标记为“已注册”, 则系统将验证学生满足先修条件, 且课程处于“open”状态, 且计划中没有冲突, 则系统将把学生加到所选的course offering中, 计划中所选的课程标记为“已注册”
2. 计划被保存在系统中.

## 1.2.2 可选流程

### 1.2.2.1 保存计划

在任何情况下, 学生可以选择保存计划而不是提交计划. 在这种情况下, “提交计划”这一步被下面步骤代替:

1. 计划中没有被标记为“已注册”的课程被标记为“选择”(selected).
2. 计划被保存在系统中.

#### 1.2.2.2 先修条件不满足或课程满员或计划冲突

如果在“提交计划”子流程中,系统检测出学生没有满足先修条件,或学生所选的课程已满,或存在计划冲突,则系统显示错误消息.学生可以选择其它课程(用例继续),或保存计划(和“保存计划”子流程一样),或取消本次操作,如果是取消操作,则用例基本流程重新开始.

#### 1.2.2.3 没有找到计划

如果在“修改计划”或“删除计划”子流程中,系统不能检索到学生的计划,则系统显示错误信息.学生确认该错误,用例基本流程重新开始.

#### 1.2.2.4 课程目录系统不可用

如果系统不能和课程目录系统通讯,系统将向学生显示错误信息,学生确认该错误,用例终止.

#### 1.2.2.5 课程注册结束

如果在用例开始的时候,系统检测到本学期的课程注册已结束,系统将向学生显示信息,用例终止.学生在本学期的课程注册结束后就不能再注册课程了.

#### 1.2.2.6 取消删除

如果在“删除计划”子流程中,学生决定不删除计划了,则删除被取消,用例基本流程重新开始.

### 1.3 特殊需求

无

### 1.4 前置条件

开始这个用例之前学生必须已登录到系统

### 1.5 后置条件

如果用例成功结束, 则会创建, 或修改, 或删除学生的计划, 否则系统的状态不变.

### 1.6 扩展点

无



- 练习：在Rational Rose 2003中创建课程注册系统的用例图。

# 用例分析举例1：家庭安全系统（SafeHome）

## 问题陈述：

系统由房主在安装时配置，通过系统控制窗口和键盘功能与房主交互进行。系统可以监控所有连接的传感器。

在安装过程中，每个传感器可以设置编号和类型，系统的启动和关闭必须有主人密码控制；传感器事件发生时，软件激活系统上的警报器，在设定的时间延时后，软件自动拨打设定的电话号码，并提供位置信息和事件性质，电话号码将每隔20秒重拨一次，直至电话接通。

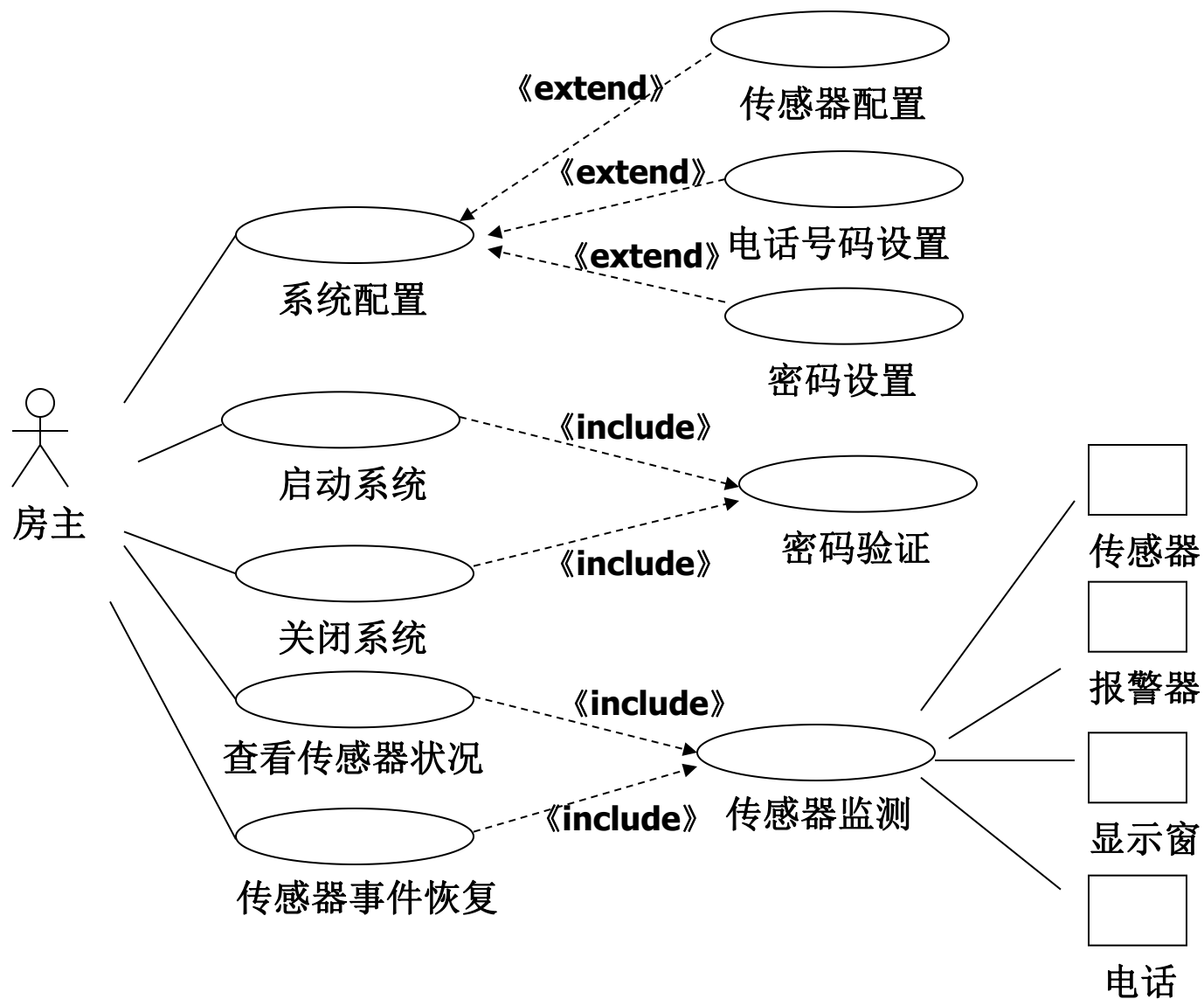
系统运行时，系统控制窗口显示提示信息和系统状态。通过键盘可以控制系统运行。

对问题陈述进行分析，从系统的**参与者**入手，分析**参与者与系统交互的功能**。

系统外部的参与者，包括人员、设备和其他系统  
在此问题中包括：

房主	启动和使用系统的主动参与者
传感器	系统监控的主要参与者
警报器	系统操纵的参与者
电话	同上
控制窗口	虽然是标准设备，但有特殊定义

# Safehome系统用况图



# 用例分析举例2：道路坑洼修复系统

## 问题陈述：

市民可以登录网站报告坑洼的位置和严重程度。每个被报告的坑洼，将登记到“市政部门的修复系统”中，并被赋予一个标识号，按街道地址、大小（1-10）、位置（路中或路边等）、城区（地址所在城区）和修复理由的优先级（坑洼严重程度）存储起来。

市政人员确认坑洼情况属实后，填写派工单并派工修复坑洼。工单内容包括：位置和大小、修理队的人数、使用的修理装备、修复所用的时间、坑洼状况（正在工作、已被修理、临时修理、未修理）、使用填料的数量和修理的开销（由使用的时间、人数、使用的材料的装备计算得到）。

坑洼经修复后，可输出坑洼的损害报告，包括市民的姓名、地址、电话号码、损害类型和修复总费用。

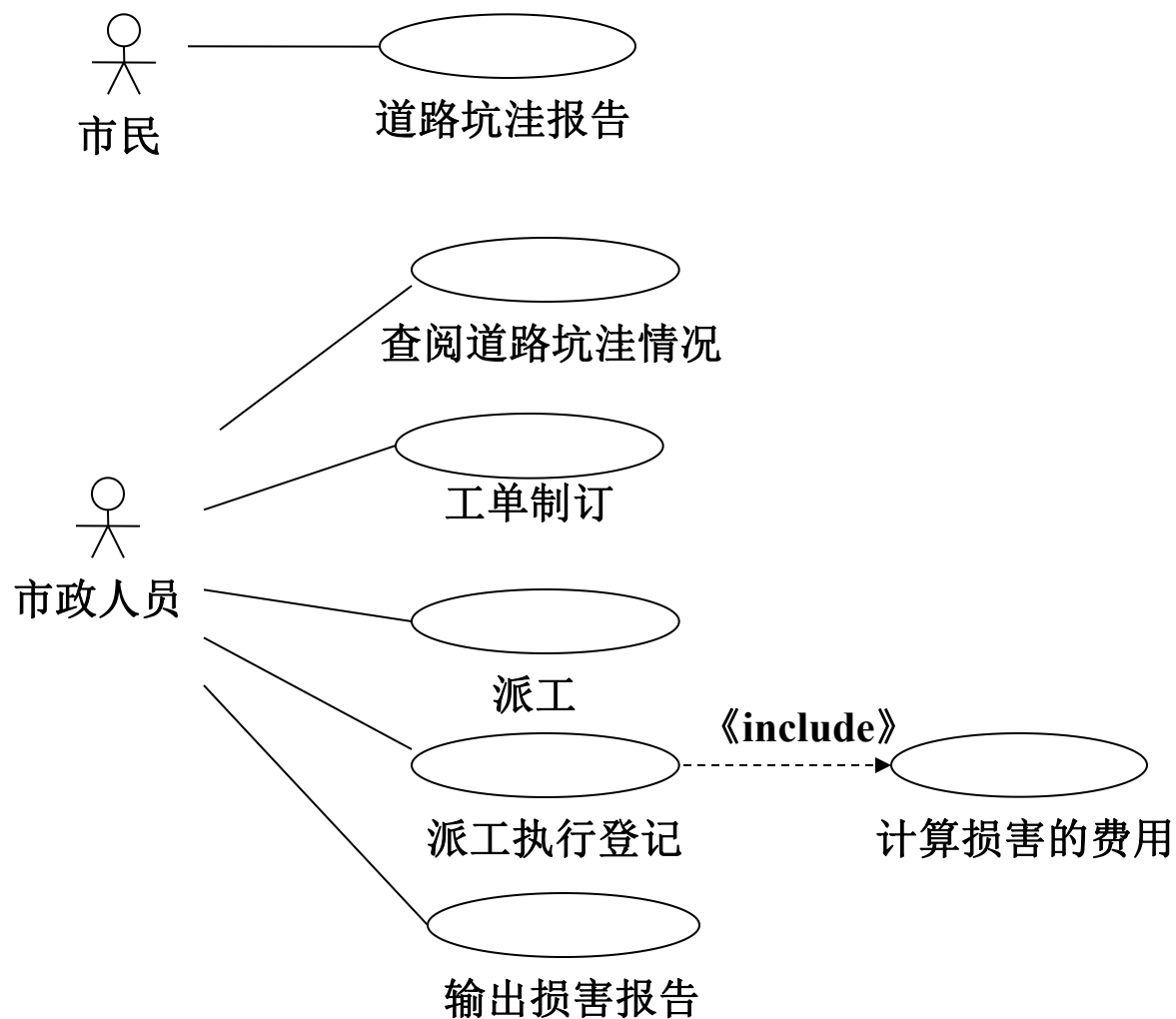
对问题陈述进行分析，从系统的**参与者**入手，分析**参与者与系统交互的功能**。

在此问题中，系统外部的参与者包括：

市民----- 登记报告坑洼

市政工作人员-----填写工单，派工，并汇总  
和输出修复情况及费用。

# 道路坑洼修复系统用例图：



# 用况分析举例3：电话客户信息系统

## 问题陈述：

是电话公司开发管理电话客户信息的交互式网络系统，功能如下：

1. 浏览电话客户信息：任何使用Internet的网络用户，都可以浏览电话客户信息（姓名、住址、电话号码等）。
2. 登录：电话公司授予每个电话客户一个帐号。授权了的电话客户，可以使用系统提供的页面来设置个人密码，并使用帐号和密码向系统注册。
3. 修改个人信息：电话客户向系统注册后，可以发送电子邮件或者使用系统提供的页面，对个人信息进行修改。
4. 删除电话客户信息：只有公司管理人员才能删除不再接受公司服务的电话客户信息。



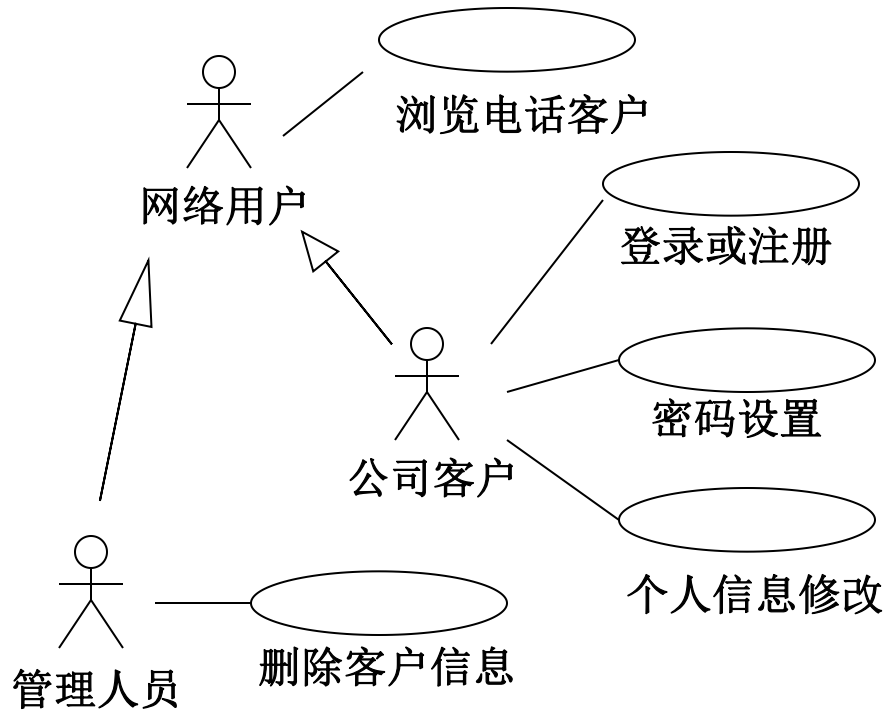
对问题陈述进行分析，从系统的**参与者**入手，分析**参与者与系统交互的功能**。

在此问题中，系统外部的参与者包括：

所有使用**Internet**的网络用户----- 查询电话客户信息  
电话客户-----用指定帐户密码向系统注册，并可修改个人信息。

公司管理人员-----删除电话客户信息

# 电话客户信息系统用例图：



# 练习题

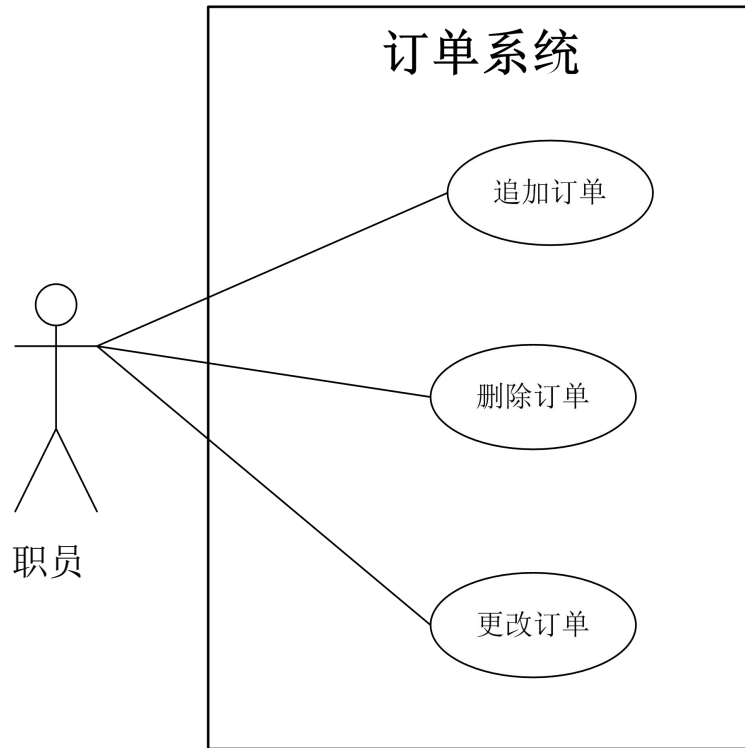
## ◆ 试画用例图描述以下内容：

某销售公司准备开发一个订单管理系统。

该公司的职员可以利用该订单系统进行订单的追加，删除和修改。

请试着用用例图来描述该系统。

# 练习题参考答案



# 总结

## ◆ 用例图

- 用例图是在系统开发的初期阶段，描述站在用户的角度上看系统提供给用户的可使用的功能。
- 参与者是与系统进行交互的角色，不是某个具体的个人。可以是人，硬件，或者外部已经存在的某个系统。共同点是要与系统有直接的信息交互。
- 用例图和用例描述合称为用例模型。