

# 用例图

任守纲

PEREC

## 案例描述：零件销售系统（1）

- ❖ R公司是一家有悠久历史的零件代理商，为基础工业的各个行业（制造业、建筑业...）提供零件，它想要开发一个在线销售系统。
- ❖ R公司代理的零件的种类非常多，有螺母、螺钉、铆钉、销、垫圈等，所以每年R公司都要出版一套零件目录，供顾客查询。
- ❖ 以前的业务流程是：顾客在目录上查到所需要的零件后，打来电话或发来传真，请求购买，销售员根据顾客的购买数量，与供应商查询后，告知顾客价格。顾客通过电话或传真告知销售员进行订购。

- ❖ 在基于Web的新系统中，顾客可以通过Internet进行购买。
- ❖ 顾客先预付一定金额存入内部账户中成为会员，然后才能购买零件。顾客可以根据自己所知道的零件的形状，大小、零件编号等指标，搜索出所需要的零件。结帐使用内部帐户支付。系统根据会员提供的送货地址和订购数量，从库存中搜索出离送货地址最近的供应商，通知供应商发货。
- ❖ 内部工作人员不定期地根据供应商方面的价格变动，对某些零件的销售价格进行更新。每个星期，各个供应商会把记录自己最新库存情况的Excel文件寄来，系统根据这些文件更新库存信息。
- ❖ 因简化的需要，以下因素略去不考虑：折扣，延迟交货...

# 需求——建造“正确”的系统



# 怎么样才是好的建模人员

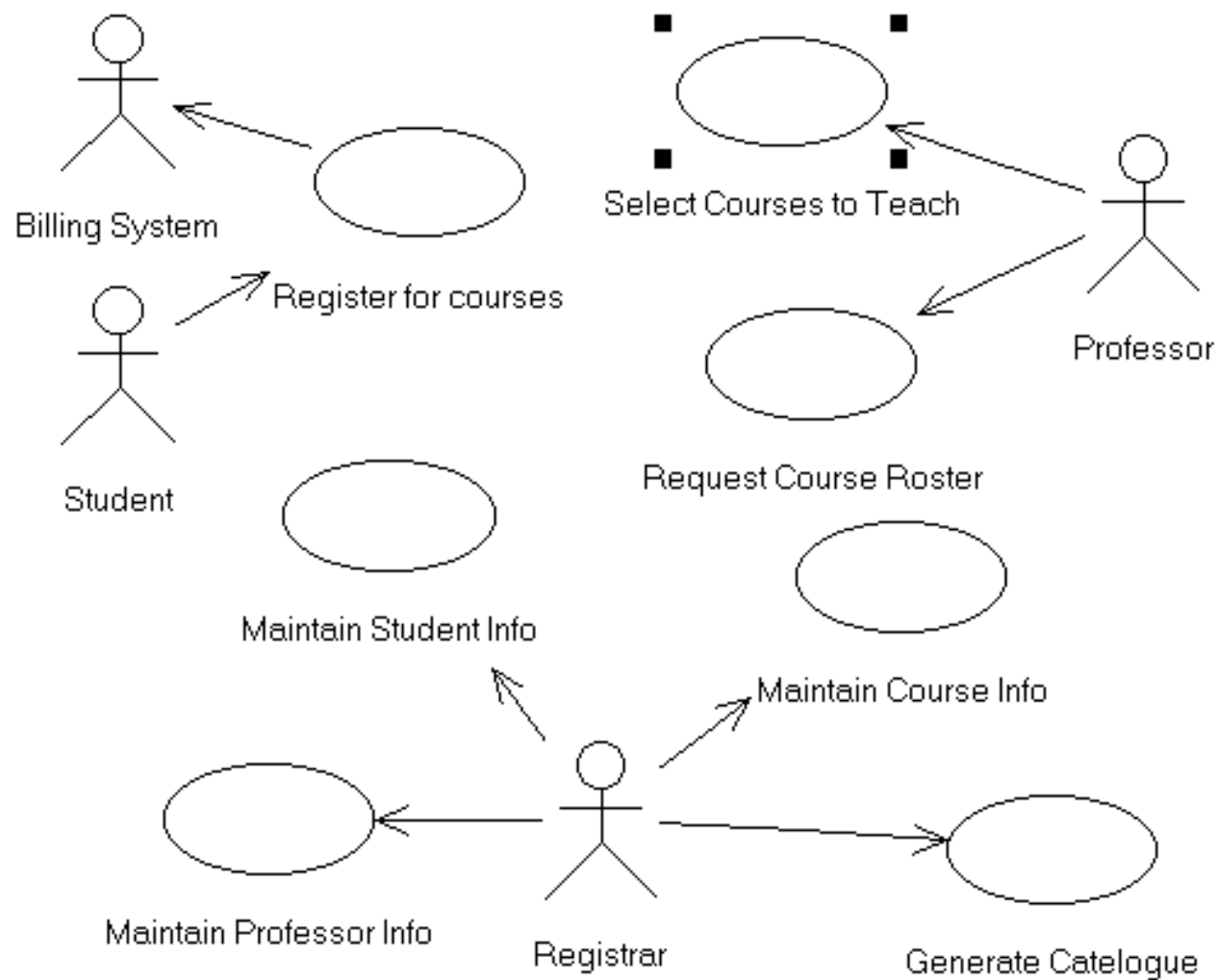
№ 5

- ⊕ 最好的建模人员——领域专家
- ⊕ 设计师要有Coding 经验，但最好不要自己Coding自己的设计，这样才能培养Coding前做设计的能力
- ⊕ 领域知识+建模技术

# 一 用例图定义

№ 6

- ⊕ 软件需求到最终实现的第一步
- ⊕ 系统分析设计的起点
- ⊕ 确定系统的业务需求
- ⊕ 确定系统的用户



课件登记系统的用例图

## 二 用例图的内容

№ 8

- ⊕ 参与者
- ⊕ 用例
- ⊕ 关系
- ⊕ 子系统



# 2.1 用例

№ 9

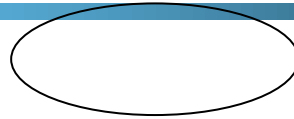
⊕ 用例——用户期望系统具有的功能

⊕ Use Case提供了一种手段

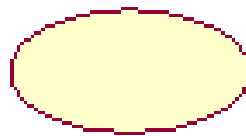
- 捕获系统需求
- 专业人士和最终用户间的连接
- 测试系统

- 表示法

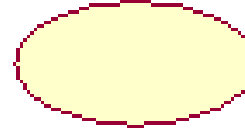
№ 10



Use case 名称



管理学生资料



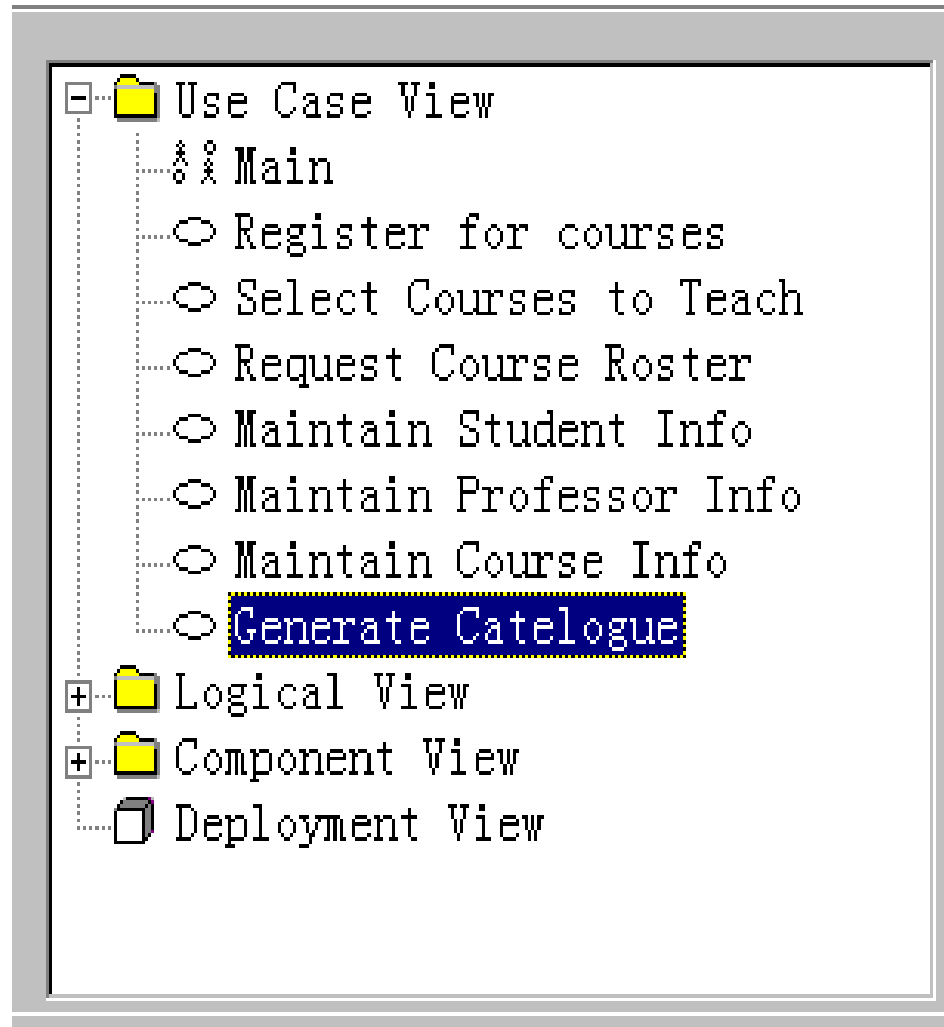
管理信箱

命名： 可带有数字、字母和除保留符（冒号）以外的任何标点组成的字符串

尽量用动词短语

# 浏览窗口中的Use Cases

№ 11



## 2.2 参与者

№ 12

⊕ 参与者——表示使用系统的对象

⊕ Actor是一些人或者另一个系统:

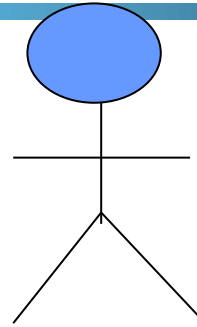
- 可以激活系统交互信息
- 可以对系统进行输入
- 可以从系统被动的接受信息

## ⊕ 通过调查发现Actor

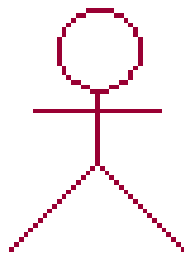
- 直接使用系统的人
- 系统的维护人员
- 系统使用的外设
- 需要与此系统想连的其它系统

## ⊕ 表示法

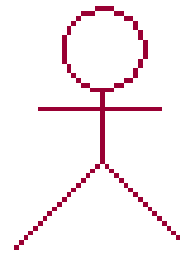
№ 14



参与者



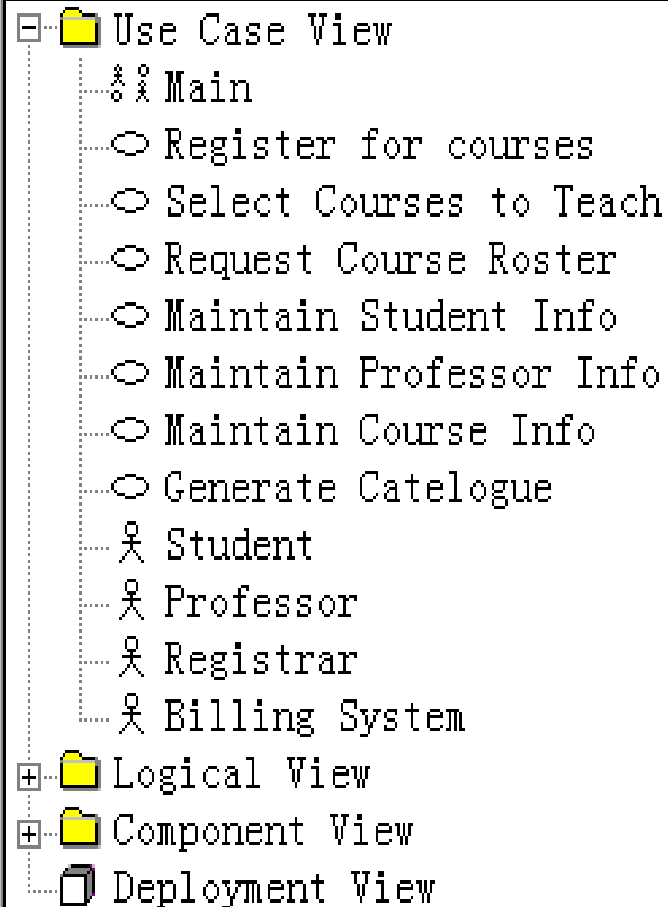
学生



老师

# 在浏览窗口中的Actor

№ 15

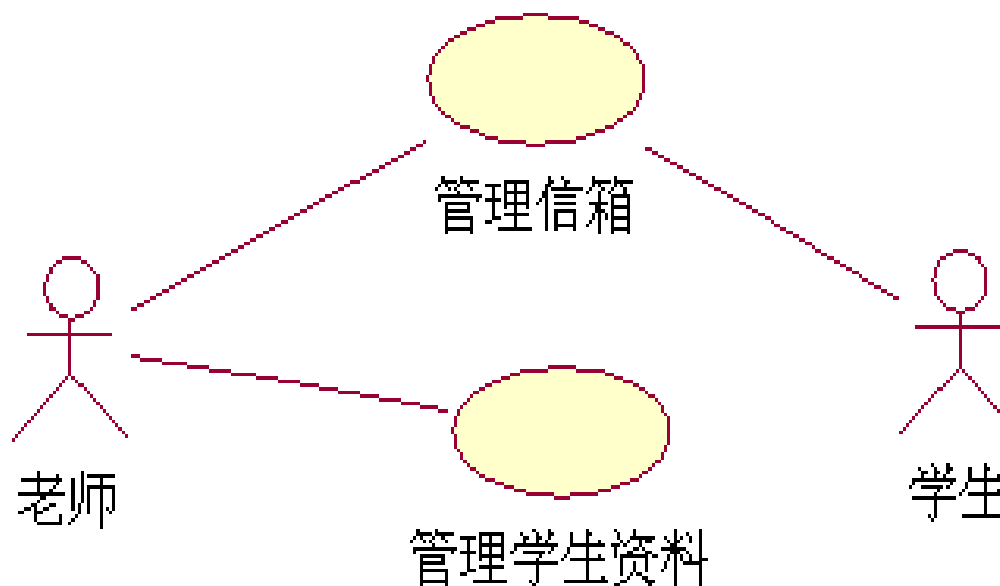


## 2.3 关系

№ 16

⊕ 关系——用来连接用例和参与者

⊕ 表示法

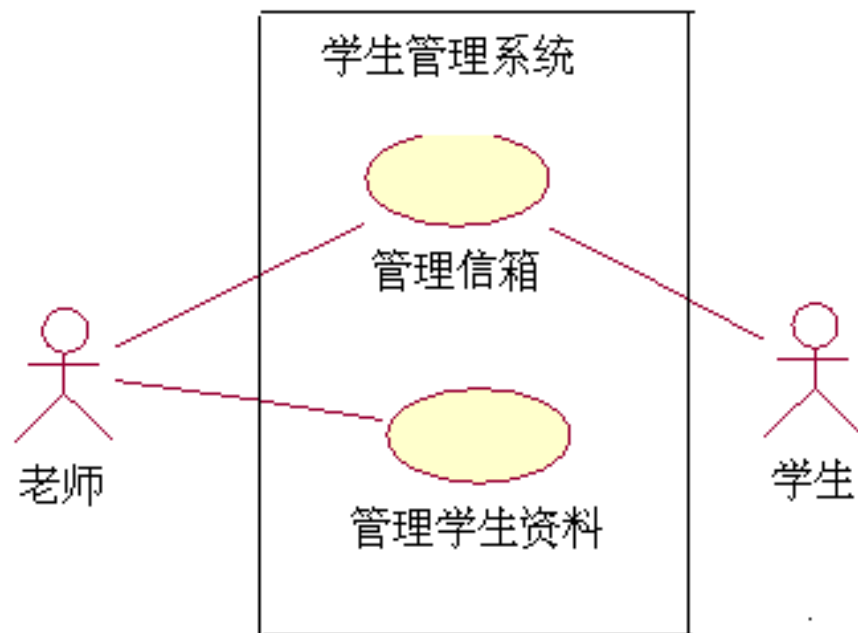




## 2.4 子系统

№ 17

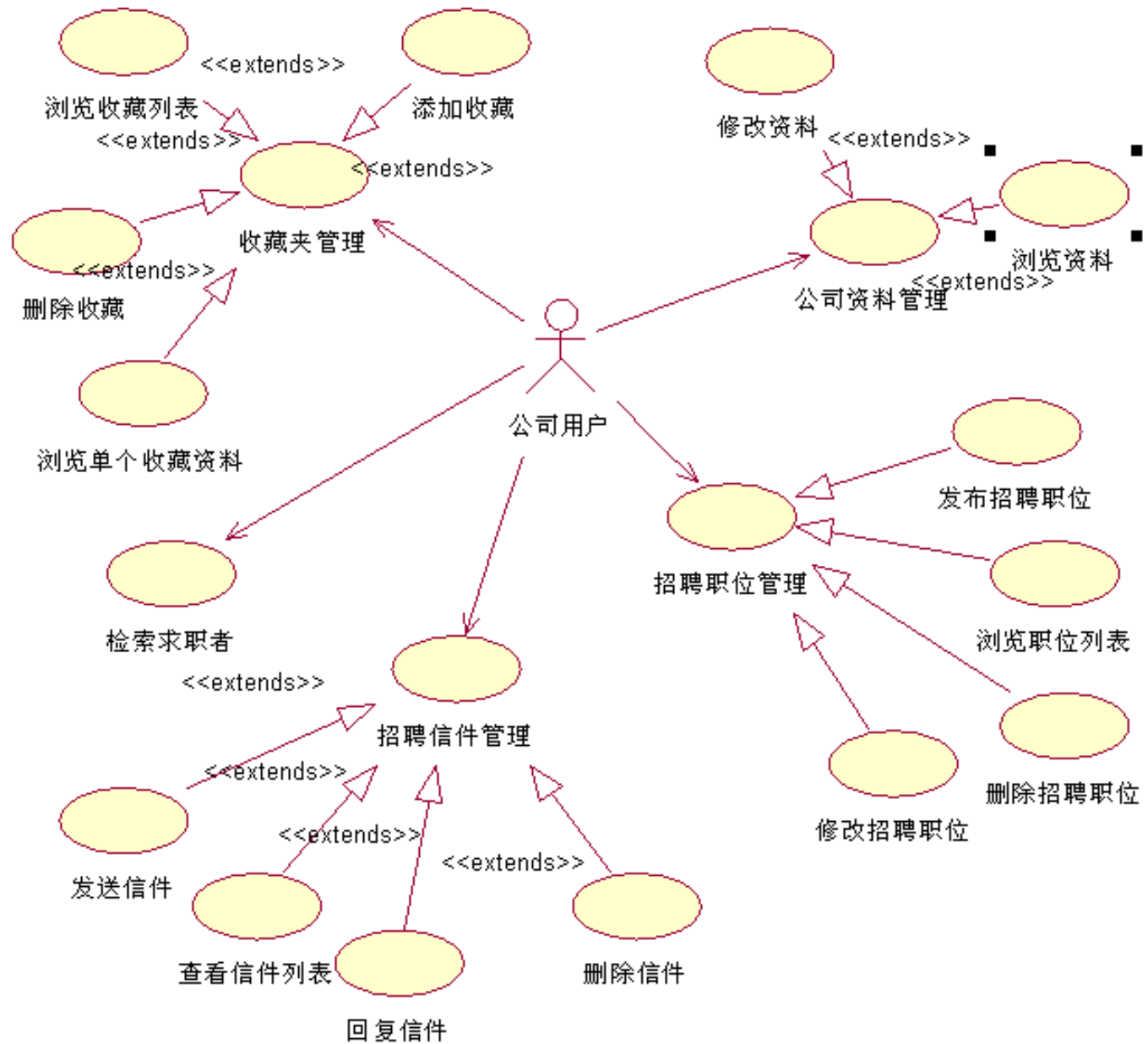
⊕ 子系统——把一个复杂的大型系统分成若干个小子系统



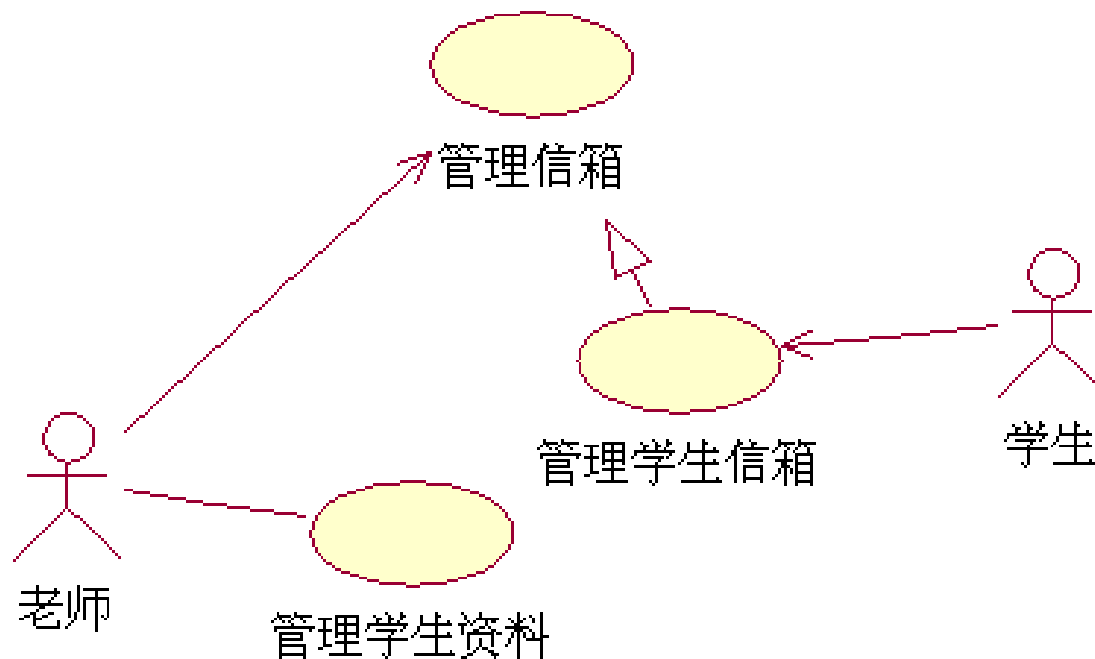
# 三 泛化技术

№ 18

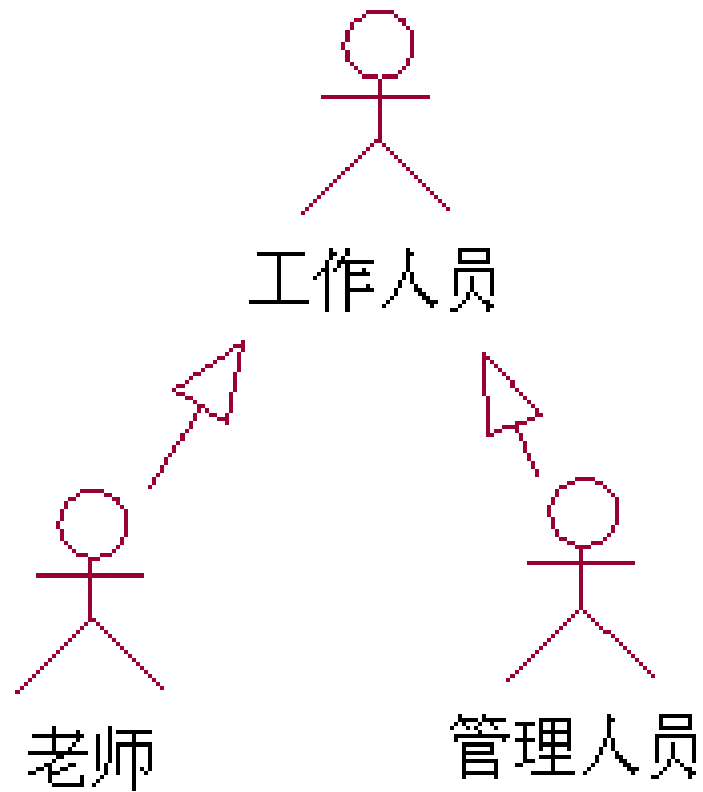
- ⊕ 泛化——表示UML中项目的继承关系
- ⊕ 包括用例泛化和参与者泛化

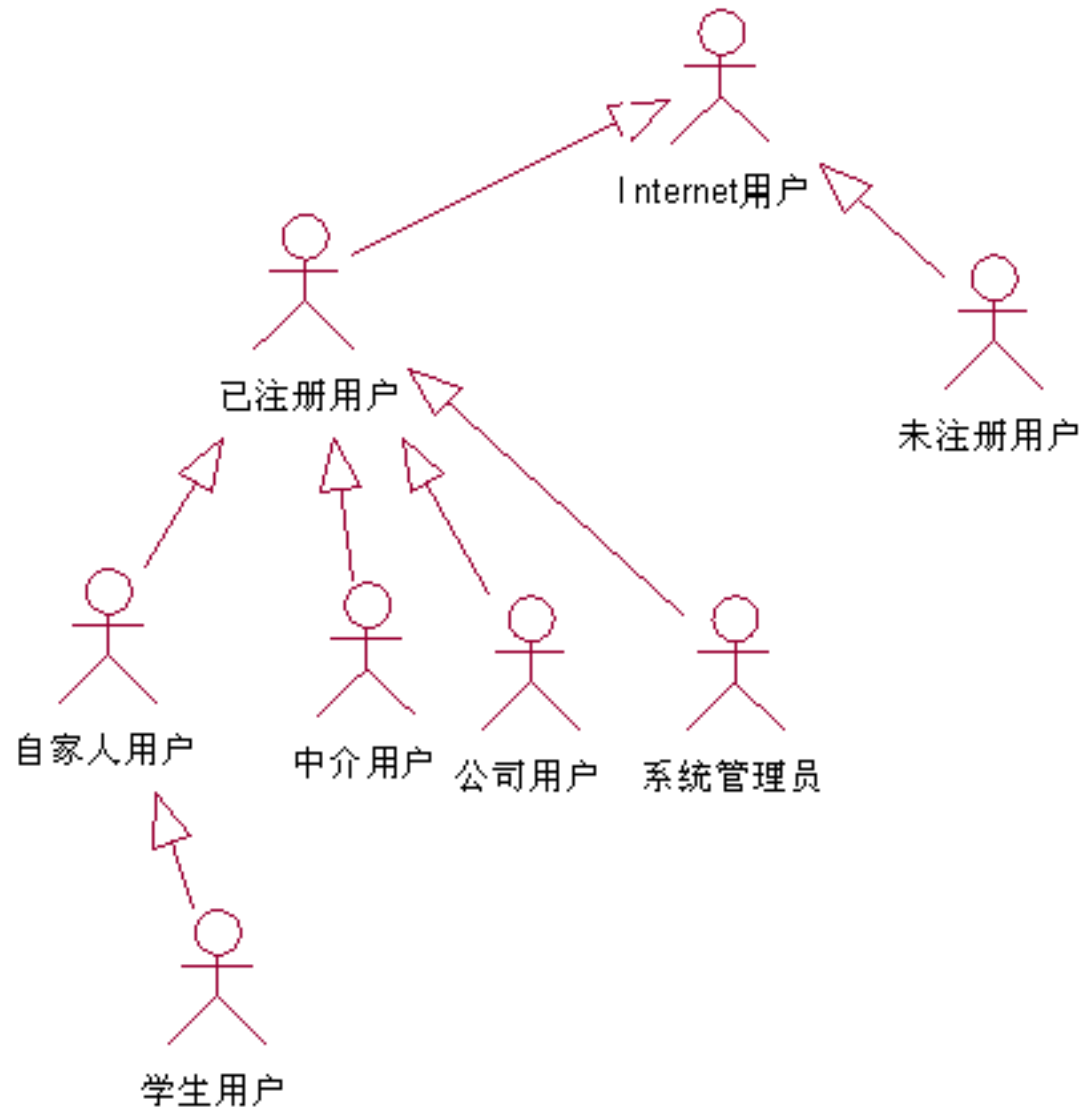


- ❖ 参与者使用了泛化的子用例，就不应该再使用泛化的父用例



## ⊕ 参与者泛化

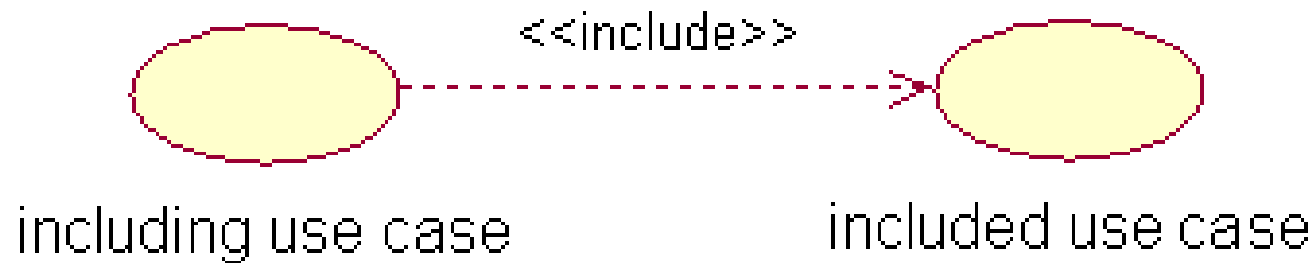




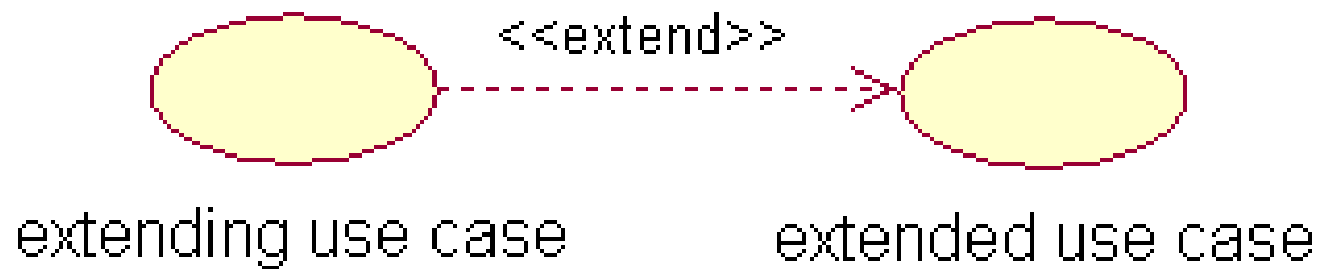
## 四 包含和扩展关系

№ 23

### ⊕ 包含关系



## ⊕ 扩展关系

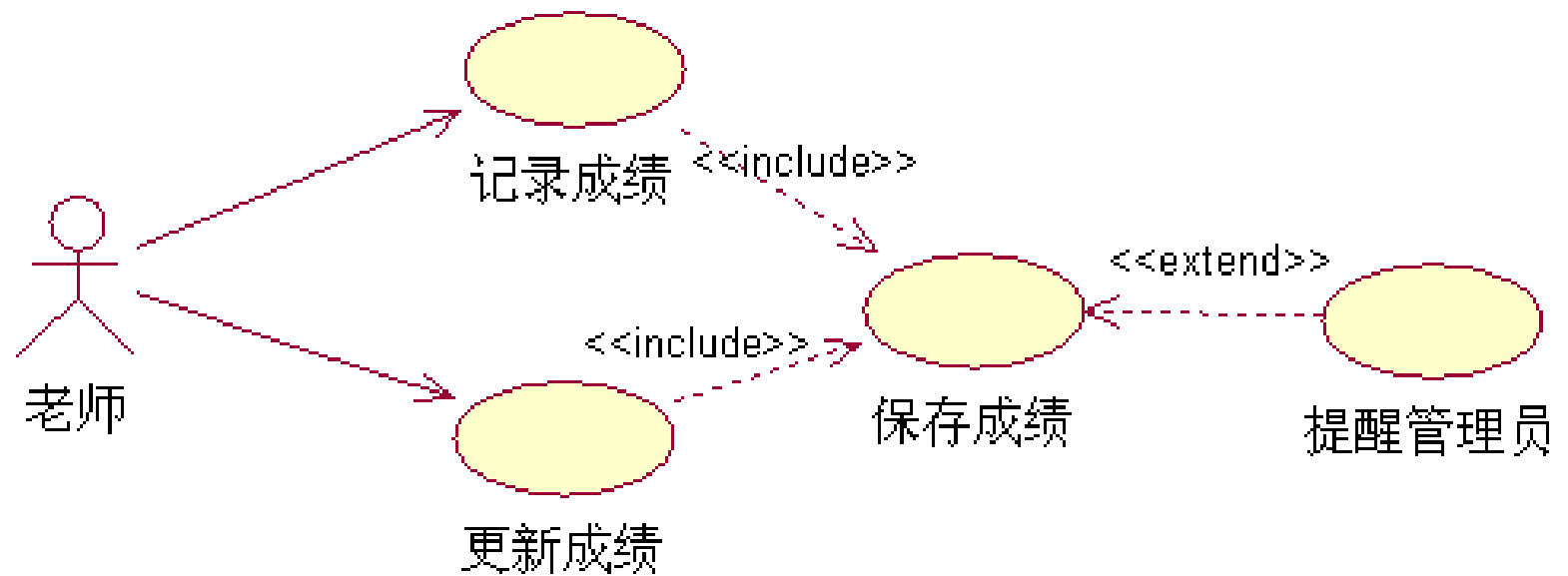




## ⊕ 区别

在包含关系中，用例**必须**包含被包含用例

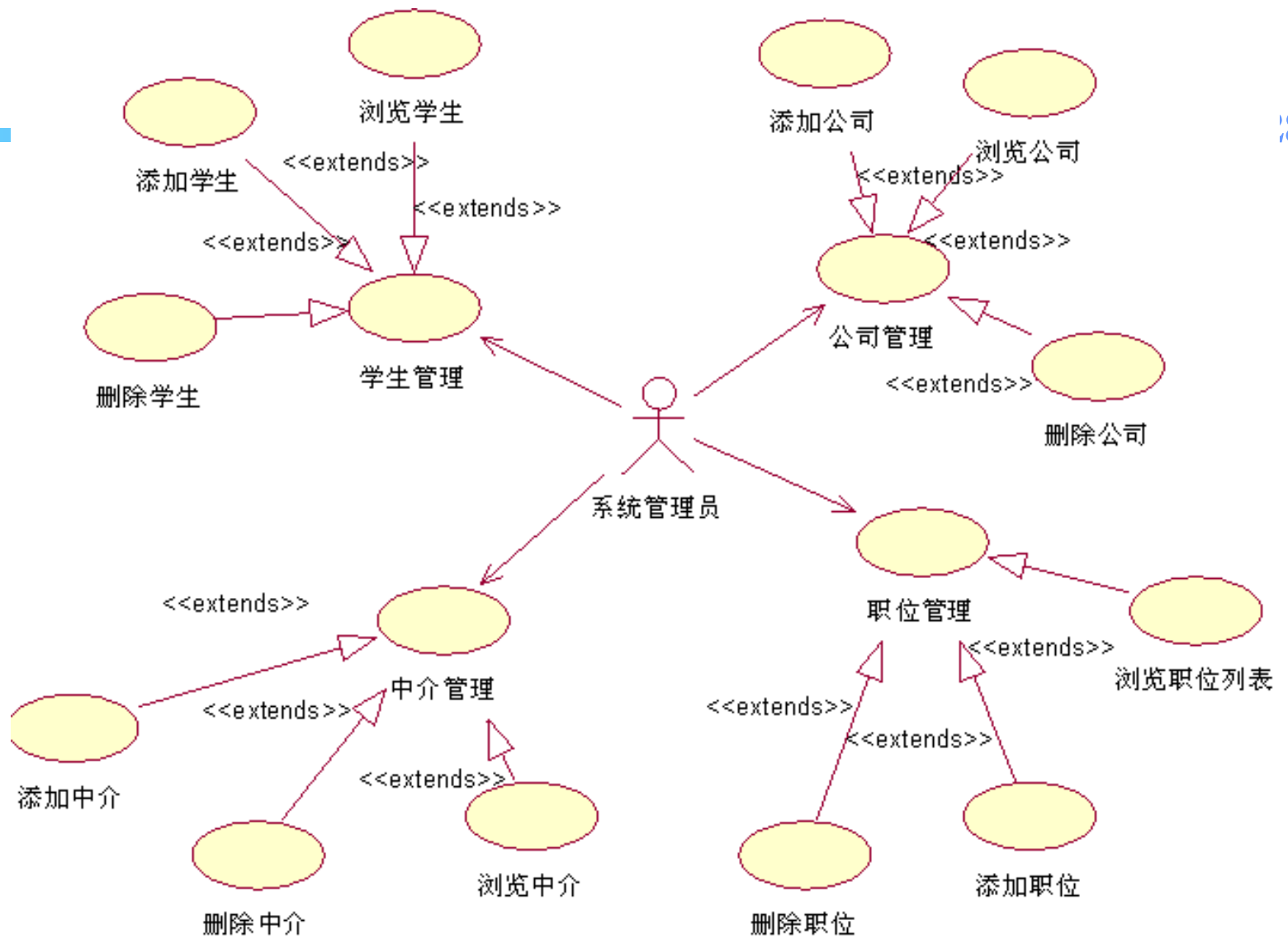
在扩展关系中，用例有**选择**使用被扩展用例的权利

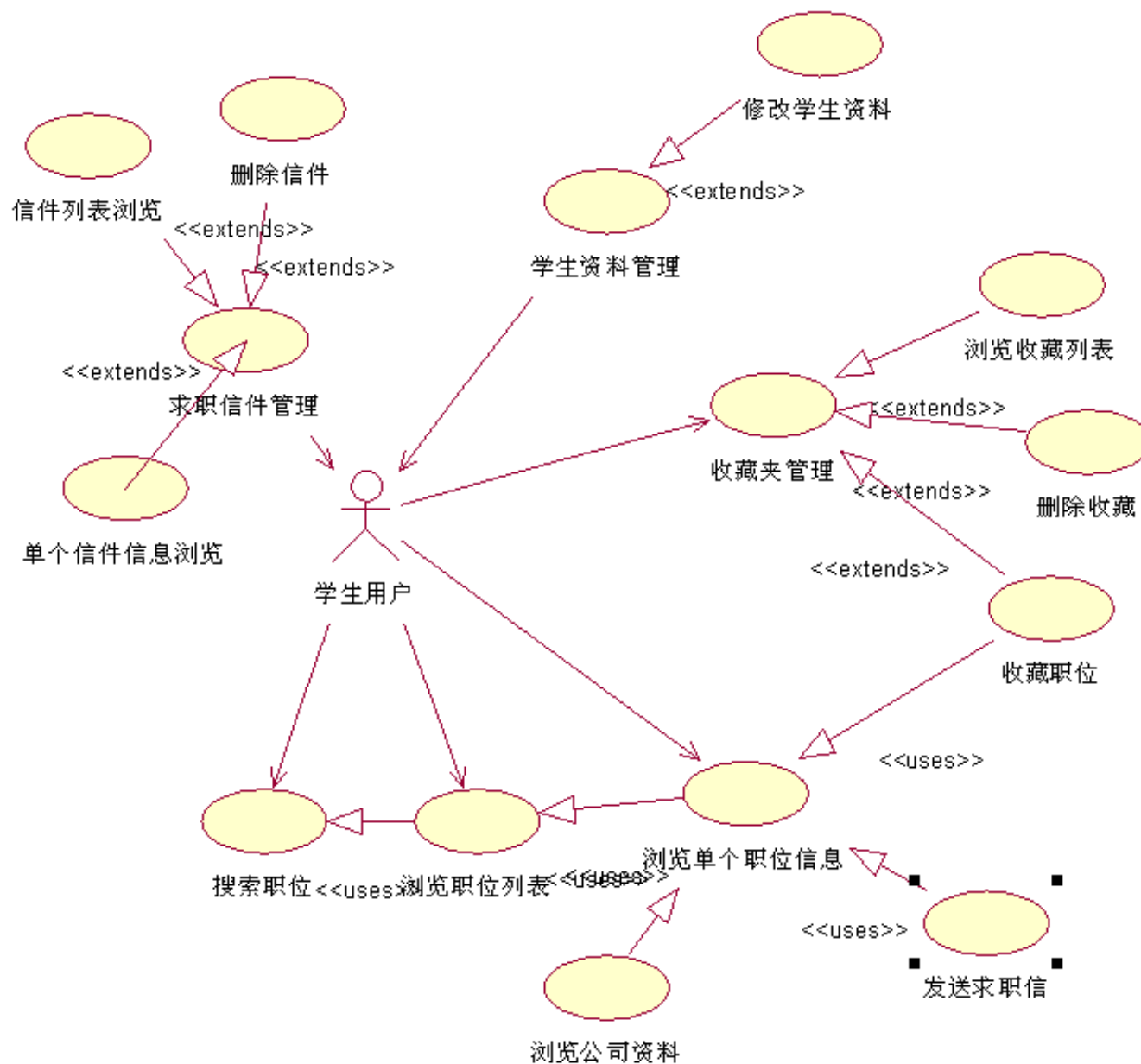


# 五 建立用例图步骤

№ 27

- ⊕ 找出系统参与者和用例
- ⊕ 区分用例的优先次序
- ⊕ 细化每个用例
- ⊕ 构建用例模型
- ⊕ 建立用户界面的模型












## 案例描述：零件销售系统（1）

- ❖ R公司是一家有悠久历史的零件代理商，为基础工业的各个行业（制造业、建筑业...）提供零件，它想要开发一个在线销售系统。
- ❖ R公司代理的零件的种类非常多，有螺母、螺钉、铆钉、销、垫圈等，所以每年R公司都要出版一套零件目录，供顾客查询。
- ❖ 以前的业务流程是：顾客在目录上查到所需要的零件后，打来电话或发来传真，请求购买，销售员根据顾客的购买数量，与供应商查询后，告知顾客价格。顾客通过电话或传真告知销售员进行订购。

- ❖ 在基于Web的新系统中，顾客可以通过Internet进行购买。
- ❖ 顾客先预付一定金额存入内部账户中成为会员，然后才能购买零件。顾客可以根据自己所知道的零件的形状，大小、零件编号等指标，搜索出所需要的零件。结帐使用内部帐户支付。系统根据会员提供的送货地址和订购数量，从库存中搜索出离送货地址最近的供应商，通知供应商发货。
- ❖ 内部工作人员不定期地根据供应商方面的价格变动，对某些零件的销售价格进行更新。每个星期，各个供应商会把记录自己最新库存情况的Excel文件寄来，系统根据这些文件更新库存信息。
- ❖ 因简化的需要，以下因素略去不考虑：折扣，延迟交货...

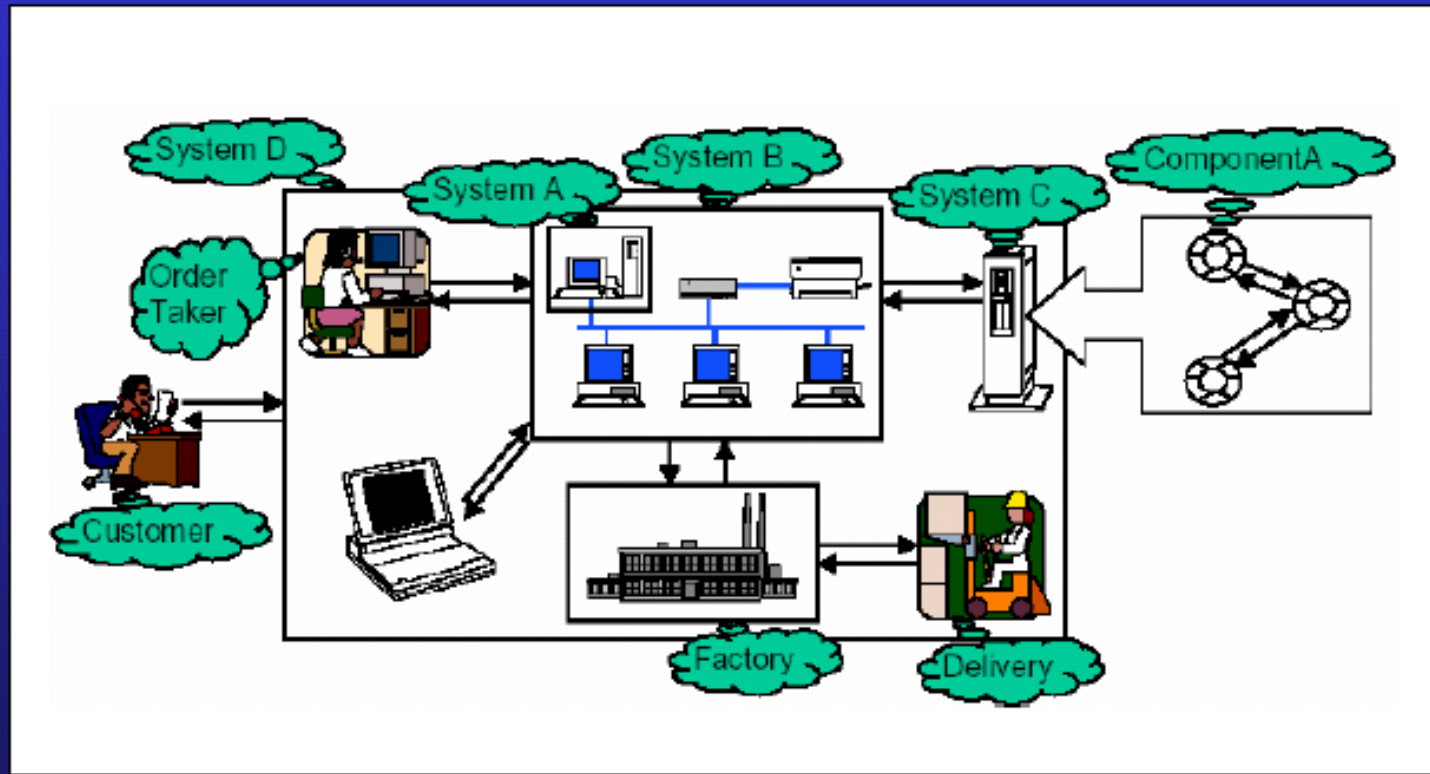
## 步骤

-  识别系统边界和参与者 
-  列出事件
-  识别用例
-  书写用例文档
-  识别用例的关系
-  对用例进行优先级排序



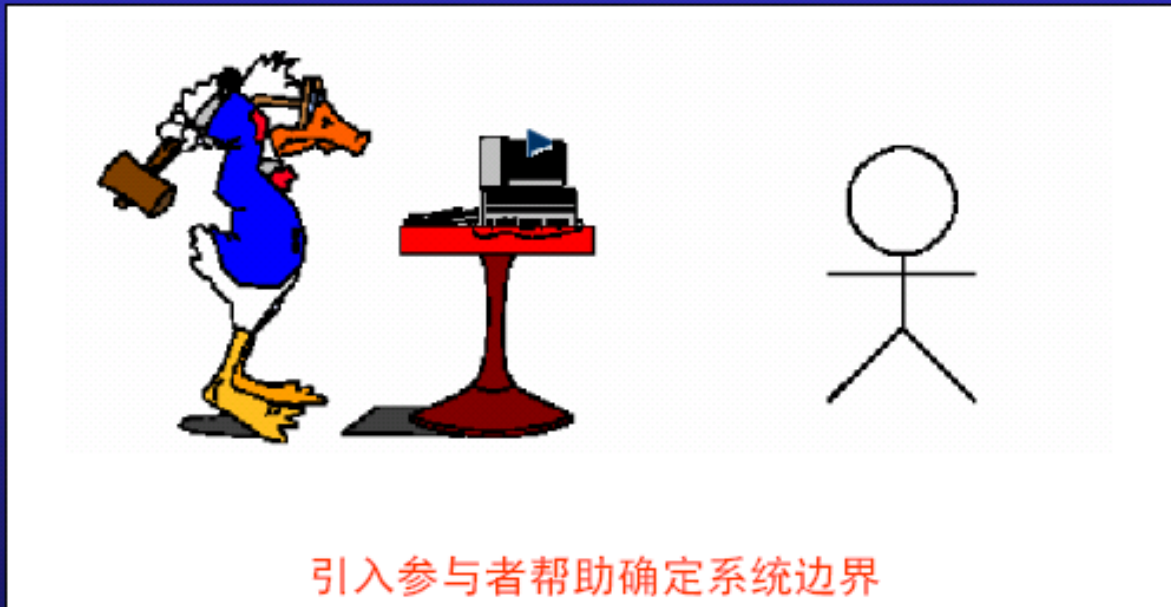
# 识别系统边界和参与者

## —— 系统边界



## ——参与者（Actor）

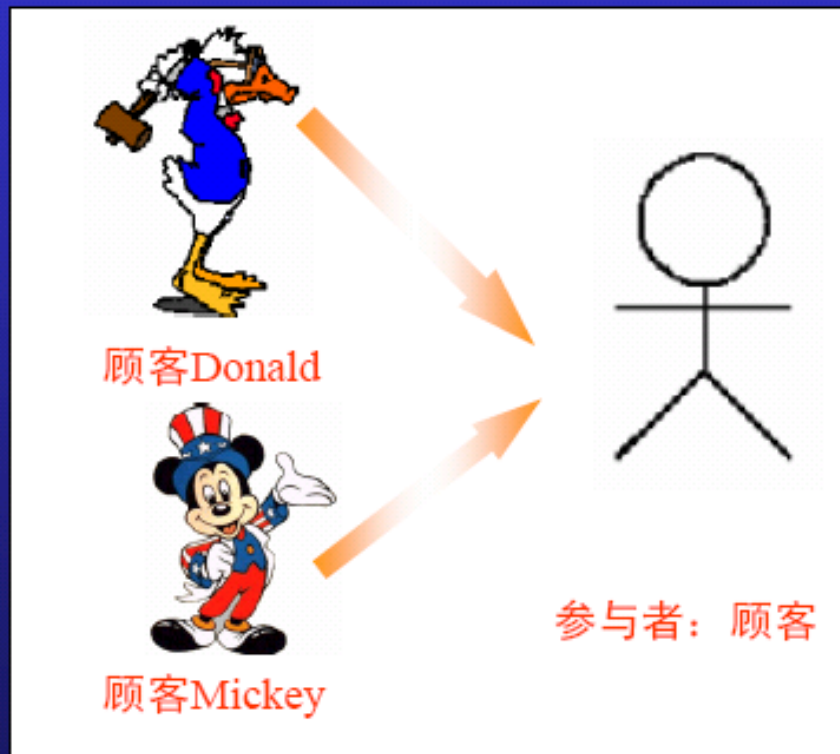
- 在系统之外，透过系统边界与系统进行有意义交互的任何事物



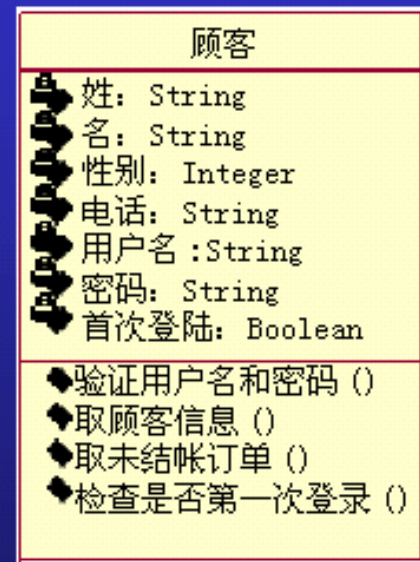
## ——参与者要点

- 参与者代表在系统边界之外的真实事物，并不是系统的成分
- 参与者透过系统边界直接与系统交互，参与者的确定代表着系统边界的确定
- 交互是有意义的
- 参与者可以是任何事物

## ——参与者在系统外面

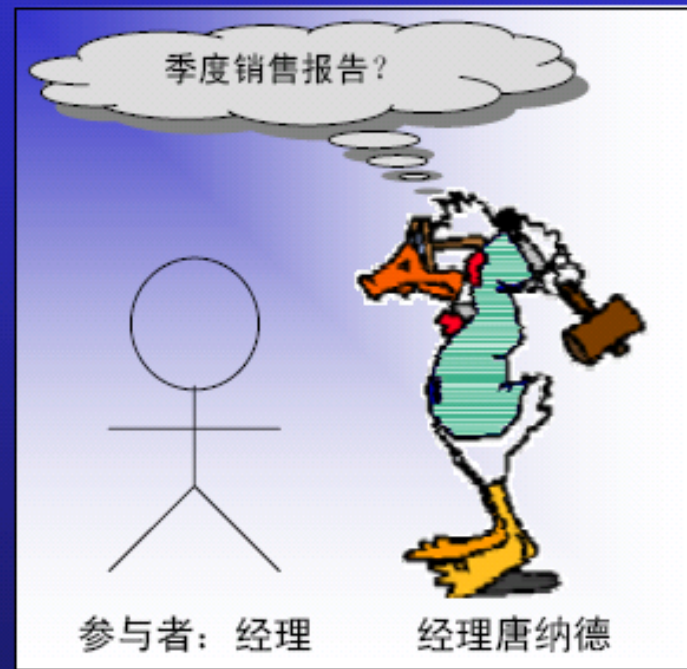


边界



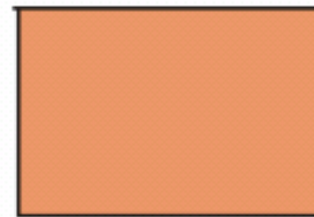
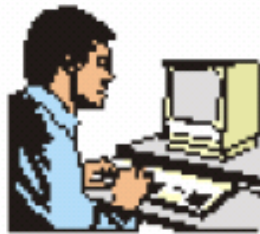
类：顾客

## ——一个人可以担任多个参与者



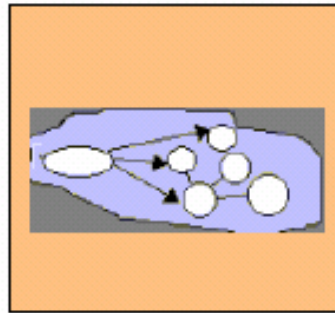
参与者不同，所需的领域知识和交互技巧不同

## ——直接与系统交互





## ——有意义的交互

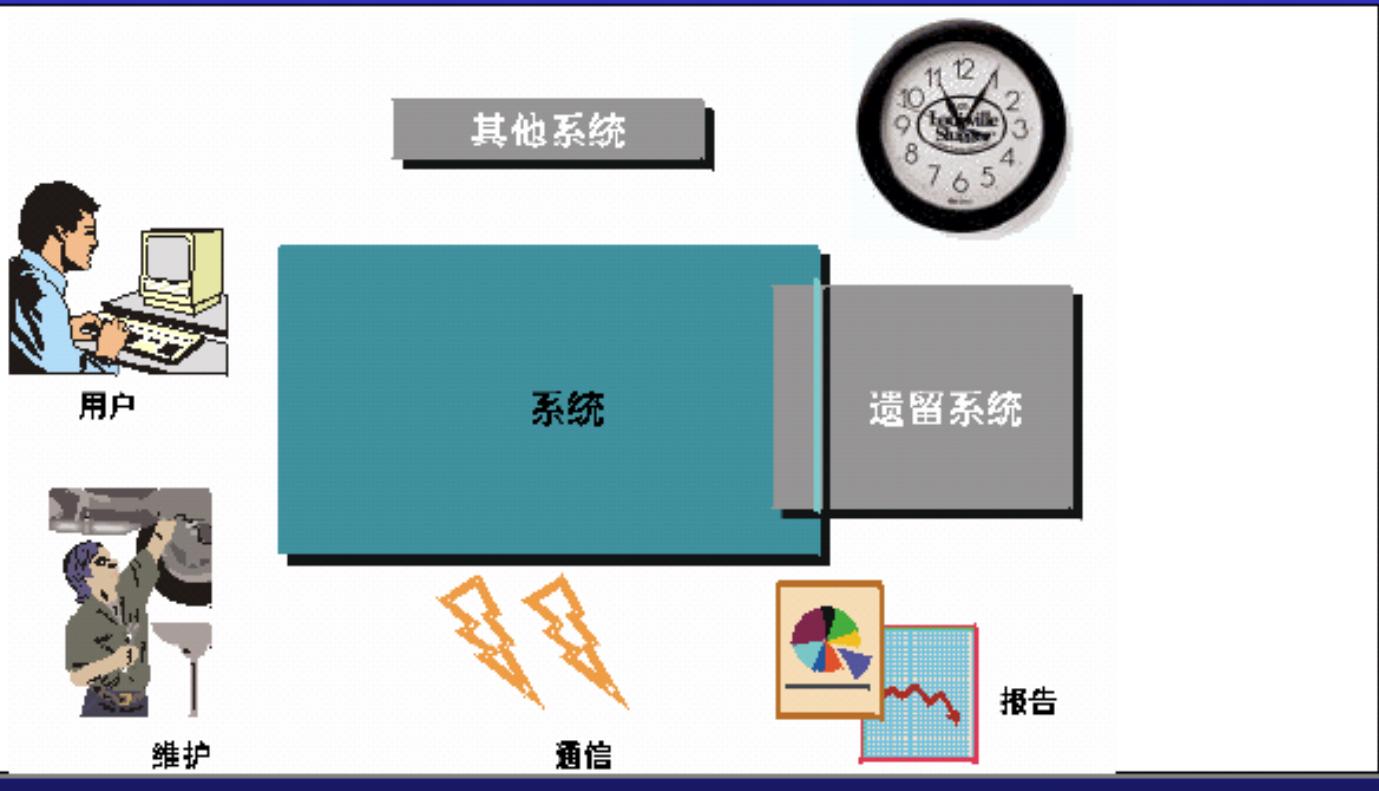


不属于系统责任

不属于系统责任

不属于系统责任

# ——任何事物





# 讨论——识别系统边界

№ 41

❖ 某企业要求开发一个企业管理系统，并与原来已有的财会管理系统相连接

❖ 某企业要求开发一个企业管理系统，并把原来已有的财会管理系统加以改造，成为企业管理系统的一部分

# 识别参与者

№ 42

- 仪器分析系统
  - 一系列样品溶液在分析仪器上进行测试，实验员把每个样品的数据输入计算机，最后，系统对数据进行分析，并给出分析结果。

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。

寻呼台系统。用户如果预定了天气预报，系统每天定时给他发天气消息；如果当天气温高于35度，还要提醒用户注意防暑。

这个叙述里，谁是寻呼台系统的Actor？用户？气温？时间？

商品销售系统。顾客通过网络下单之后，系统计算出总计金额，税金，运费，并将数目传递给一个外挂的会计系统，该系统是另外购买的。

有几个Actor？

# 总结：识别参与者思路

№ 46

- ❖ 谁使用系统的主要功能？
- ❖ 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？
- ❖ 谁负责维护、管理并保持系统正常运行？
- ❖ 系统需要应付（处理）哪些硬设备？
- ❖ 系统需要和哪些外部系统交互？
- ❖ 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

# 零件销售系统

№ 47

- ❖ 谁使用系统的主要功能? — 潜在会员, 会员
- ❖ 谁改变系统的数据? — 会员, 货管员, 经理
- ❖ 谁从系统获得信息 — 潜在会员, 会员, 经理, 货管员
- ❖ 谁需要系统的支持以完成日常工作任务? — 经理, 货管员
- ❖ 谁负责维护、管理并保持系统正常运行? — 系统管理员
- ❖ 系统需要应付(处理)哪些硬设备? — 没有特殊硬设备
- ❖ 系统需要和哪些外部系统交互? — 可能与供应商的系统交互
- ❖ 谁(或什么)对系统运行产生的结果(值)感兴趣? — 会员, 经理
- ❖ 时间、气温等内部外部条件 — 时间

# 零件销售系统一候选参与者

No 48



潜在会员



会员



货管员



经理



供应商系统

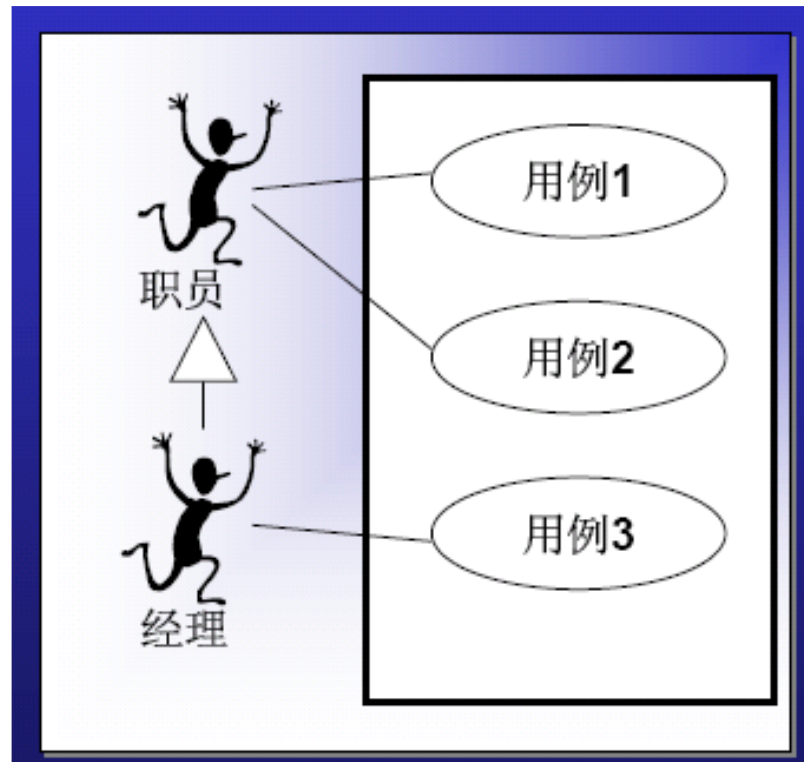


时间



# 参与者泛化关系

№ 49



识别系统边界和Actor（参与者）

列出事件



识别用例

书写用例文档

识别用例的关系

对用例进行优先级排序

## ❖ 系统必须响应的外部事件和内部事件

- ❖ 外部事件：来自系统外部

- ❖ 顾客+下定单

- ❖ 内部事件：来自系统内部

- ❖ 和时间有关：每天晚上检查账户

## ——头脑风暴法

❖ “主语+动词(+宾语)”



❖ 主语：Actor 的候选，例如：乘客，顾客，店员。

❖ 动词：表示行为。例如：买，发送，修改...

❖ 宾语：动词所代表行为的目标

# 零件销售系统的事件

№ 53

Actor	事件	业务目的	Actor	事件	业务目的
潜在会员	检索零件		会员	修改个人资料	
潜在会员	输入查询条件		经理	开放账户	
潜在会员	提交查询条件		经理	打印销售情况报表	
潜在会员	查看零件的特征		经理	打印会员情况报表	
潜在会员	查看零件的价格		经理	检索会员	
会员	决定购买某零件的数量		货管员	更新库存	
会员	输入送货地址		货管员	更新价格	
会员	结账		供应商	发货	
会员	改变购买数量		会员	收货	
会员	从订单中删除某一项		会员	付款	
会员	取消订单		时间	每天晚上检查账户	
潜在会员	注册				
潜在会员	提供会员资料				
会员	登录				
会员	提交用户名、密码				
会员	修改密码				

- 识别系统边界和Actor（参与者）
- 列出事件
- 识别用例
- 书写用例文档
- 识别用例的关系
- 对用例进行优先级排序



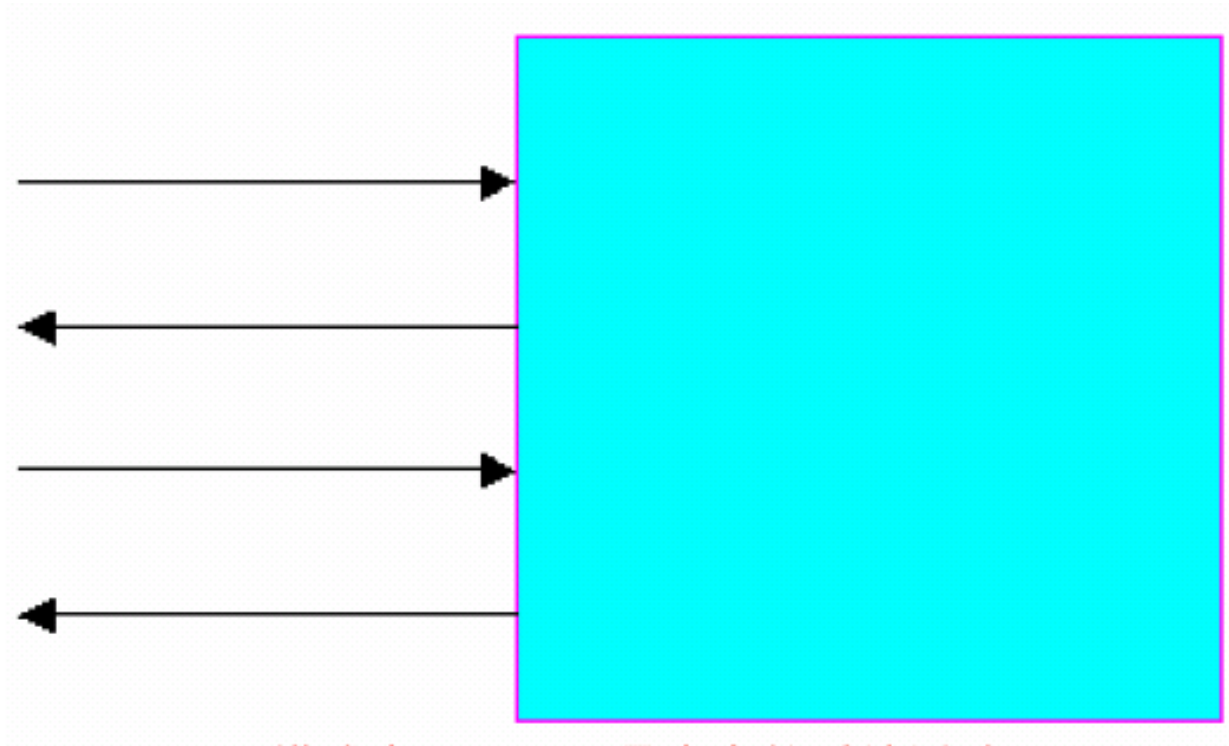
# 用例要点

№ 55

- 可观测→用例止于系统边界
- 结果值→用例是目标导向的
- 系统执行→结果值由系统生成
- 由参与者观测→业务语言，用户观点
- 一组用例实例→用例的粒度

# 用例止于系统边界

№ 56



描述交互，而不是内在的系统活动



# 用例是目标导向的

№ 57

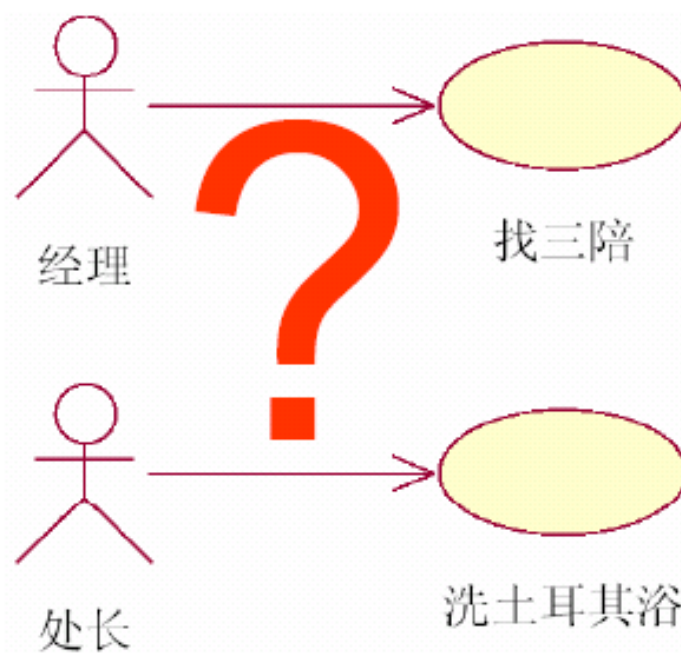
系统的存在是因为：参与者有一些需要使用它来满足的目标



记账：媒介变了，目标没变

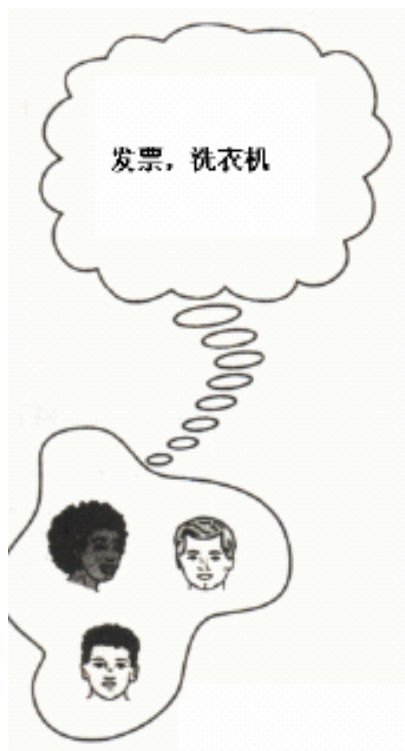
## 结果由系统生成

№ 58



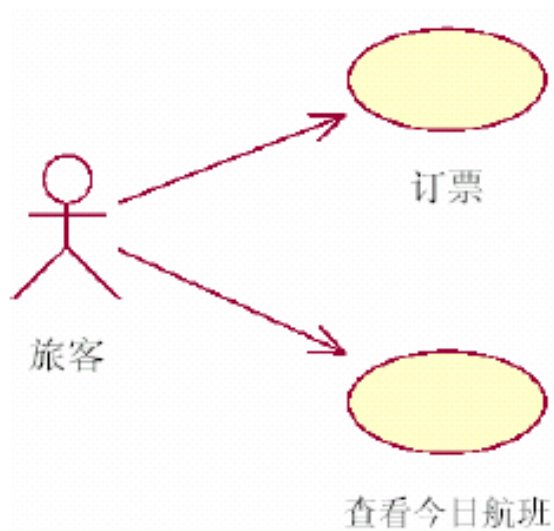
## 业务语言而非技术语言

№ 59

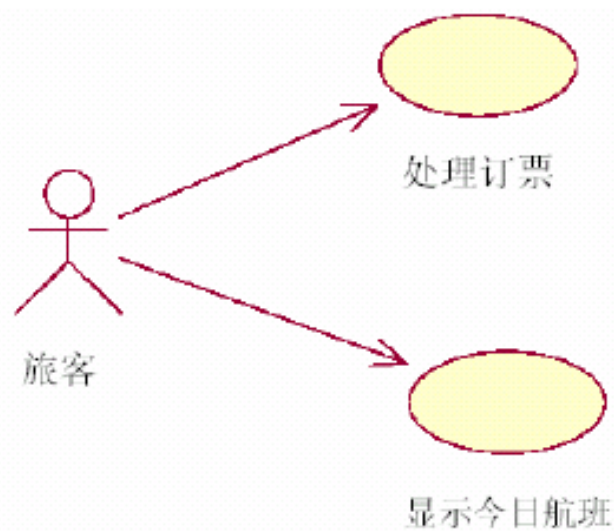


## 用户观点，非系统观点

№ 60



用户观点



系统观点

## 用例

- 呼叫某人
- 接听电话
- 发送短消息
- 记住电话号码
- .....

## 用户观点



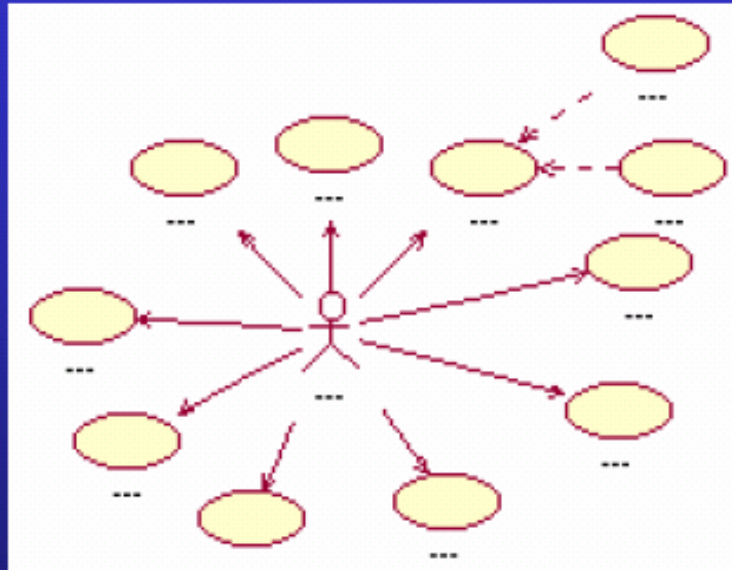
## 功能

- 传输/接收
- 电源/基站
- 输入输出（显示，键盘）
- 电话簿管理
- .....

## 系统观点

# 用例粒度

№ 62

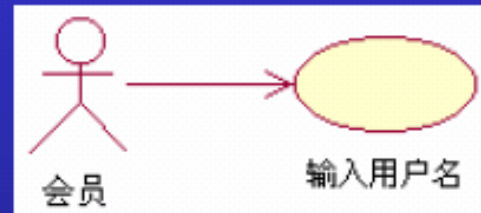


❖ 粒度过细，陷入功能分解

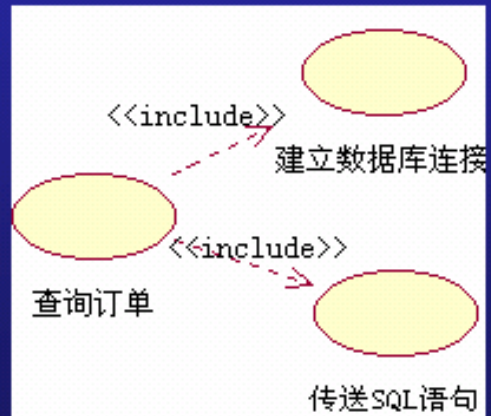
❖ 过细的粒度，一般都会导致技术语言的描述，而不再是业务语言

## ❖ 常见错误:

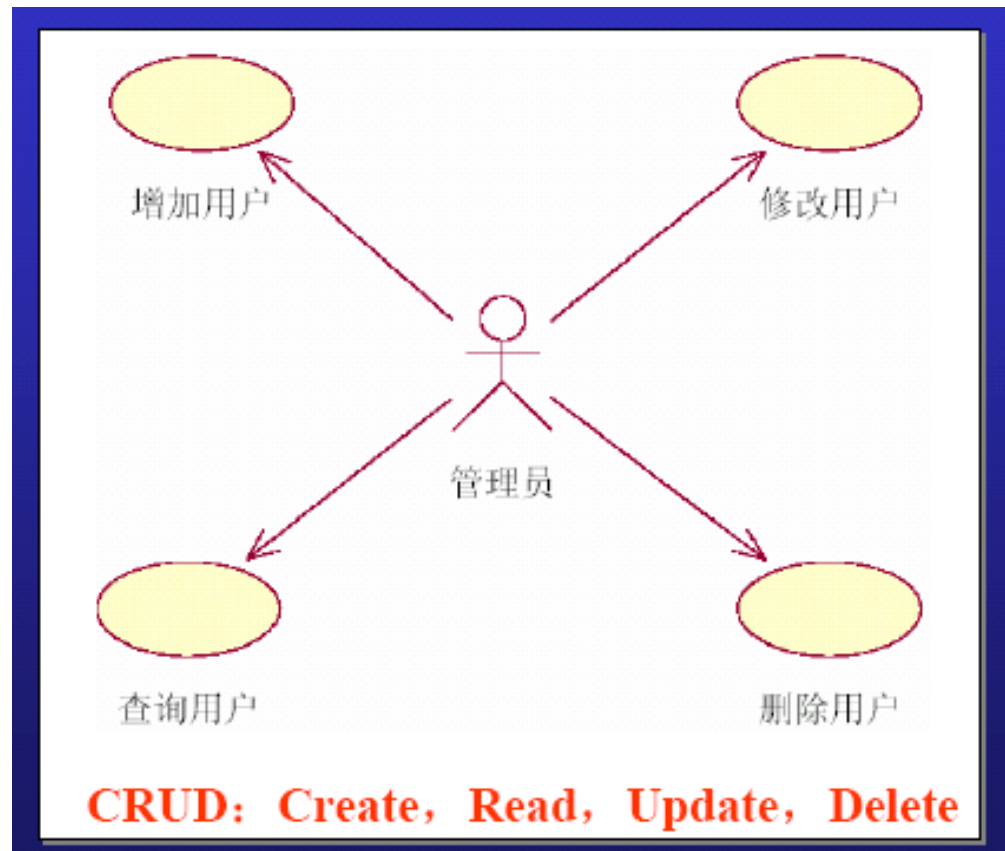
❖ 把交互的某个步骤当作用例



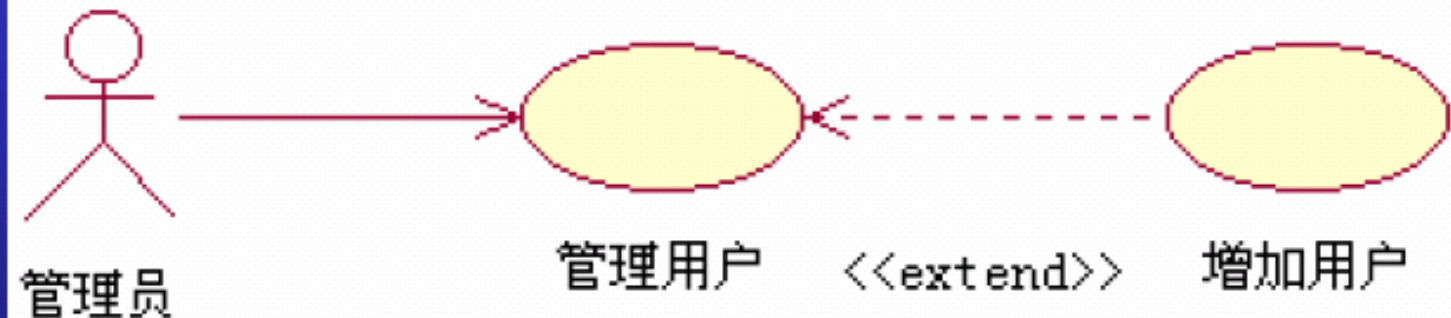
❖ 把系统活动当作用例



## 四轮马车的错误







可以把包含复杂交互的路径独立出去形成用例

## 讨论与练习

№ 66



商品搜索



进行商品搜索



进货管理

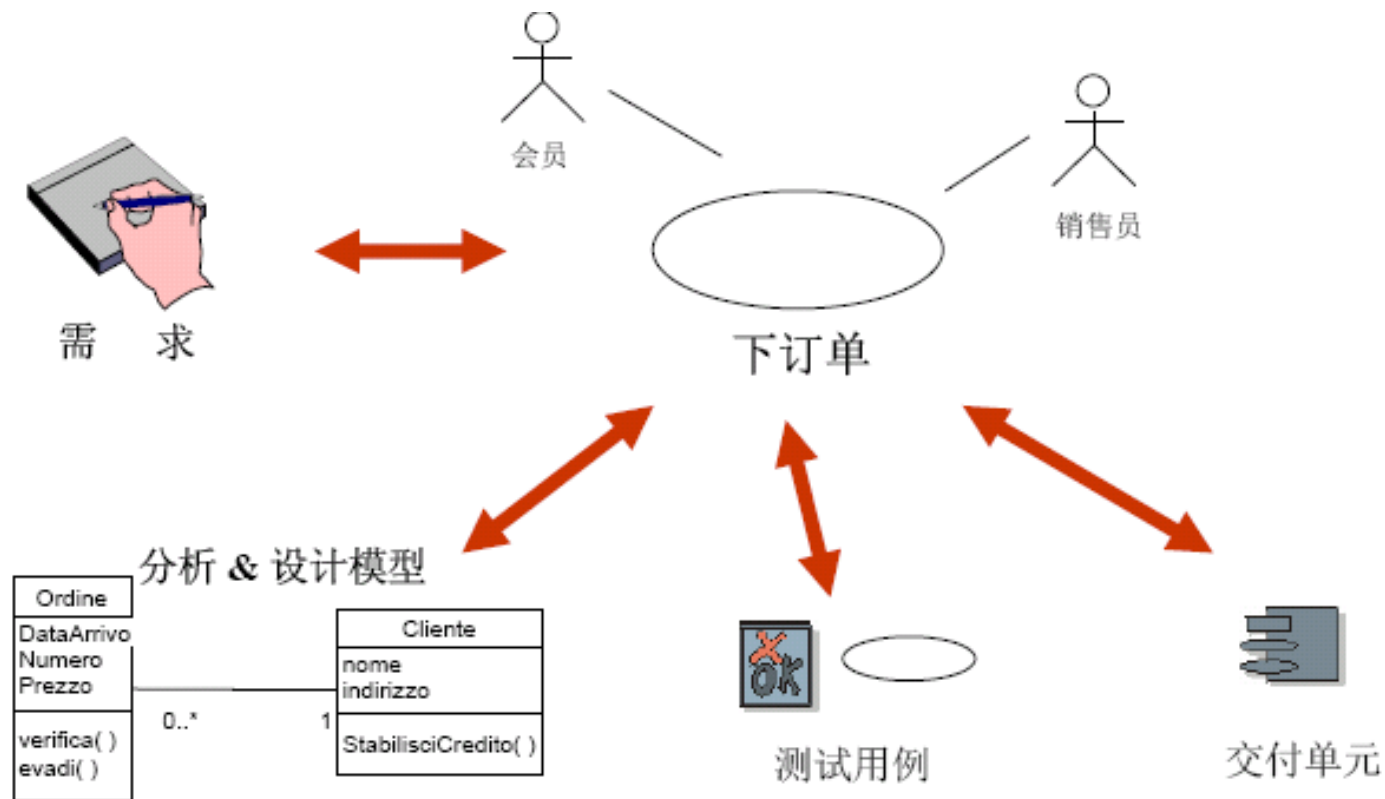


结账功能

以上用例的名字是否合理？

# 用例在开发中的地位

№ 67



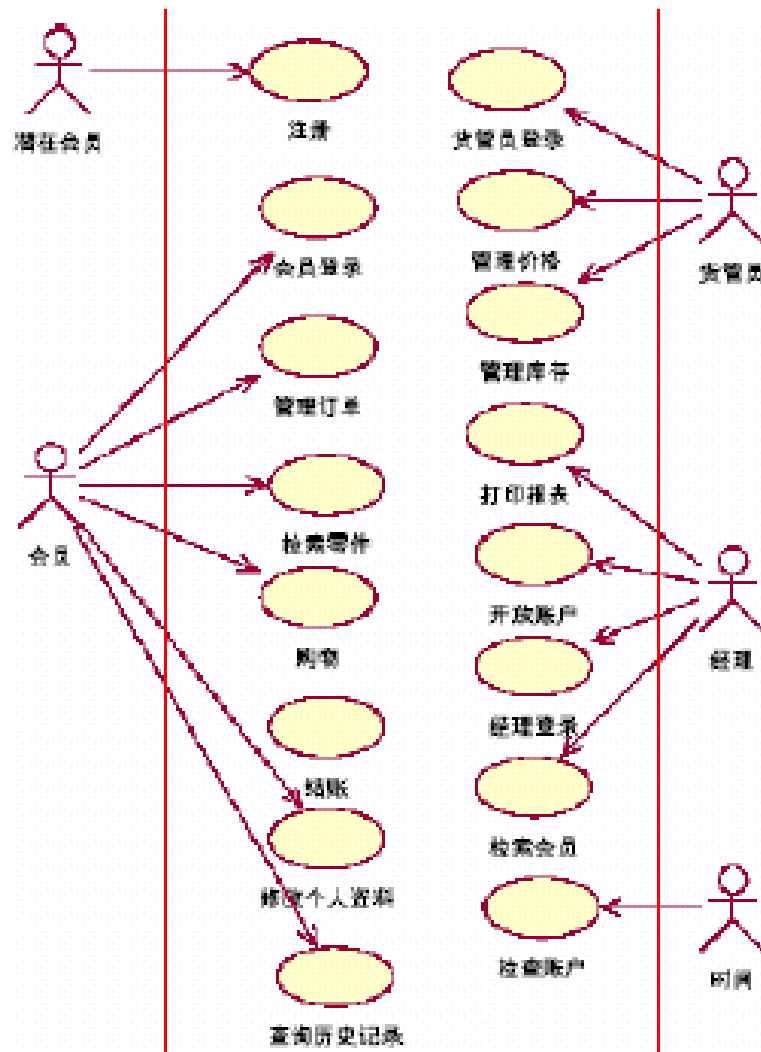
# 从事件识别出用例

№ 68

Actor	事件	业务目的	Actor	事件	业务目的
潜在会员	检索零件		会员	修改个人资料	
潜在会员	输入查询条件		经理	开放账户	
潜在会员	提交查询条件		经理	打印销售情况报表	
潜在会员	查看零件的特征		经理	打印会员情况报表	
潜在会员	查看零件的价格		经理	检索会员	
会员	决定购买某零件的数量		货管员	更新库存	
会员	输入送货地址		货管员	更新价格	
会员	结账		供应商	发货	
会员	改变购买数量		会员	收货	
会员	从订单中删除某一项		会员	付款	
会员	取消订单		时间	每天晚上检查账户	
潜在会员	注册				
潜在会员	提供会员资料				
会员	登录				
会员	提交用户名、密码				
会员	修改密码				

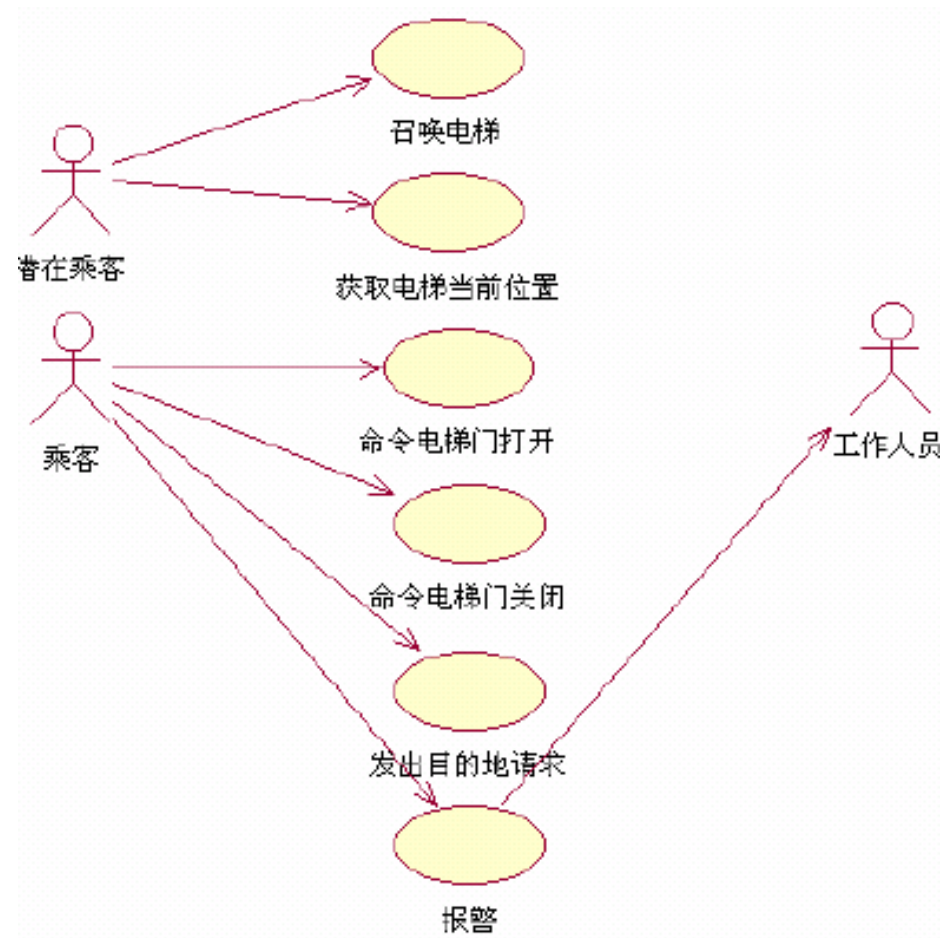
# 用例：零件销售系统

№ 69





电梯——参与者和用例？



# 酒店联合订房系统

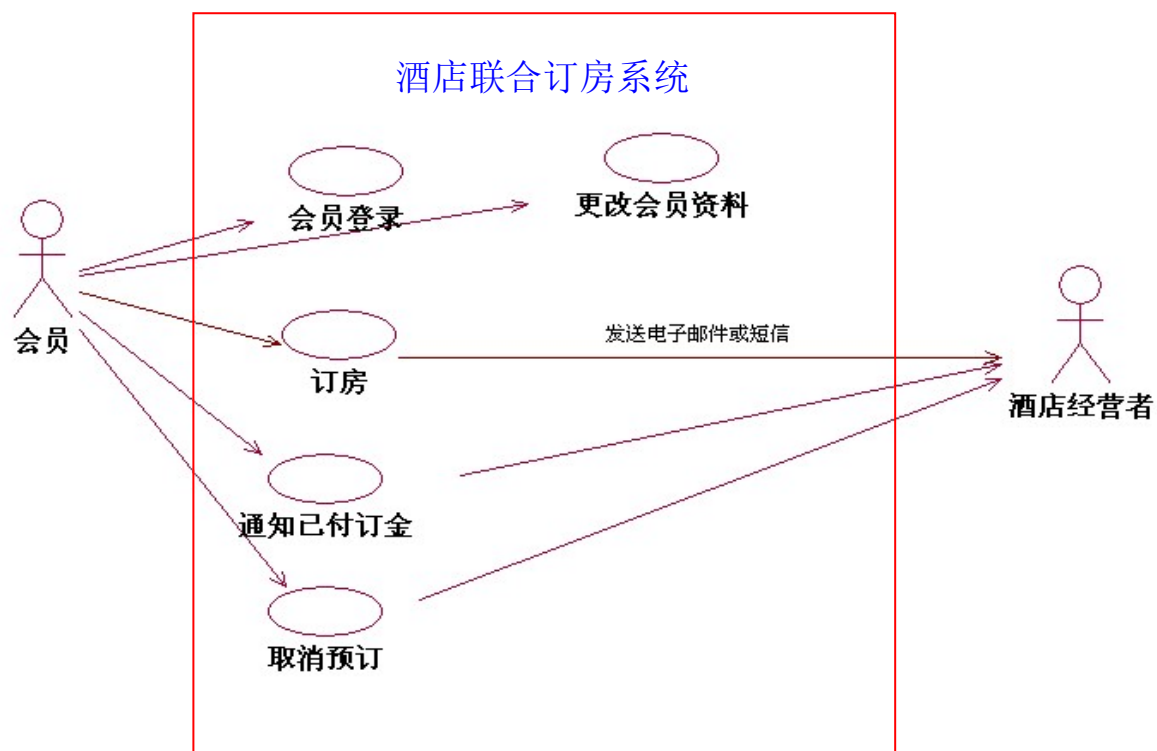
72

- ⊕ 1 用例图
- ⊕ 2 用例—会员登录
- ⊕ 3 用例—订房
- ⊕ 4 用例—通知已预订
- ⊕ 5 定时不定量



# 1 用例图

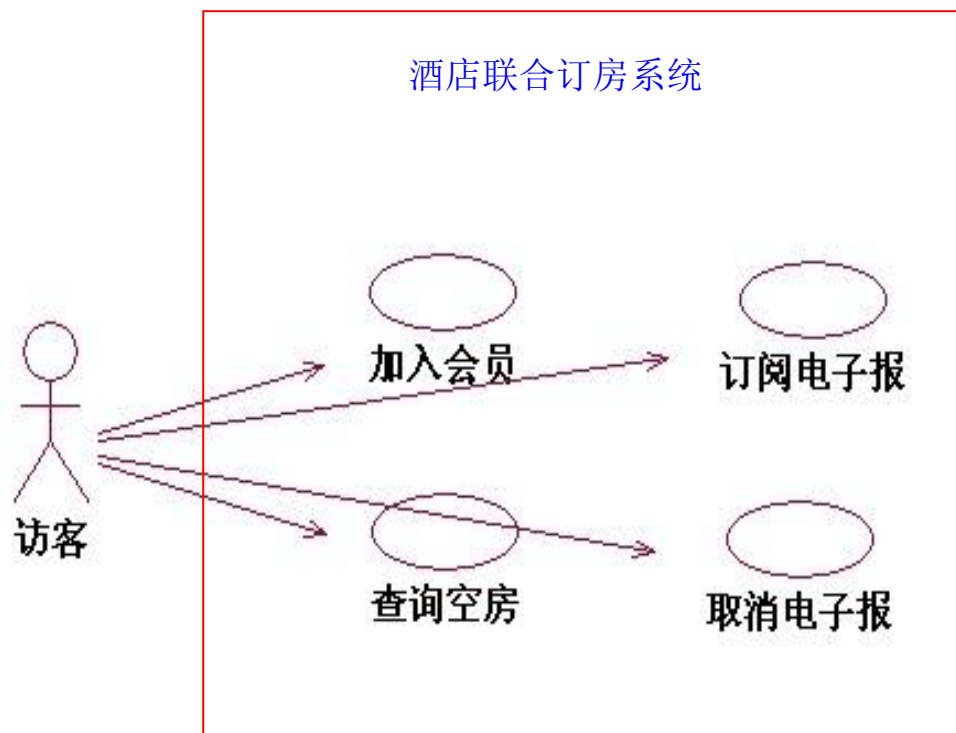
73



用例图(会员)

# 1 用例图

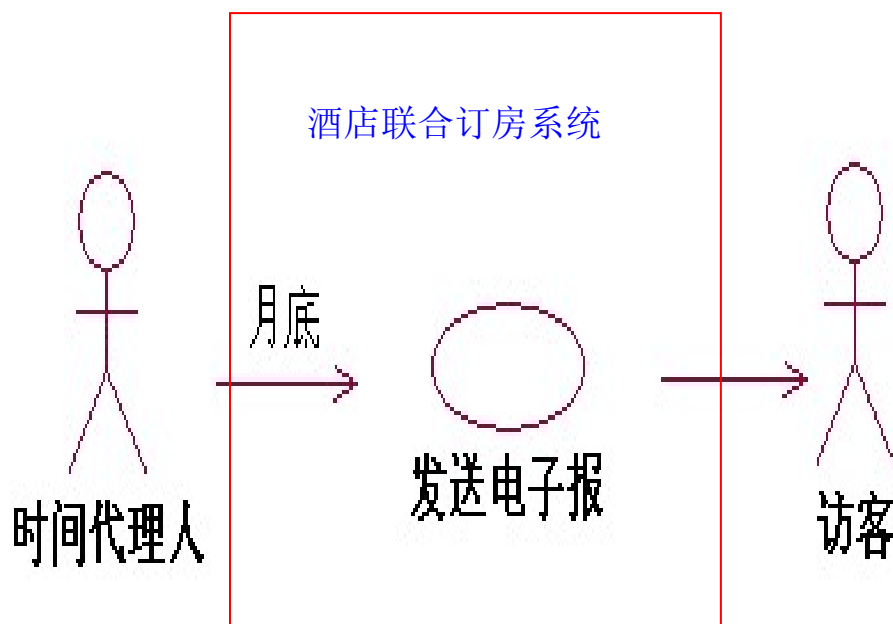
74



用例图(访客)

# 1 用例图

75



用例图(时间代理人)

# 2 用例—会员登录

用例	会员登录		
启动者	会员	支持者	
<b>主要流程</b> 1.会员输入电子邮件和密码。 2.系统确认会员身分之后，出现欢迎信息。			
<b>替代流程</b> <input type="checkbox"/> 数据不完整：客户端提醒会员填入数据，直到数据完整才传送到服务器端。 <input type="checkbox"/> 验证失败：累积5次登录失败，即锁定，并出现请会员主动联系系统管理员的信息。			
<b>企业规则</b> BR1：以会员电子邮件做为会员代号。 BR2：会员累积5次登录失败，即锁定该会员账号。只要登录成功，则失败次数归零。 BR3：一个人只能申请一个会员身份。			
<b>议题与其他</b> 1.由于，一个人只能申请一个会员身份，所以将类图中的个人与会员合并为一。			

## “会员登录”用例描述

### 3 用例—订房

用例	订房		
启动者	会员	支持者	酒店经营者
<p>主要流程</p> <ol style="list-style-type: none"><li>1. 会员挑选一家酒店。</li><li>2. 系统显示出这家酒店所有的房型名称、床型、空房数和房价。</li><li>3. 会员挑选预定的房型、房间数以及预订日期。</li><li>4. 系统减少可预订的空房数，并且新增一条订房事务。</li><li>5. 系统显示出事务代号、订金与总价。</li><li>6. 系统提醒会员需要48小时内支付预订金。</li><li>7. 系统发送订房通知给酒店经营者和会员。</li></ol>			

### “订房”用例描述

### 替代流程

SR1：画面上要标记出必须填写的字段，并且在客户端先检验字段，并且提醒用户填写完整数据，直到必填字段完整之后，才会回送到服务器端。

### 企业规则

BR4：订房事务序号的编码规则为“预订yyyyMMdd0001”，每日以流水号0001起始，每日流水号最大到9999。

BR5：订金=总价 $\times$ 0.1。

BR6：会员需在事务成立后，48小时内支付预订金。

### 议题与其他

1.编写“系统规则”(System Rule,SR)，做为整个订房系统都必须遵守的规则。

2.企业规则和系统规则将集中管理，用例描述中仅片面记录规则初次出现的时刻。

## “订房”用例描述

## 4 用例—通知已预订

用例	通知已预订		
启动者	会员	支持者	酒店经营者
<b>主要流程</b> 1. 会员选择一条未预订的订房事务。 2. 会员填入预订金额、预订账号、预订时间。 3. 系统记录预订数据。 4. 系统发送预订通知电子邮件或短信给酒店经营者和会员。			

### “通知已预订”用例描述

## 4 用例—通知已预订

---

### 议题与其他

- 1.遗漏了“查询订房数据”和“查询历史订单”这两个用例。
- 2.删掉用例描述中未使用的字段。

### “通知已预订”用例描述



## 4 用例—通知已预订

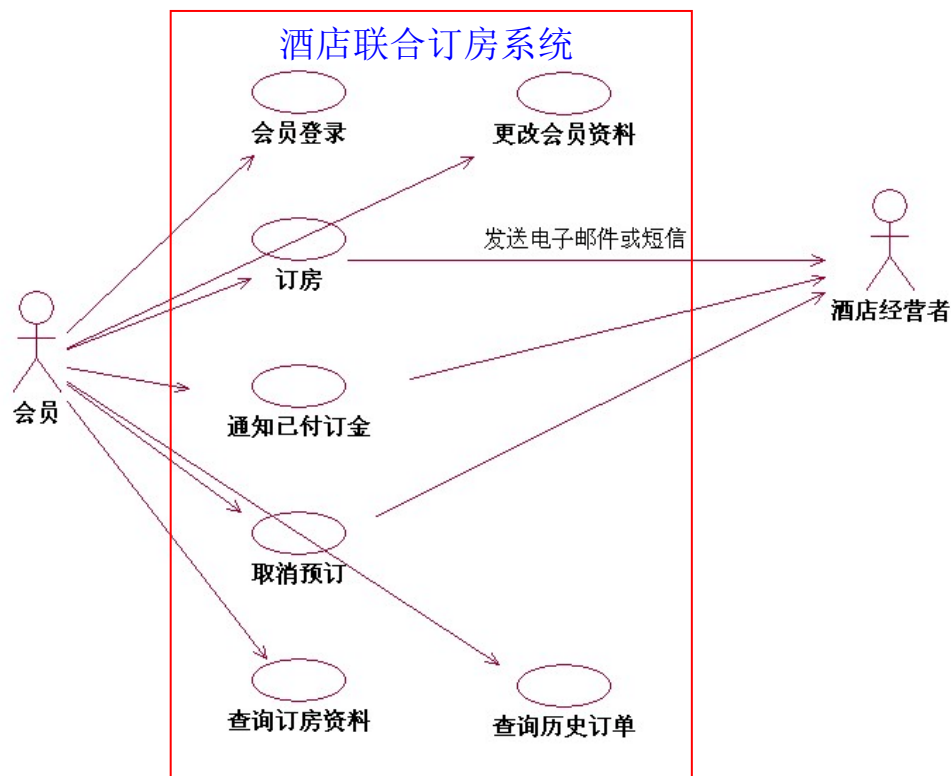
81

订房
-订房序号: String -订房时间: DateTime = now -预订日期: Date = now -订金: Integer = 0 -总计: Integer -支付订金时间: DateTime = now -支付订金帐号: String
+支付订金(支付订金帐号: String, 支付订金时间: DateTime, 订金: Integer)

更新类图(通知已预订)

## 4 用例—通知已预订

82



更新用例图(会员)

# 5 定时不定量

83



彼此联动

## 5 定时不定量

84

- ⊕ 类图必须经过序列图的测试。掌握定时不定量的做法，轮流转动不同的齿轮，别想要一次完成所有的用例描述、类图或序列图。
- ⊕ 取而代之的是，先转动一下类图，接着转动一下用例描述，然后再转动一下序列图，轮流动一动，才能用不同的角度检验彼此、带动彼此的进展。

# 课程登记问题描述

№ 85

- ⊕ 每学期开始学生需要一份课程表，它包含本学期所提供的课程列表及每门课程的相关信息。比如：导师名称、科系、必要条件、课程时间、上课地点，可以帮助学生作出合理的决定
- ⊕ 新系统规定学生可以选择四门必修课程。此外，他还要选择两门候补课程以防某门课程人员满额或被取消。每门课程人数不得多余10人或少余3人。一旦学生完成登记过程，登记系统将信息传入记费系统以便计算学生在本学期的学费数额/

- ⊕ 导师需要随时访问系统，知道有那一门课程需要任教。他也可以了解他的课有那些学生
- ⊕ 每学期开始，学生有一段试听时间，学生可以改变所选课程内容。在这段时间学生必须可以访问系统随时更改课程选项