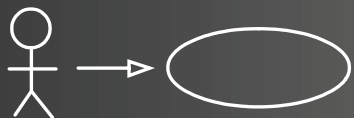


UML全程实作

需求

Think



<http://www.umlchina.com>

核心 workflow

*愿景

*业务建模

选定愿景要改进的业务组织

业务用例图

现状业务序列图

改进业务序列图

*需求

系统用例图

书写用例文档

提升
销售

*分析

类图

序列图

状态图

*设计

建立数据层

精化业务层

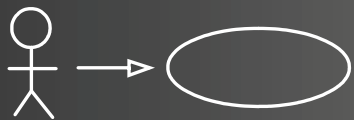
精化表示层

降低
成本



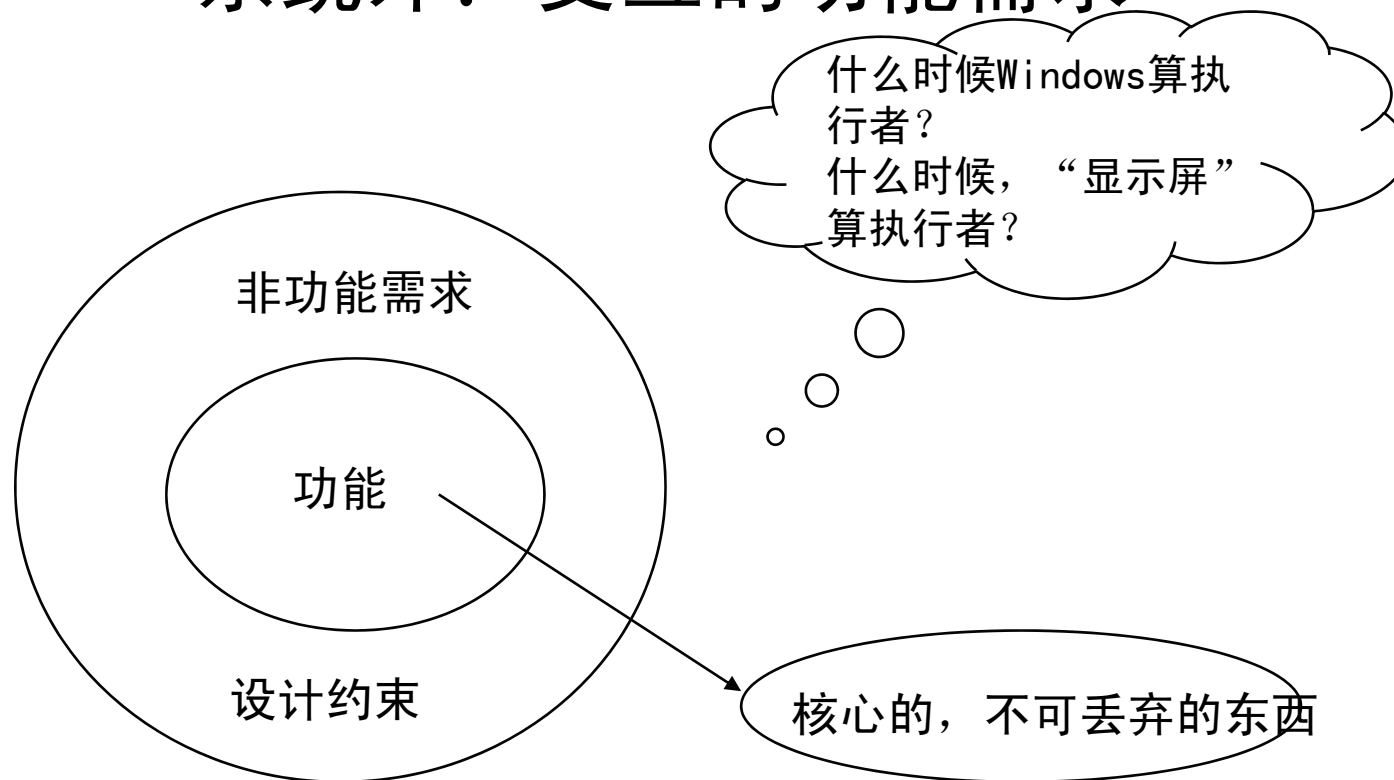
系统执行者

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

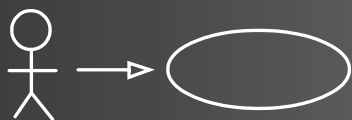


系统执行者

——系统外：交互的功能需求

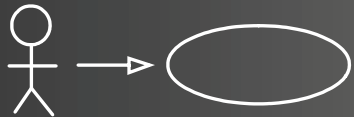
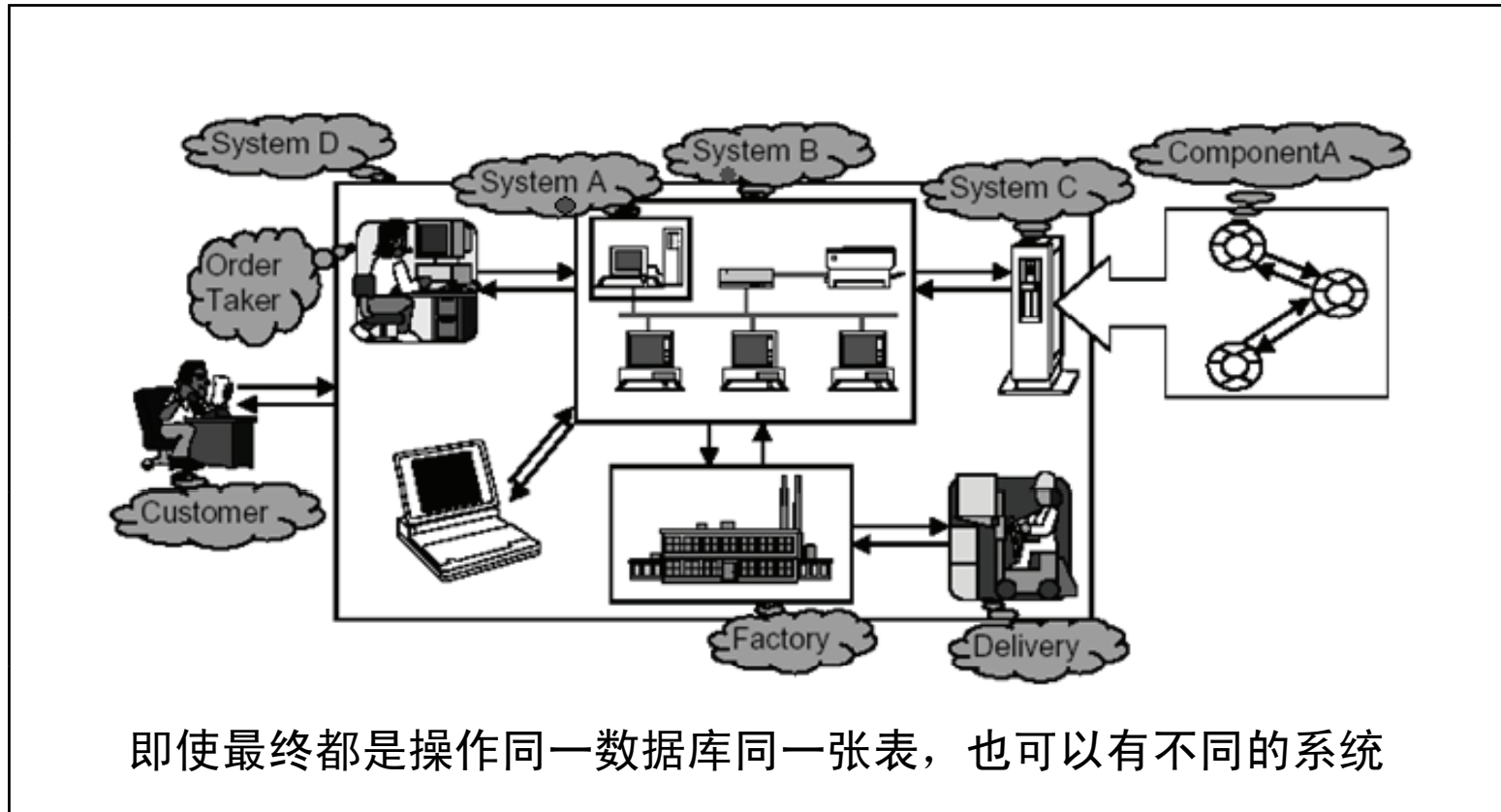


有多少个执行者，代表有多少个接口



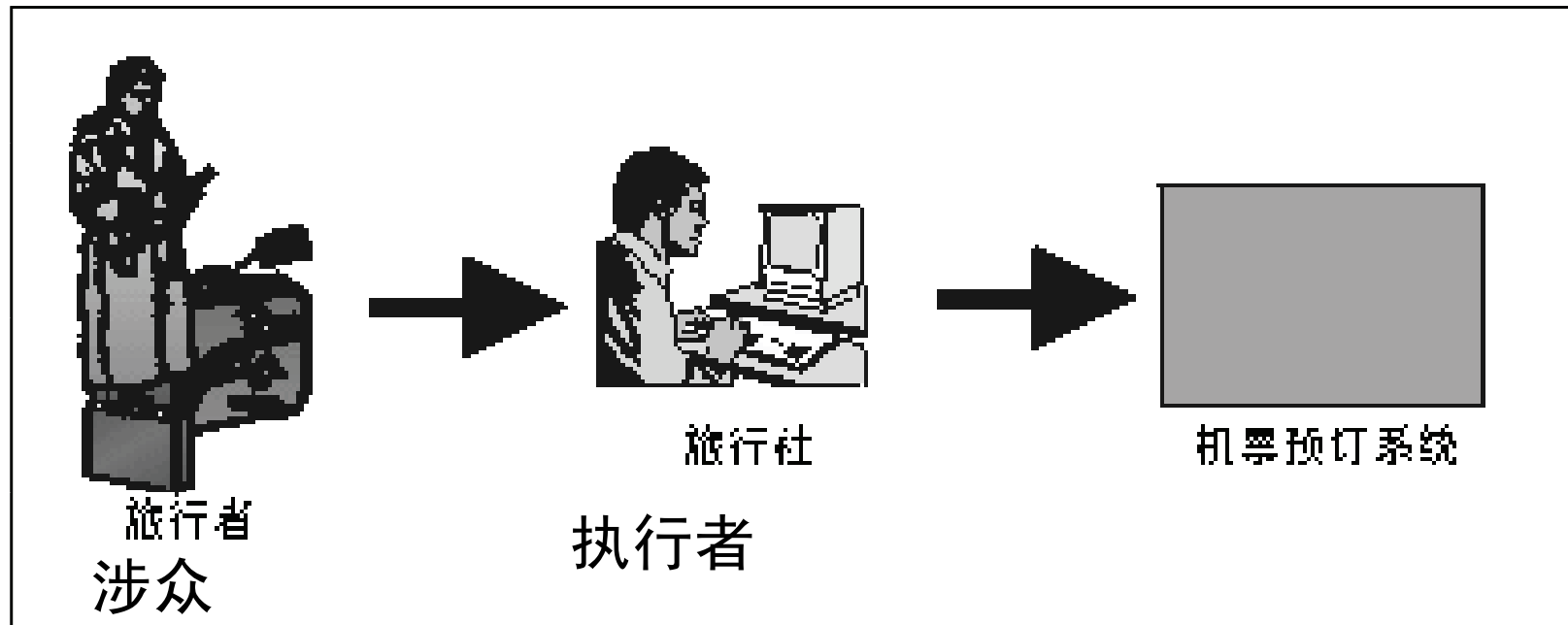
系统执行者

—— 责任的边界，不是物理的边界

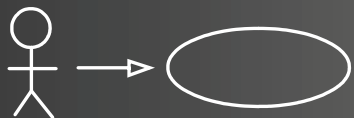


系统执行者

——直接与系统交互

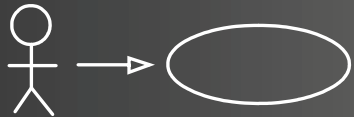
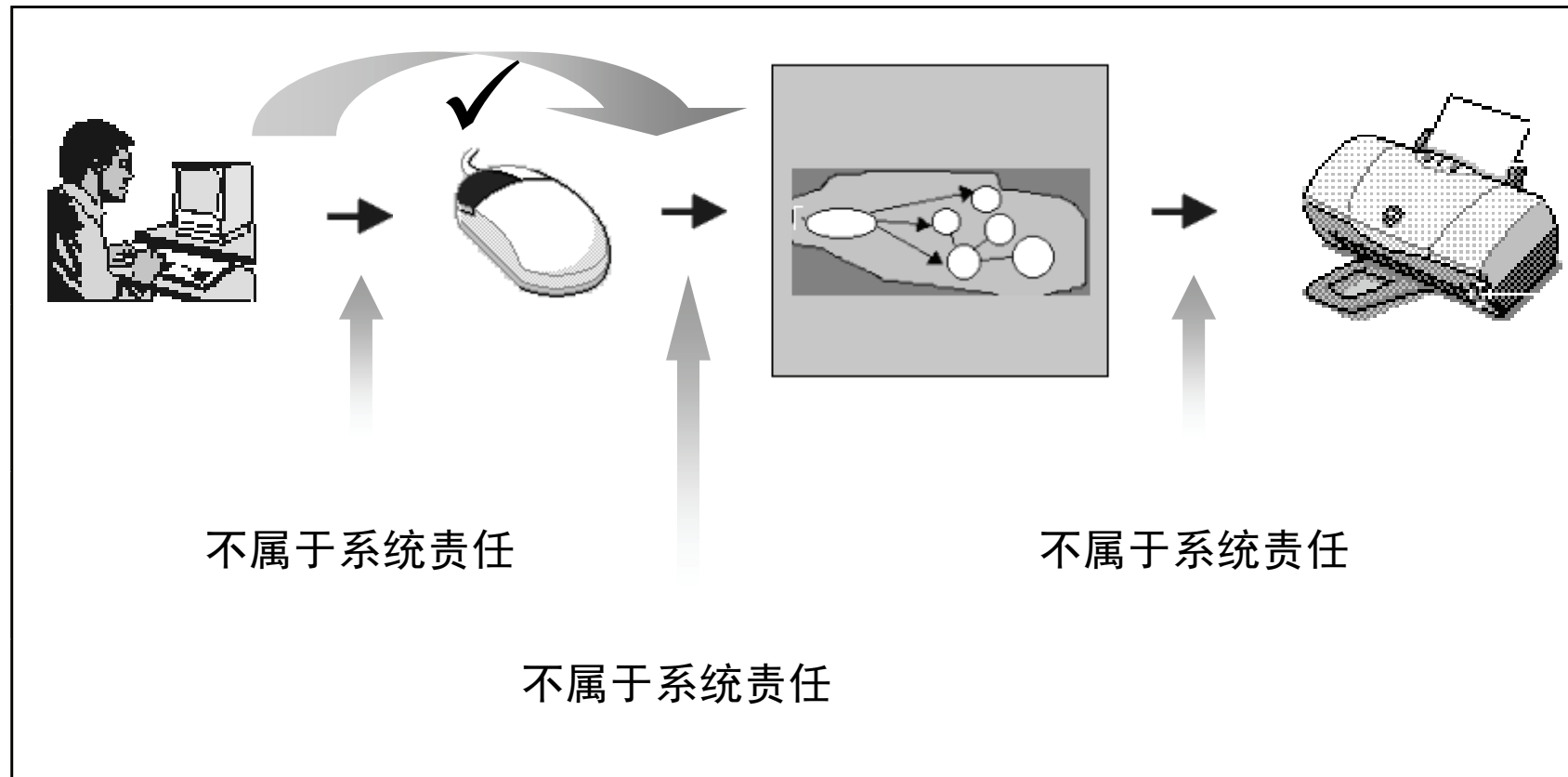


执行者与重要无关



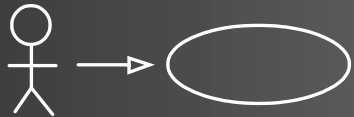
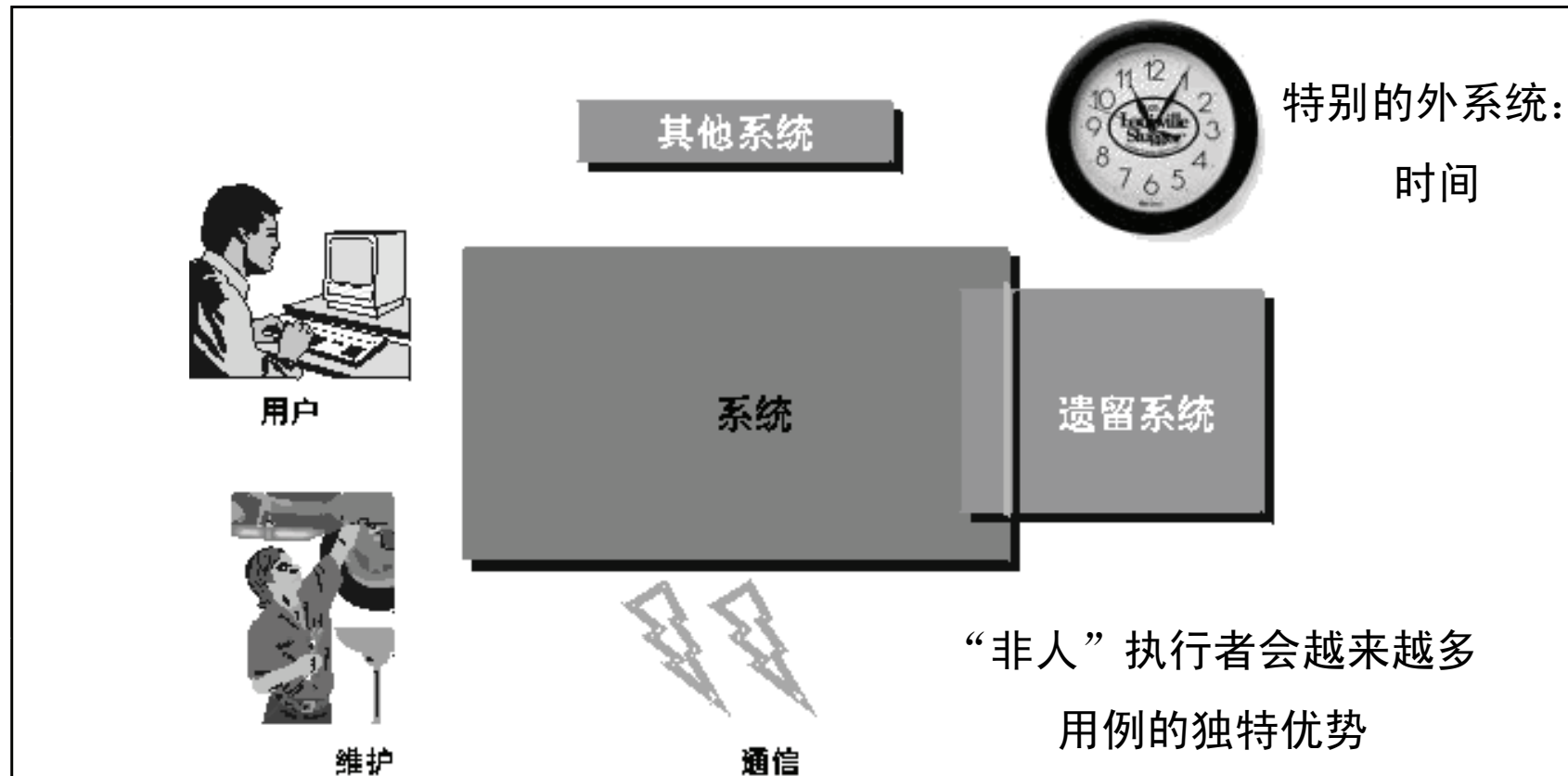
系统执行者

——有意义的交互



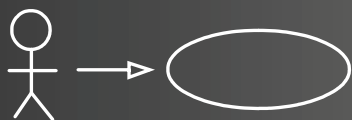
系统执行者

——任何事物



系统执行者

——讨论与练习、项目实作

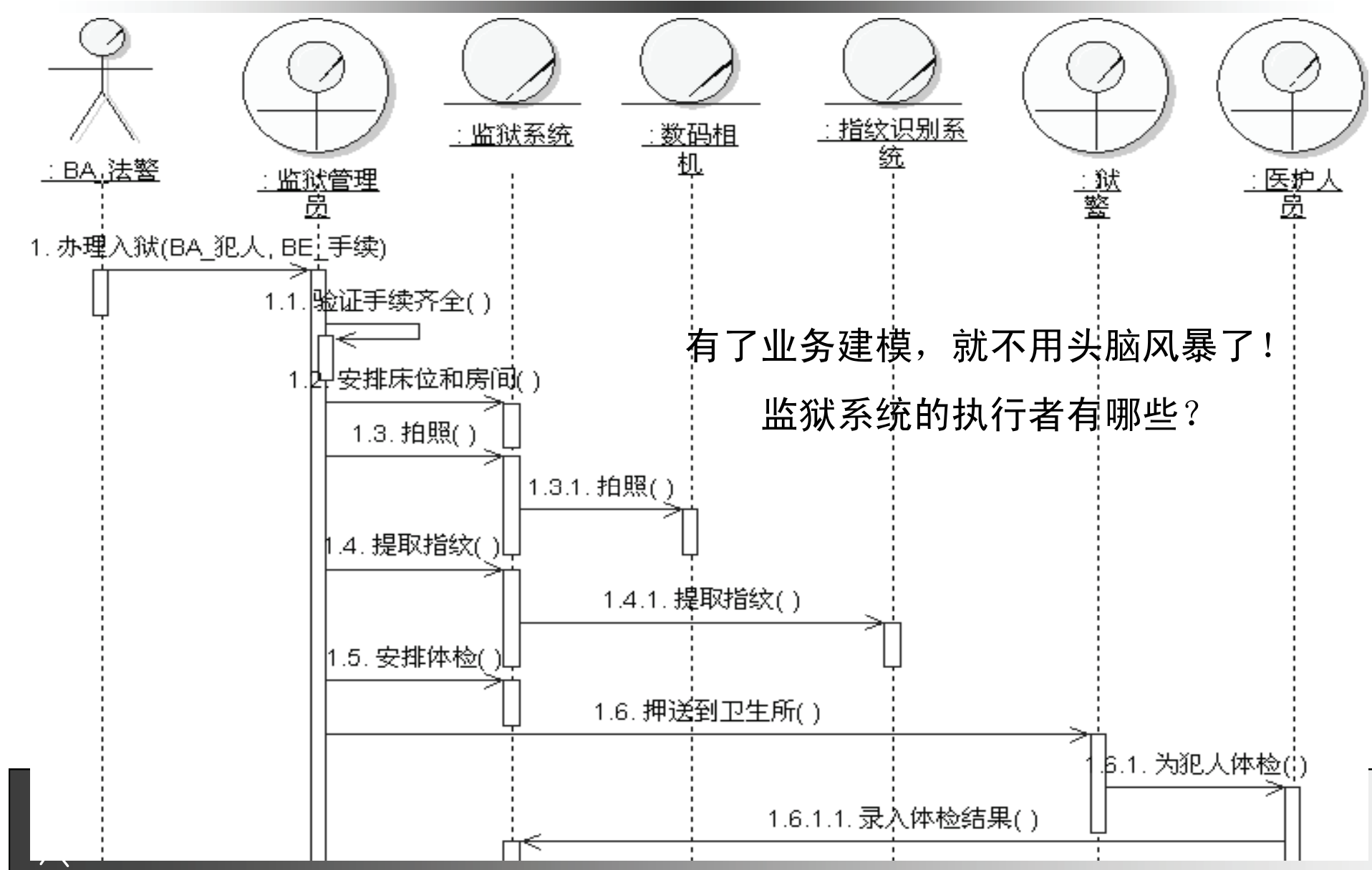


系统执行者

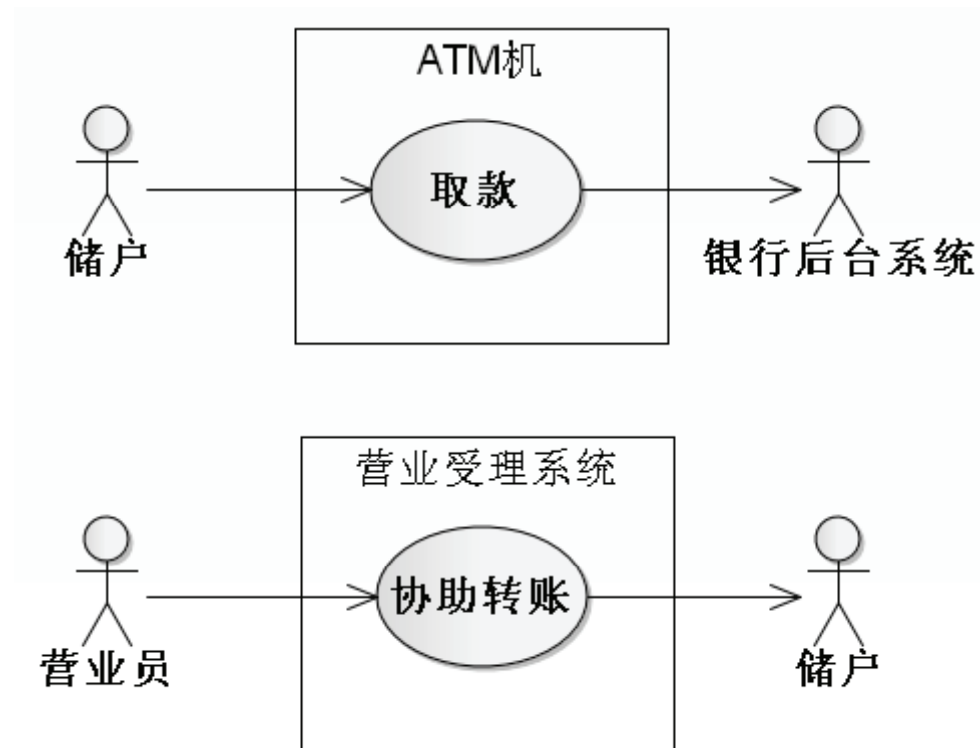
- ❖ 谁使用系统的主要功能？
- ❖ 谁改变系统的数据
- ❖ 谁从系统获取信息
- ❖ 谁需要系统的支持以完成日常工作任务？
- ❖ 谁负责维护、管理并保持系统正常运行？
- ❖ 系统需要应付（处理）哪些硬设备？
- ❖ 系统需要和哪些外部系统交互？
- ❖ 谁（或什么）对系统运行产生的结果感兴趣？
- ❖ 有没有自动发生的事件



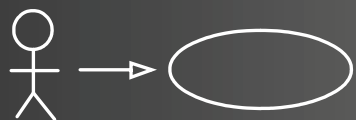
系统执行者



系统执行者



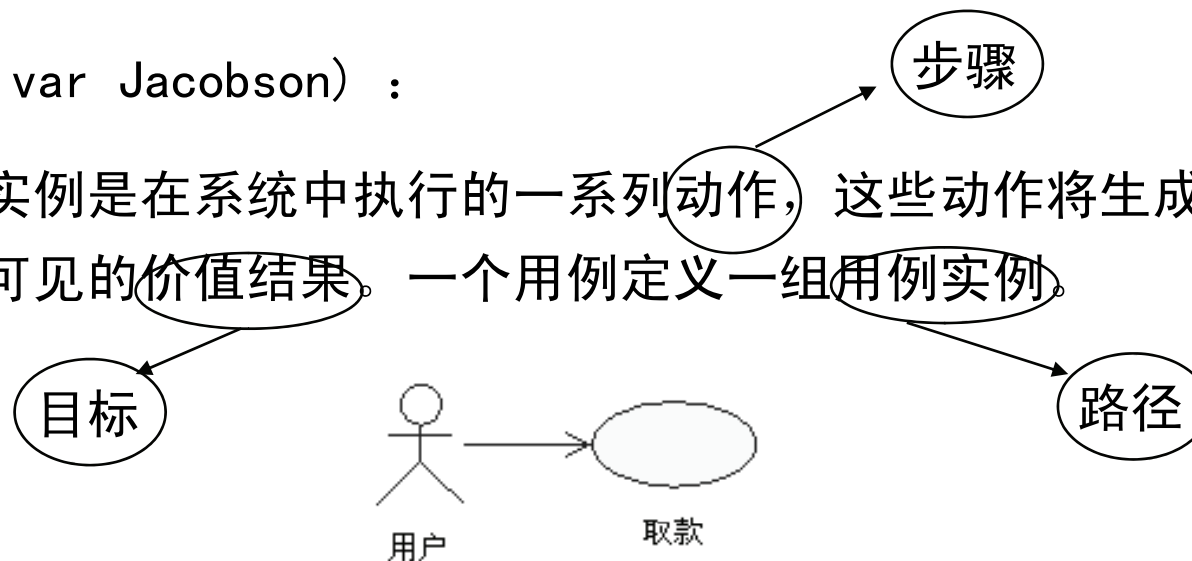
主执行者、辅执行者



系统用例

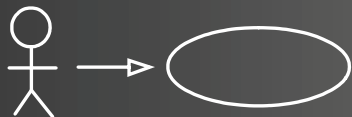
❖ RUP (Ivar Jacobson) :

用例实例是在系统中执行的一系列动作，这些动作将生成特定执行者可见的价值结果。一个用例定义一组用例实例。



❖ 通俗一些

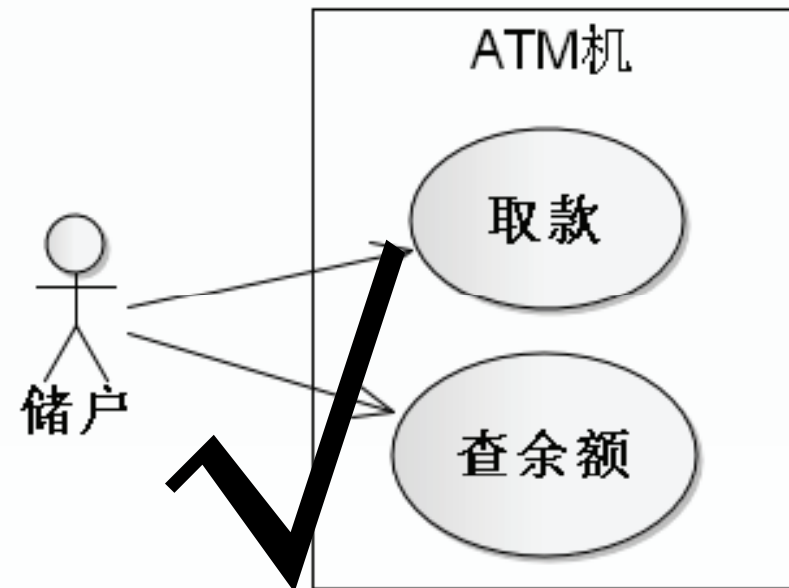
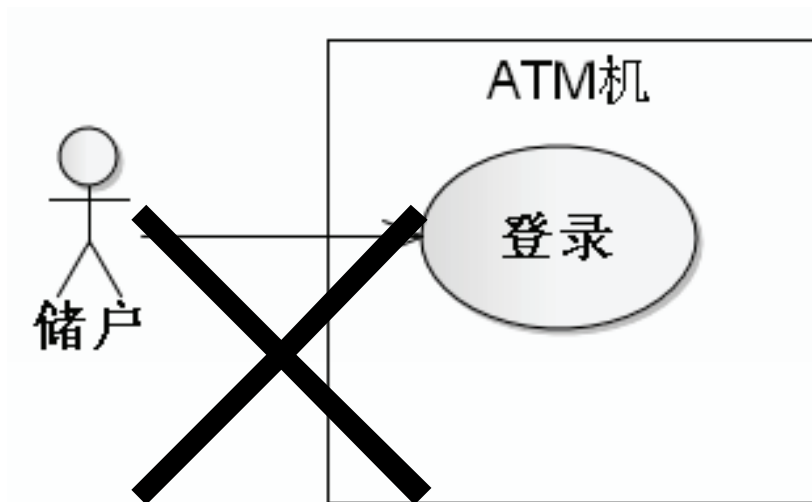
执行者通过系统达到某个目标



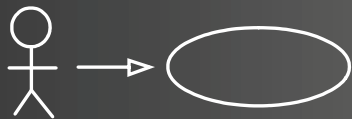
系统用例



契约和承诺!

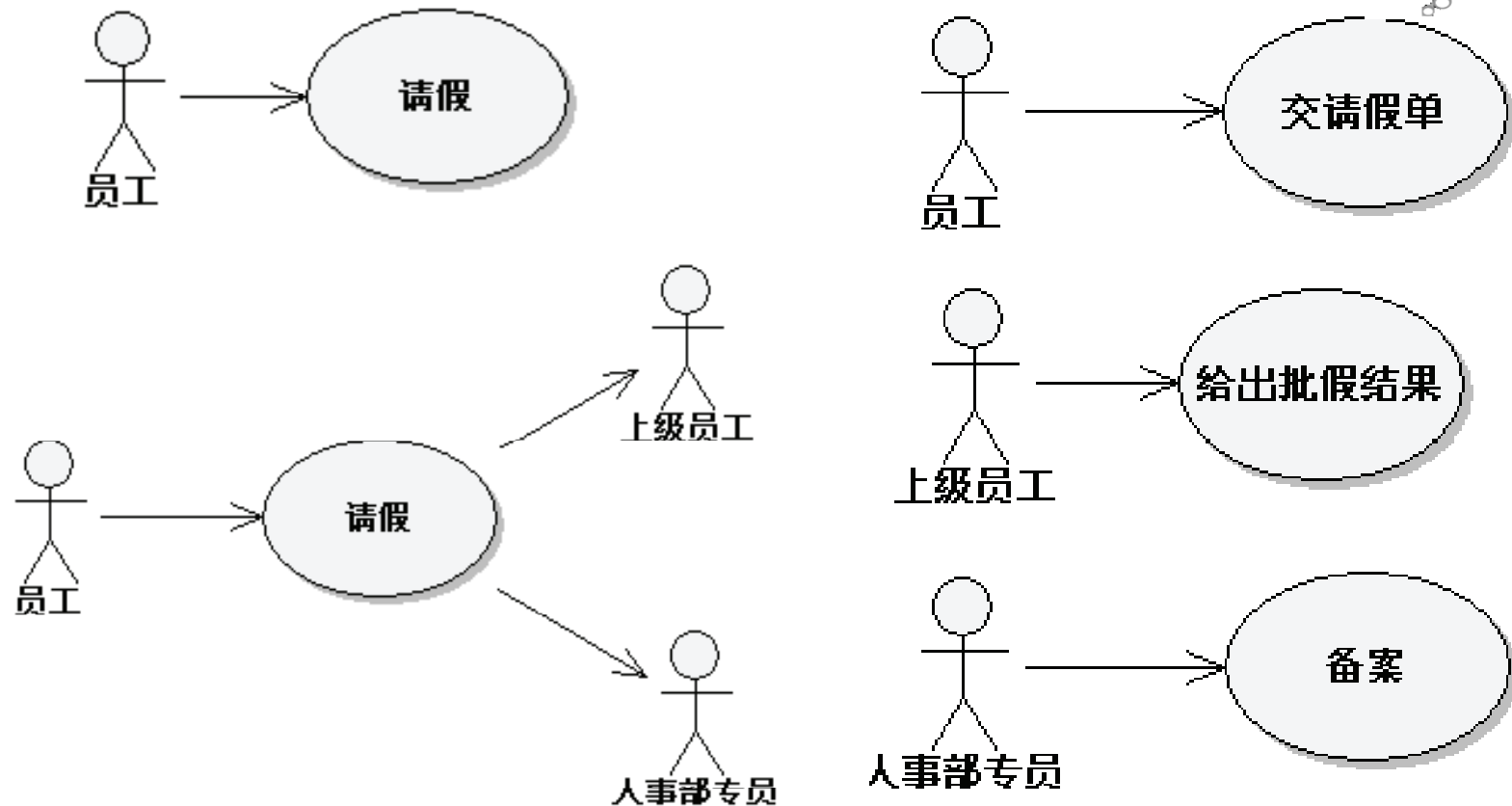


用例是“能卖”的价值

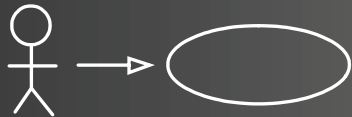


系统用例

研究对象：人事系统

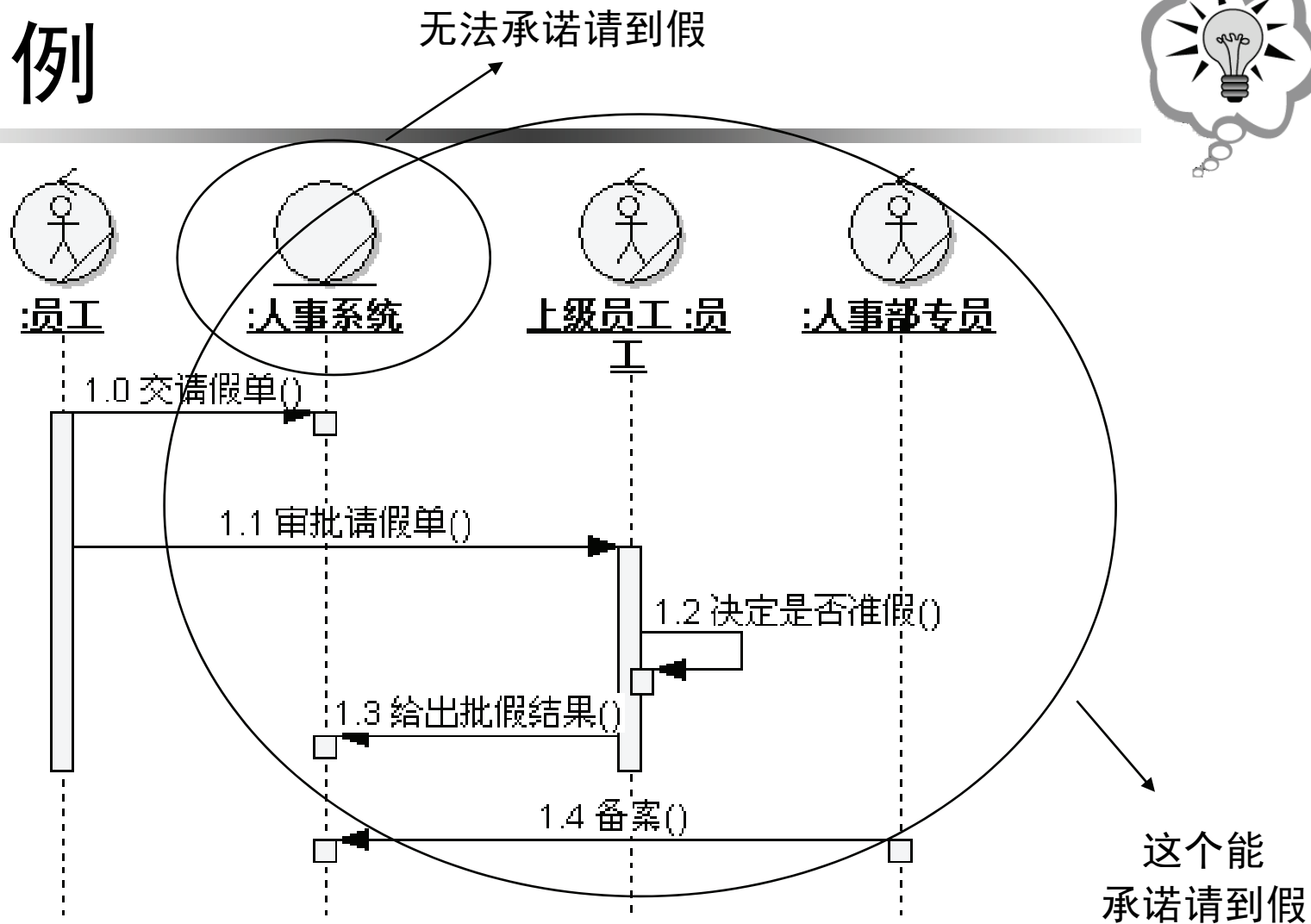


用例多大——期望、契约、承诺

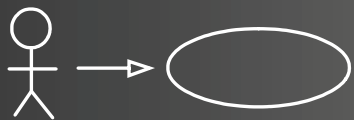


<http://www.umlchina.com>

系统用例



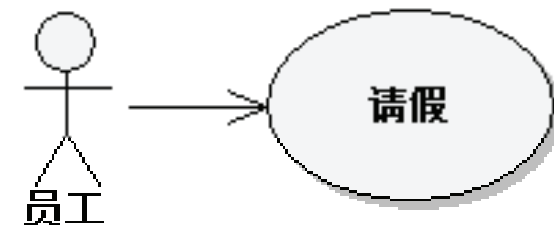
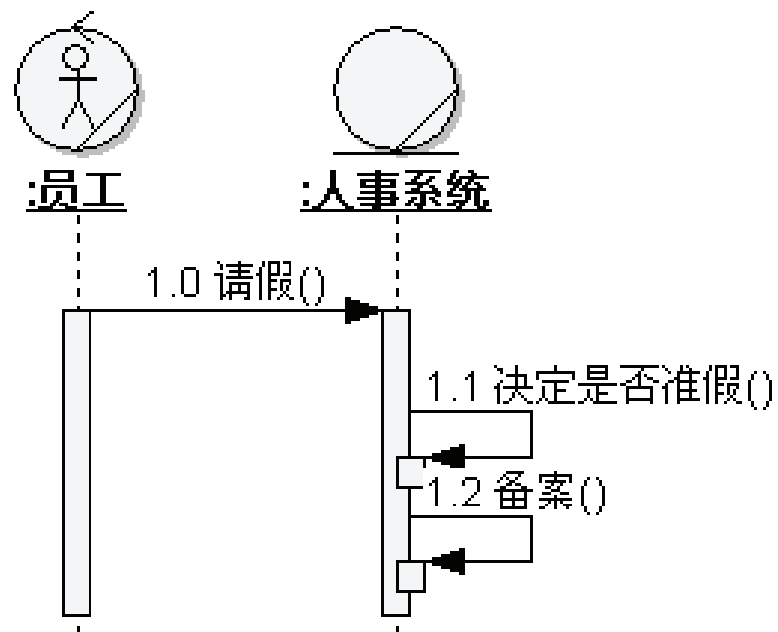
会画业务序列图，这种困惑就会消亡



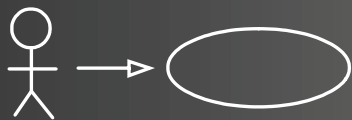
系统用例



研究对象：人事系统

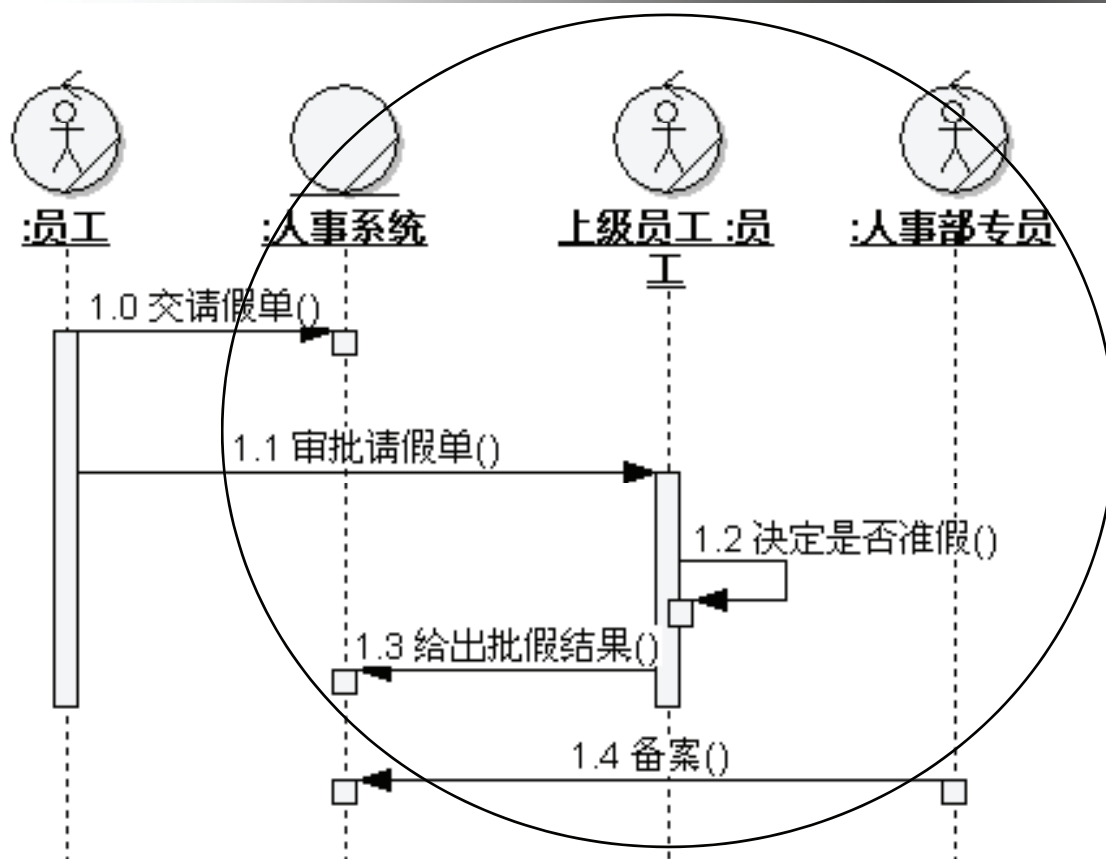


如果是这样，又不一样

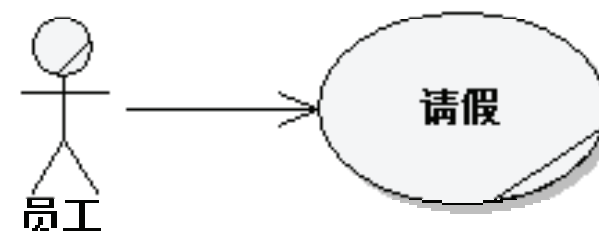




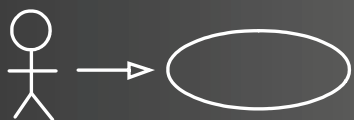
系统用例



研究对象：请假部（上级+人事部）



如果是这样，也不一样





系统用例

这个能算用例吗？

建立数据库连接

业务类请求连接数据库

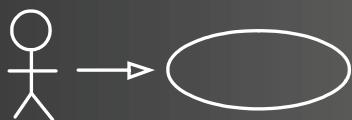
系统请求ADO.NET建立数据库连接

.....

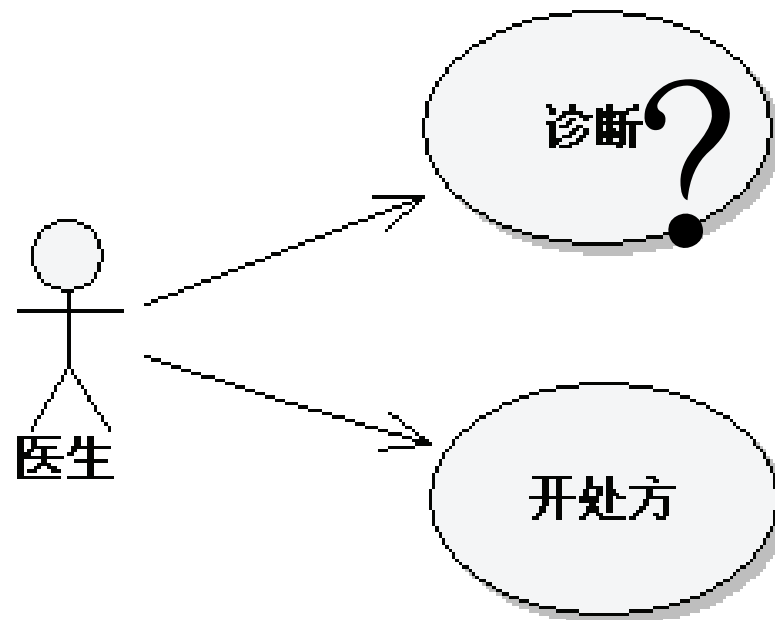
涉众：开发人员

只要能写出涉众可理解和验证的路径、步骤
都可以作为用例

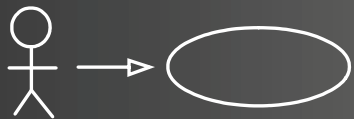
写出文档，这种困惑也会消亡



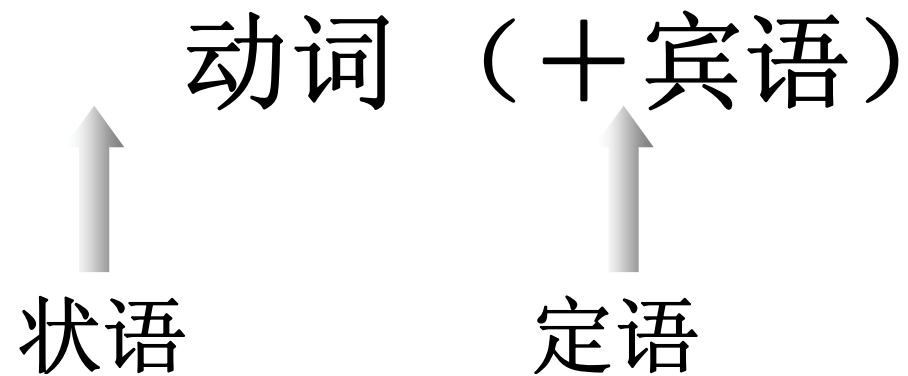
系统用例



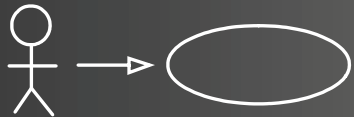
不是某某做什么，而是某某【用系统】做什么



系统用例



命名：动宾结构



系统用例

❖弱动词：进行、使用、复制、加载、生成...

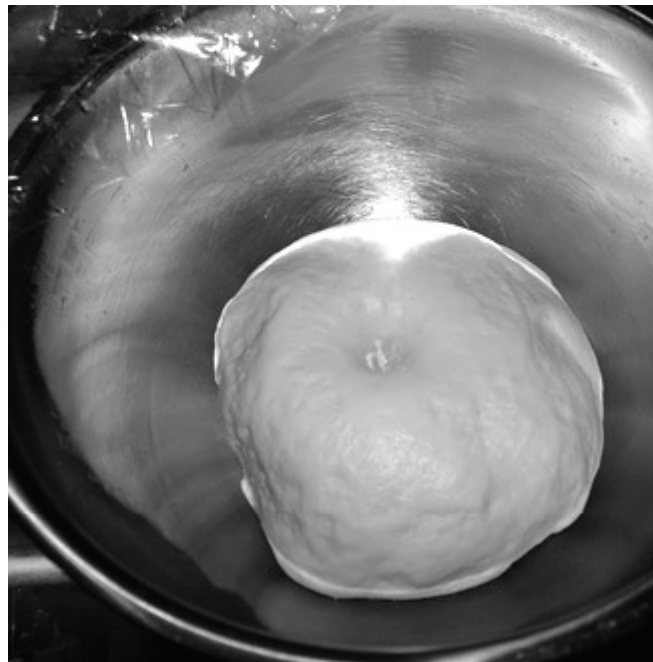
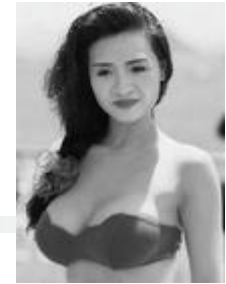
❖弱名词：数据、报表、表格、表单、系统...

慎用弱动词弱名词——会掩盖真正的业务



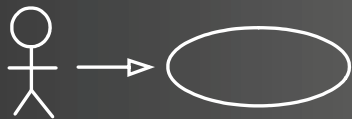
系统用例

——“粒度”



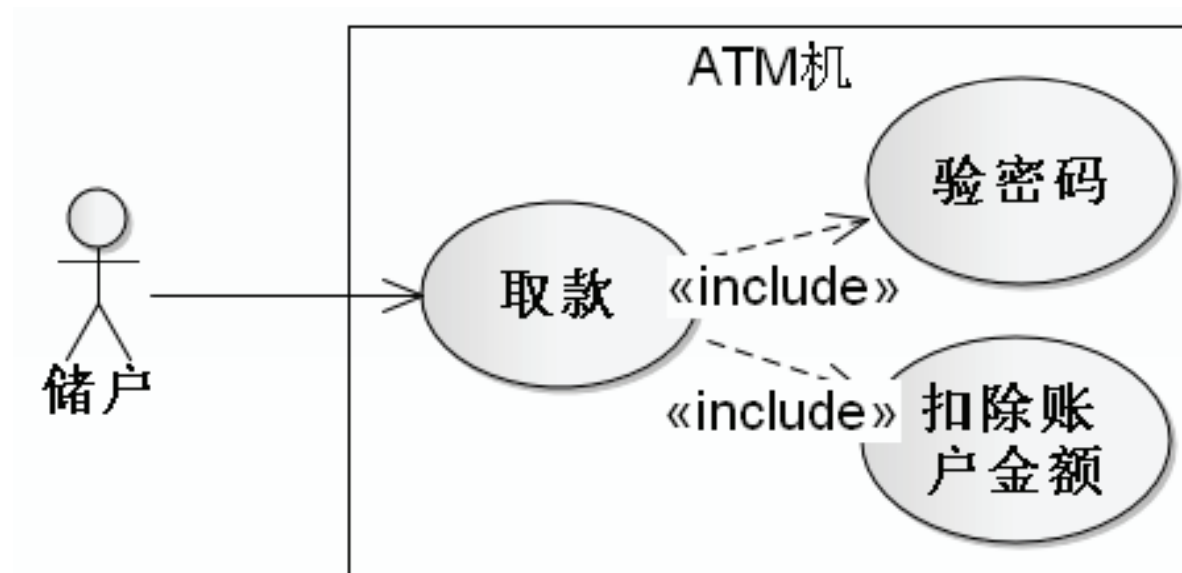
用例不是面团

用例不存在“粒度问题”

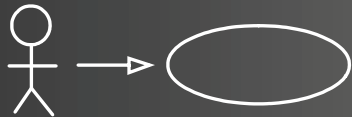


系统用例

——“粒度”

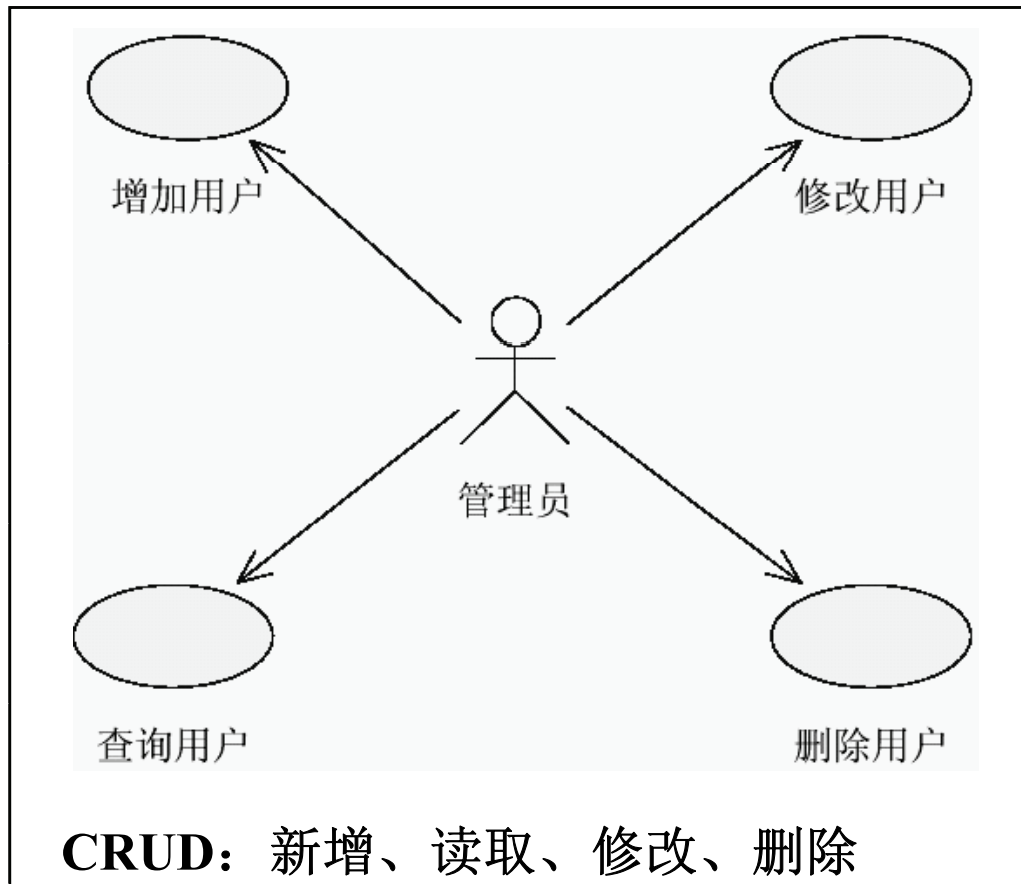


最常犯错误：把步骤当作用例



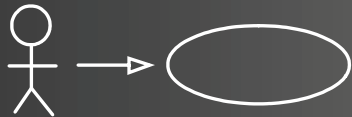
系统用例

——“粒度”：四轮马车

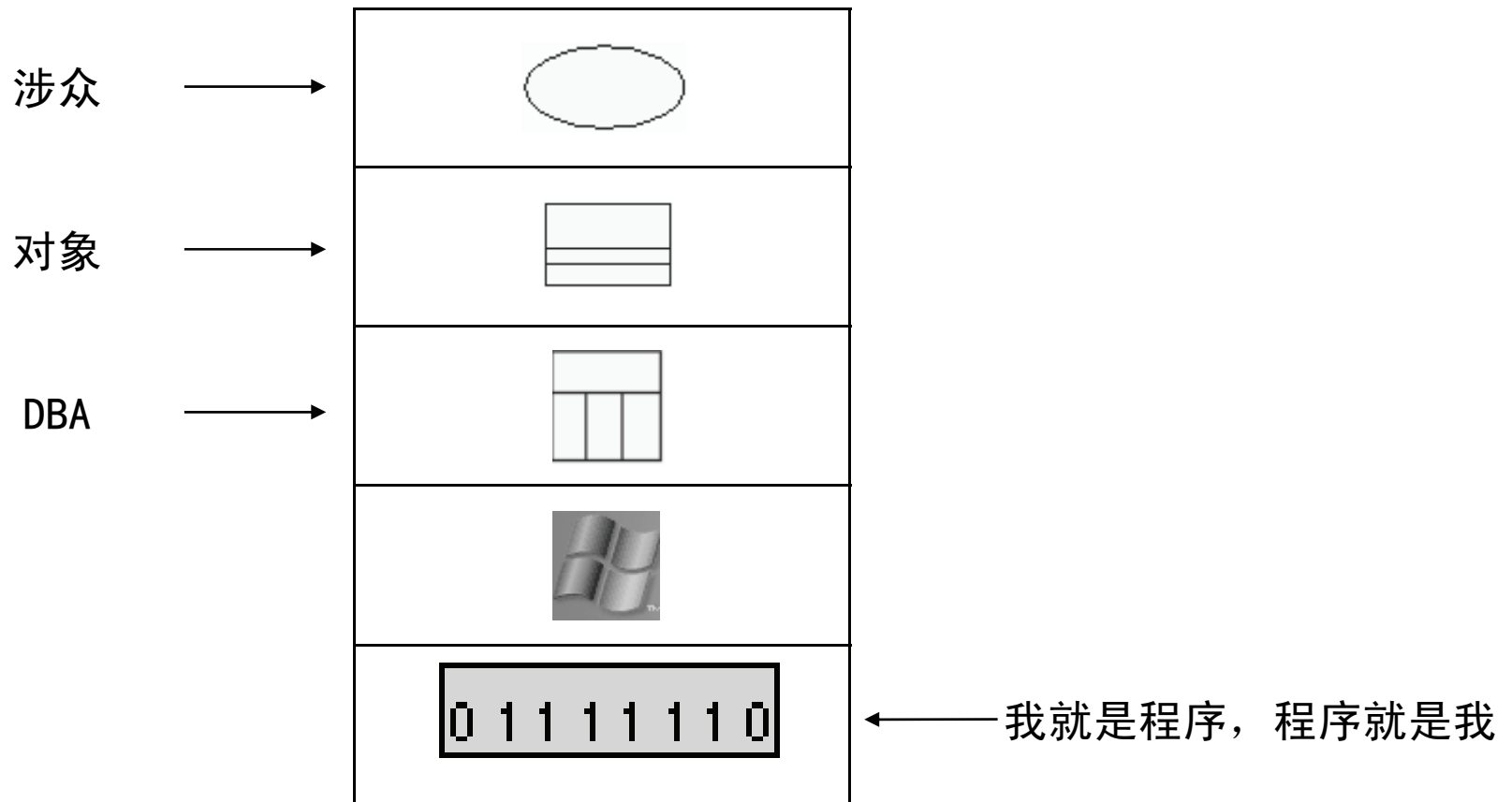


❖用有色眼镜看，所有业务最终都会成为CRUD

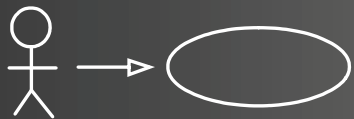
❖多问：为什么要CRUD？光CRUD能为执行者提供价值吗？



系统用例

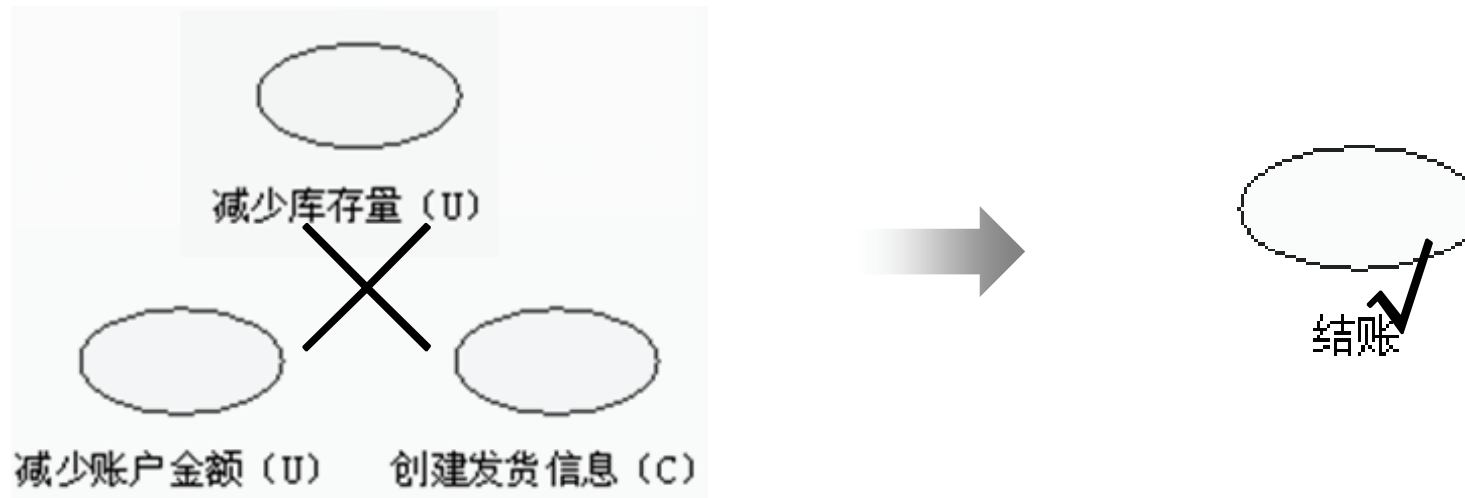


需求只有一个视角：涉众视角

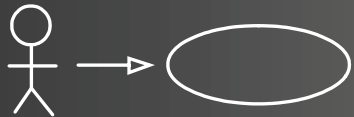


系统用例

——“粒度”：四轮马车

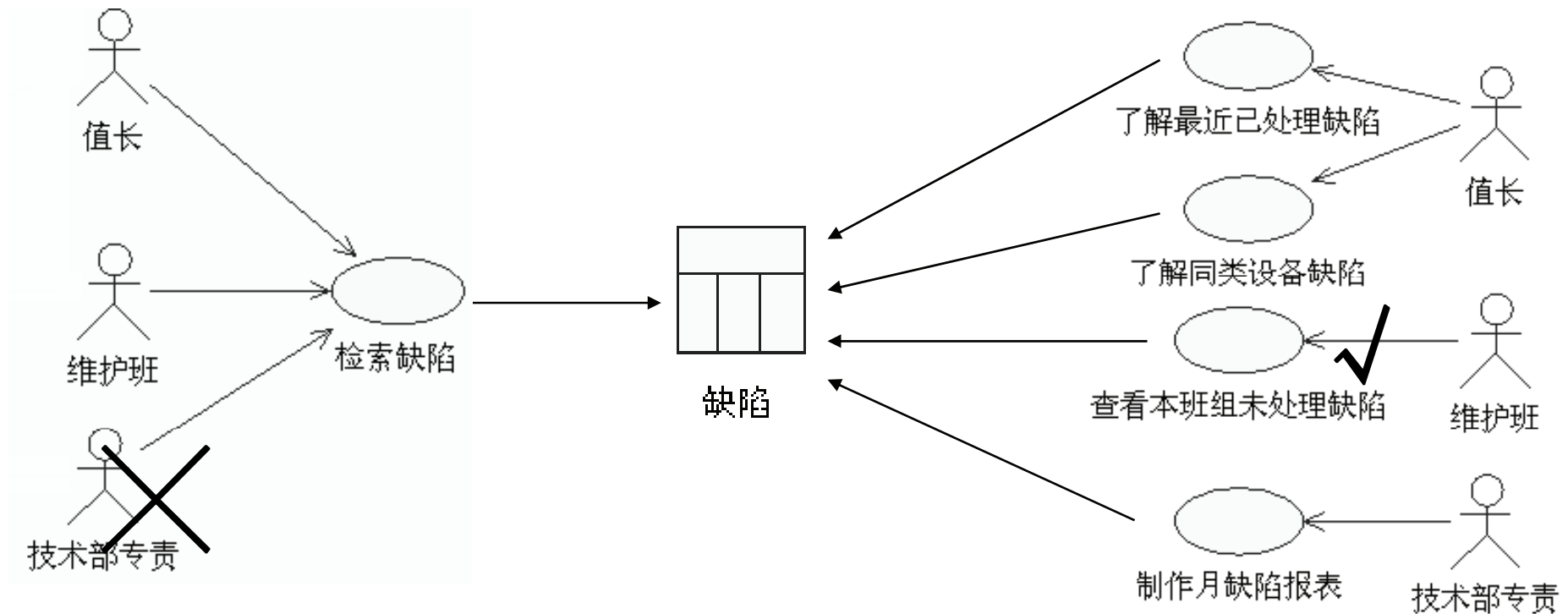


一个用例背后可能隐藏着许多数据操作

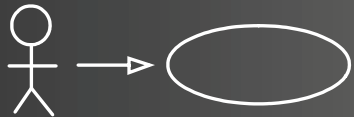


系统用例

——“粒度”：四轮马车



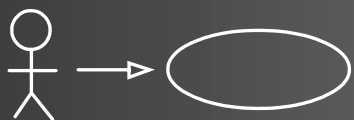
多个用例会操作同一项数据



系统用例



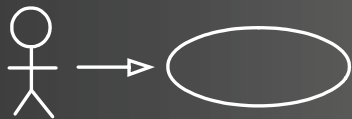
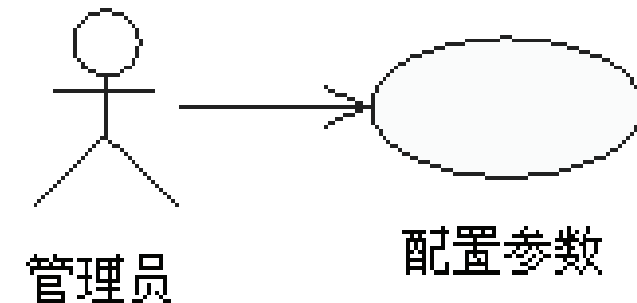
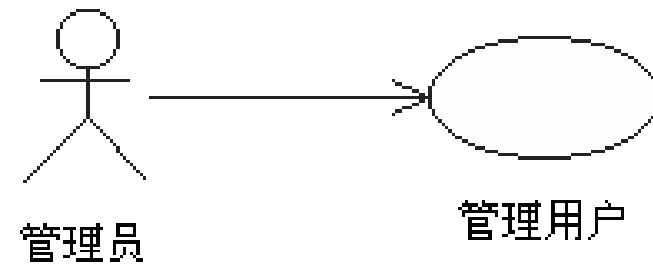
需求是不“复用”的，“复用”的是设计



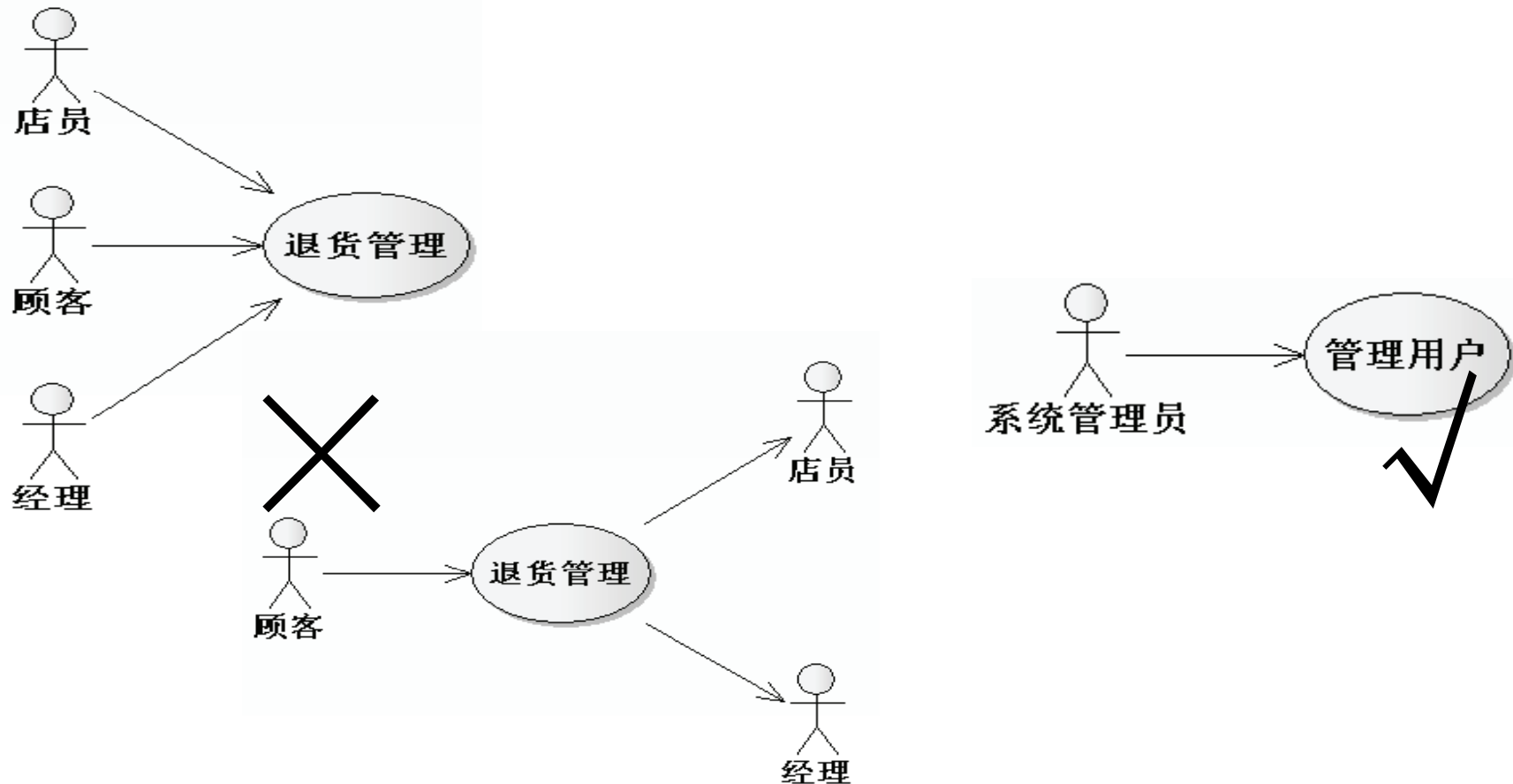
系统用例

——“粒度”：纯CRUD

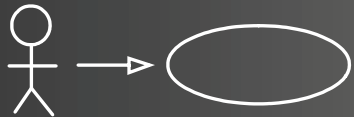
- ❖ 如果CRUD不涉及复杂的交互，一个用例“管理××”即可
- ❖ 不管是C、R、U、D，都是为了完成“管理”的目标
- ❖ 甚至很多种基本数据的管理都可以用一个用例表示



系统用例

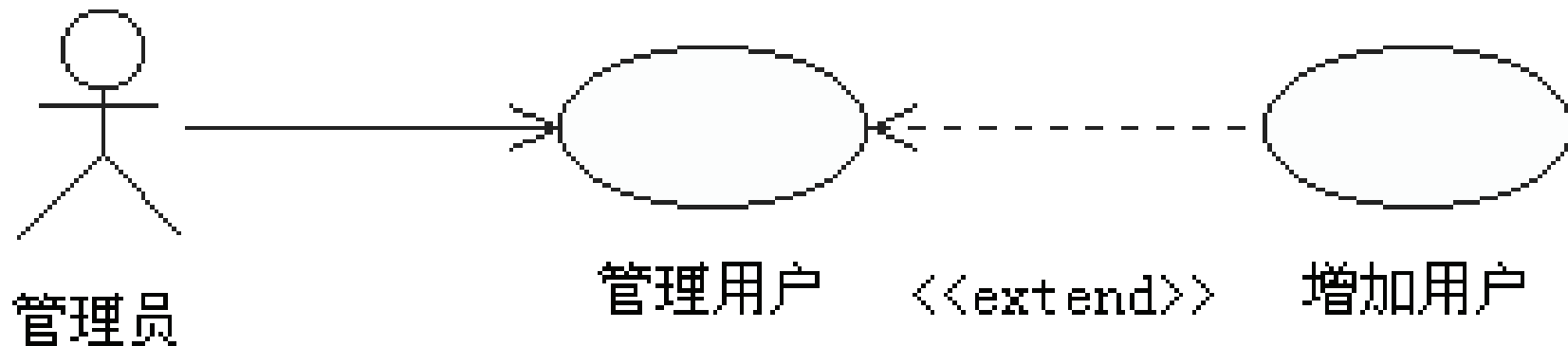


“管理××” 仅限于CRUD用例

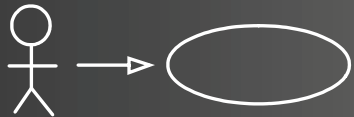


系统用例

——“粒度”：灵活处理CRUD

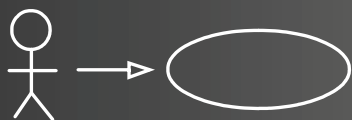


可以把包含复杂交互的路径独立出去形成用例

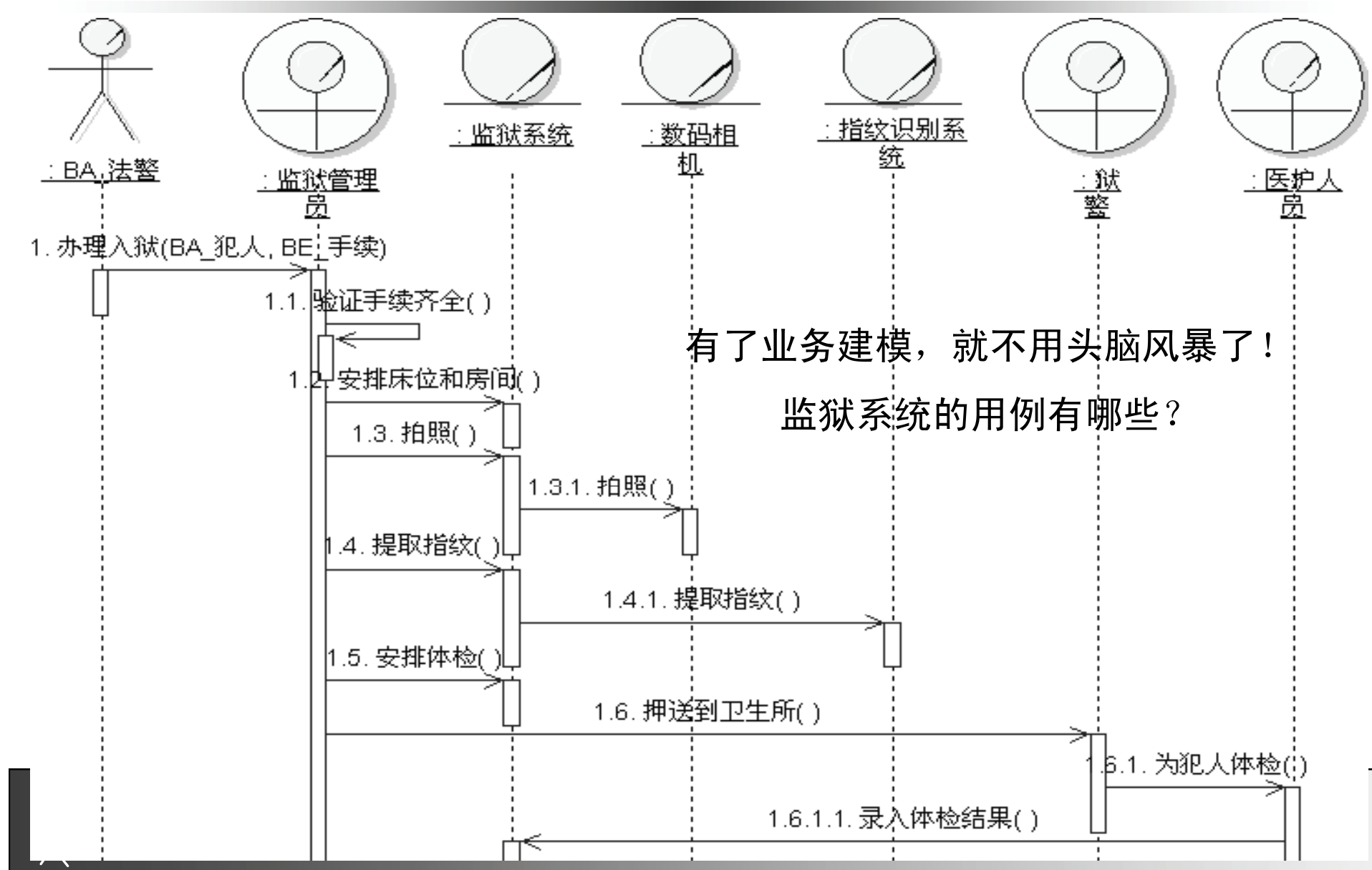


系统用例

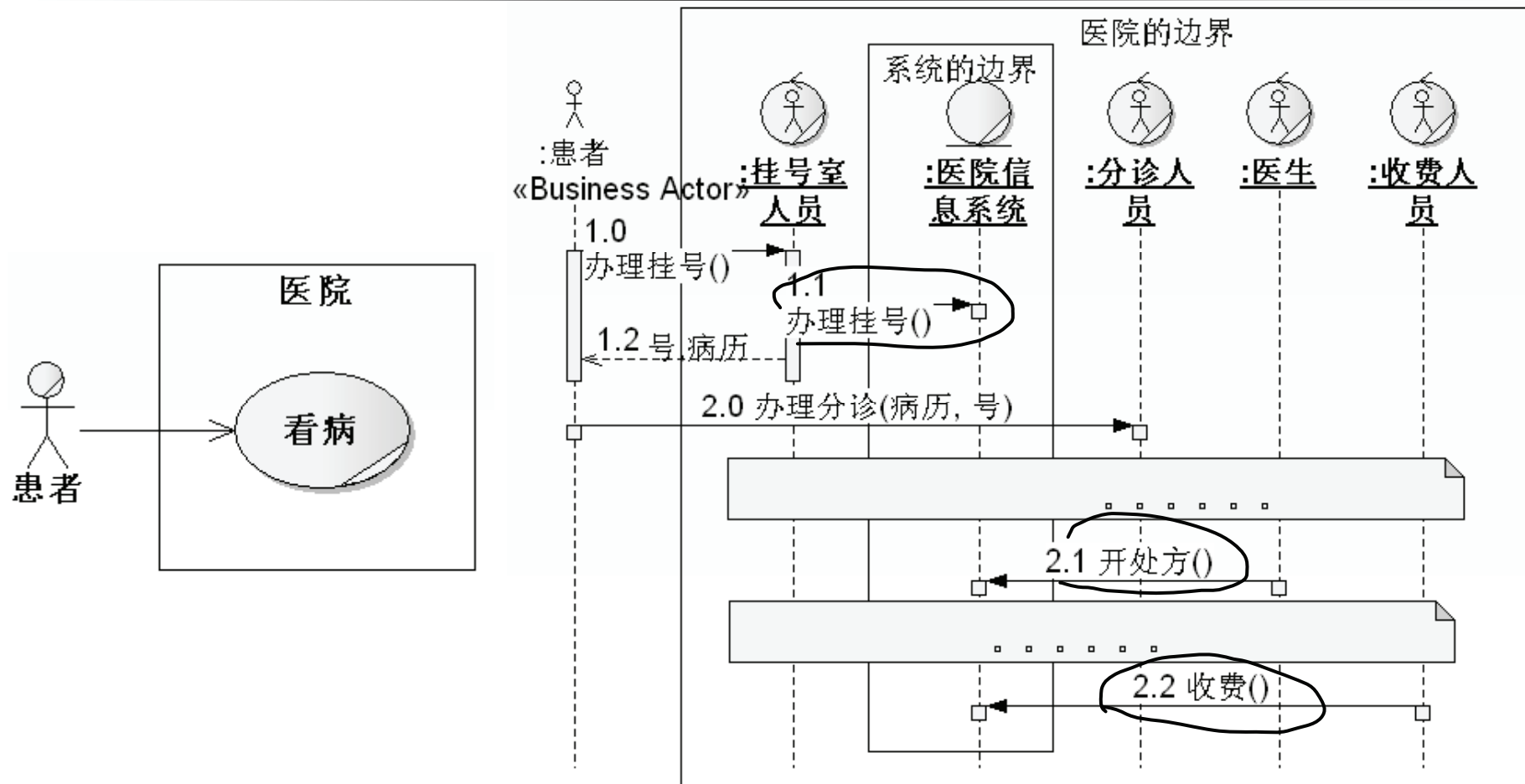
——讨论与练习、项目实作



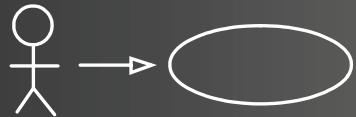
系统用例



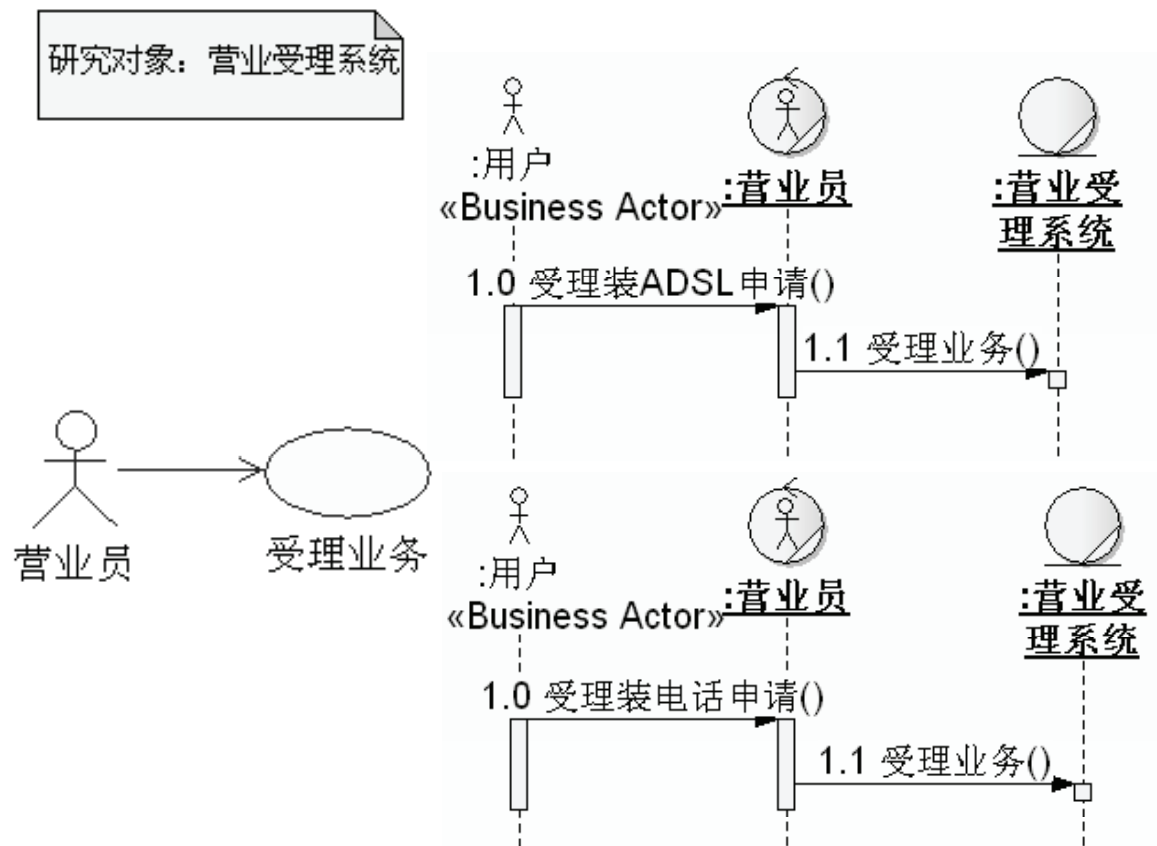
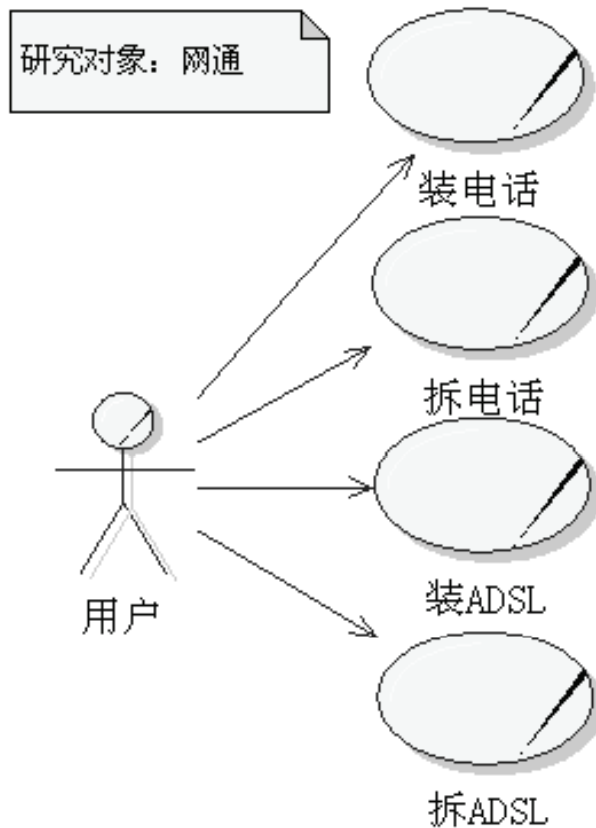
业务用例 vs. 系统用例



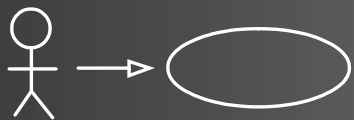
一个业务用例，多个系统用例



业务用例 vs. 系统用例



多个业务用例，一个系统用例



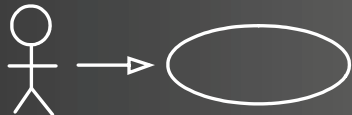
系统用例

——检查

	用例	用例	用例	用例	用例	用例	用例	用例	用例	用例
目标		●	●	●			●		●	
目标		●								
目标					●	●				
目标			●							
目标				●						

是不是所有用例都有愿景目标？

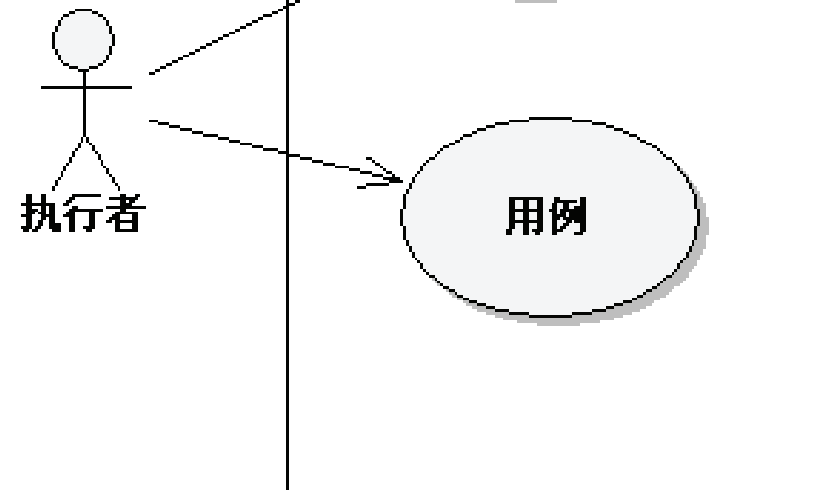
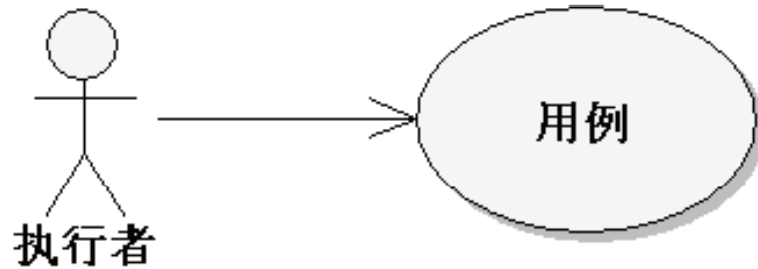
是不是所有愿景目标都有用例？



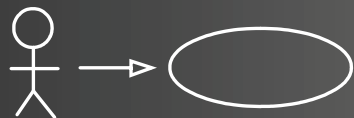
系统用例

——提示

研究对象: xxxx

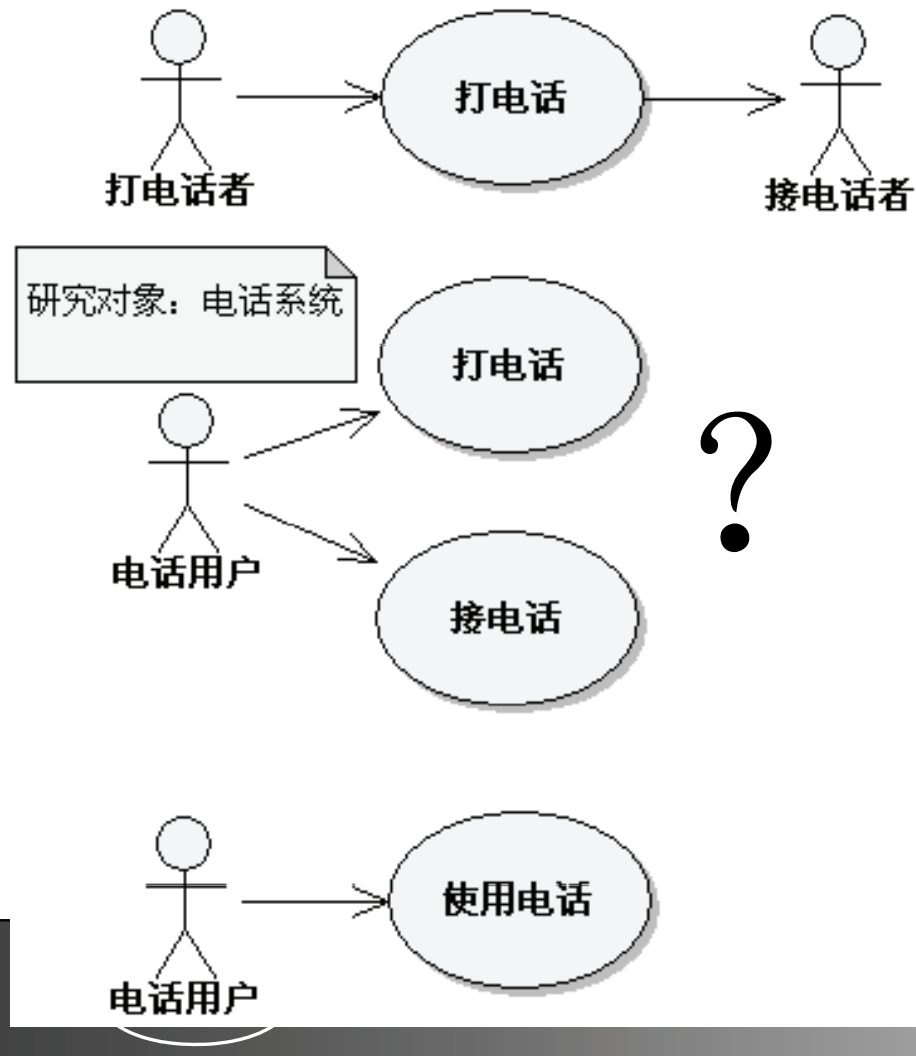


一定要说清楚研究对象



系统用例

——提示

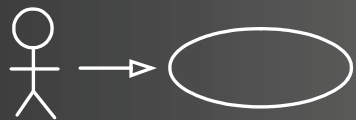
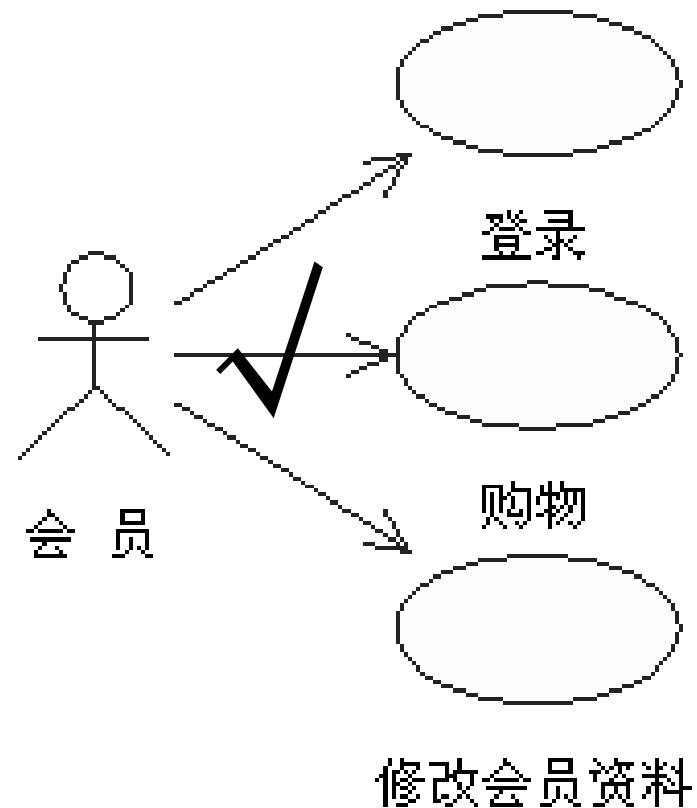
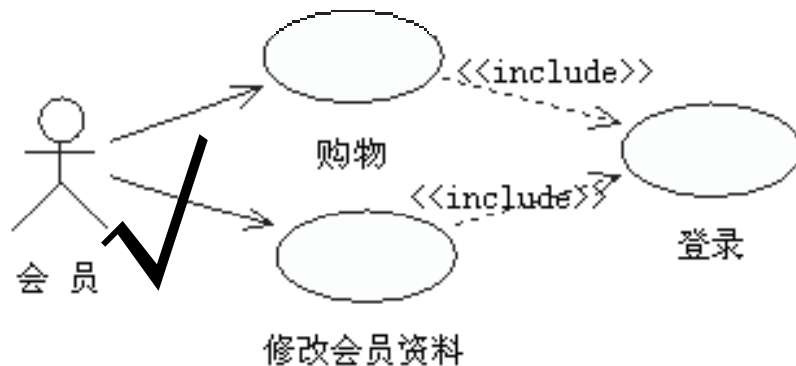
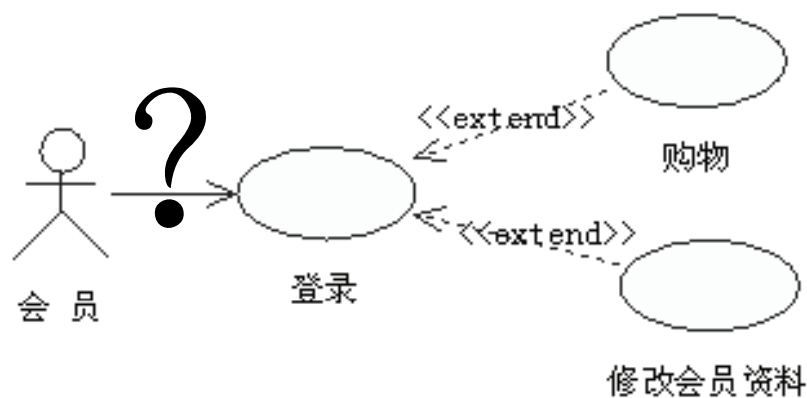


到底哪一种更合适，
完全取决于涉众的看法

不去揣摩涉众的看法
一味想找“标准答案”
无异于掩耳盗铃

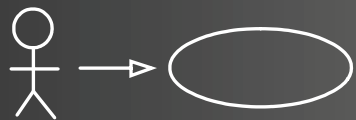
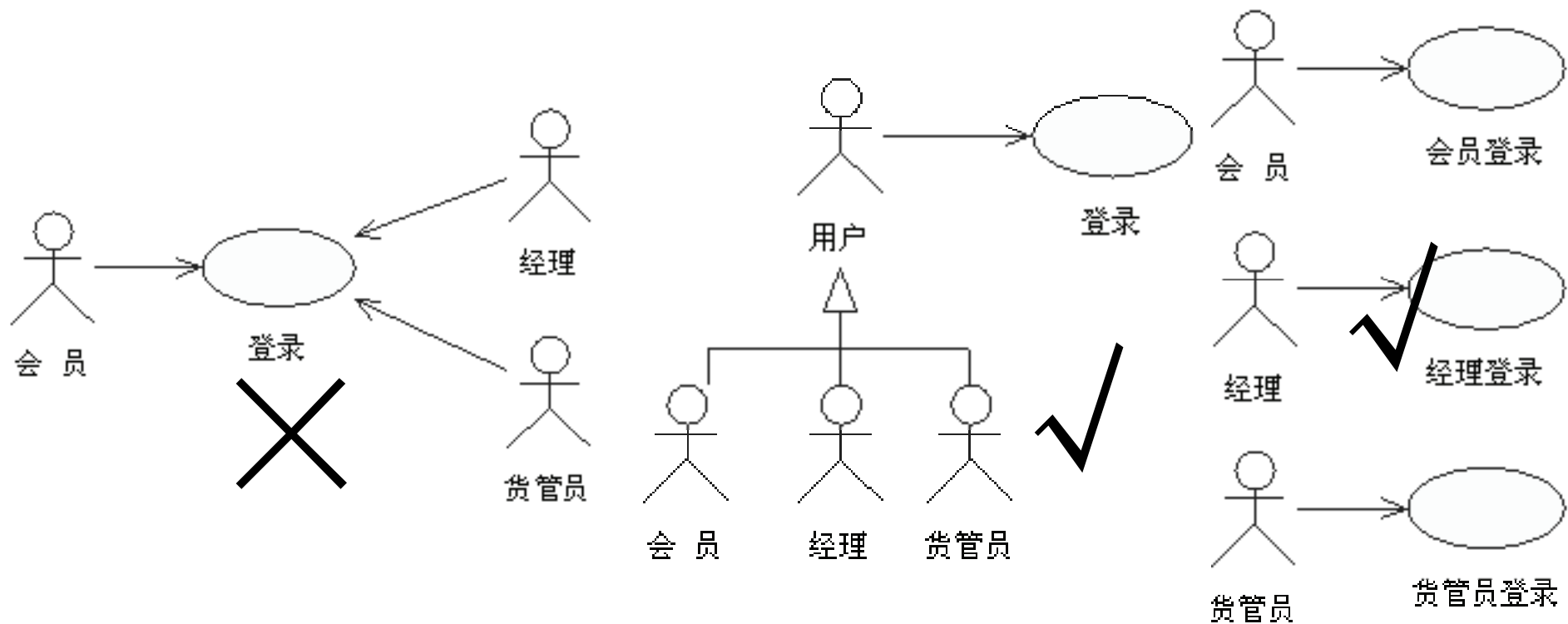
系统用例

——提示：登录怎么处理

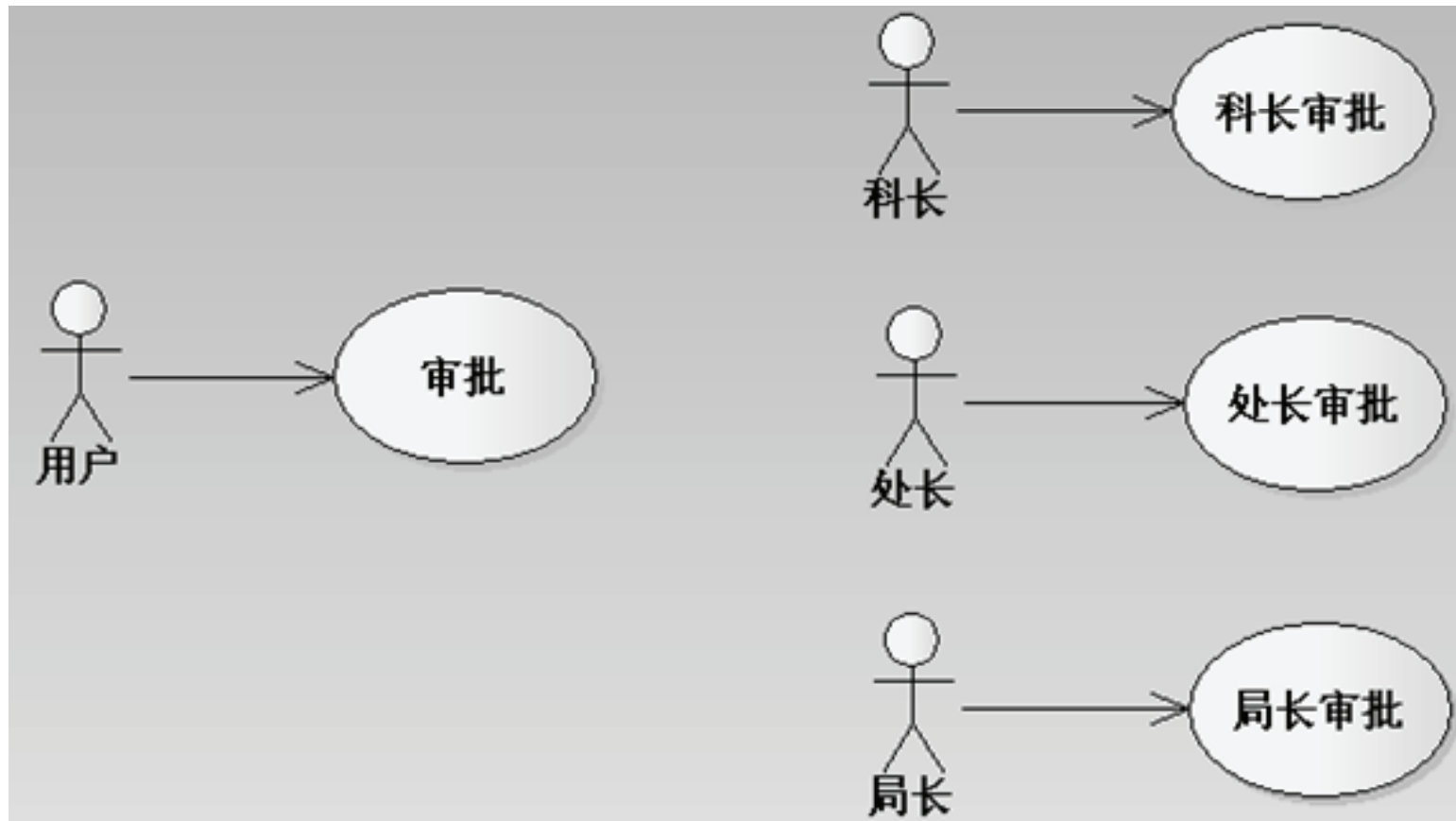


系统用例

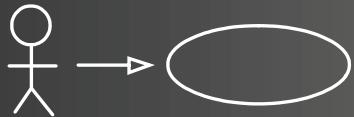
——提示：几个登录？



系统用例

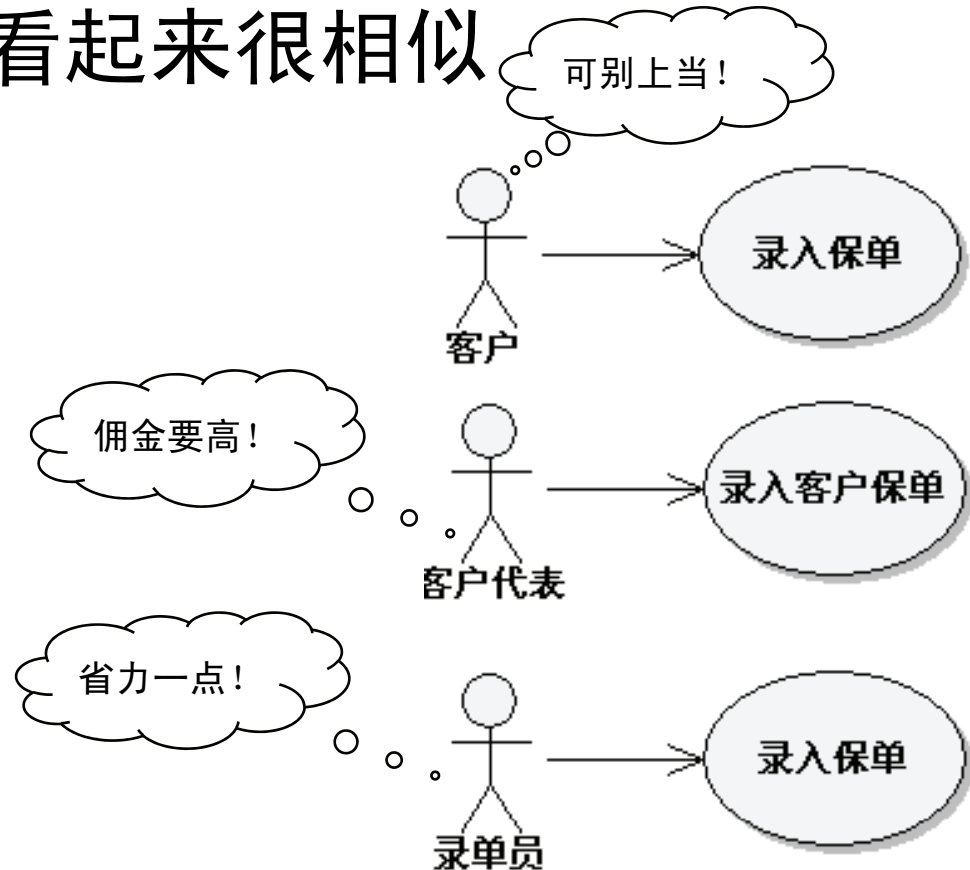
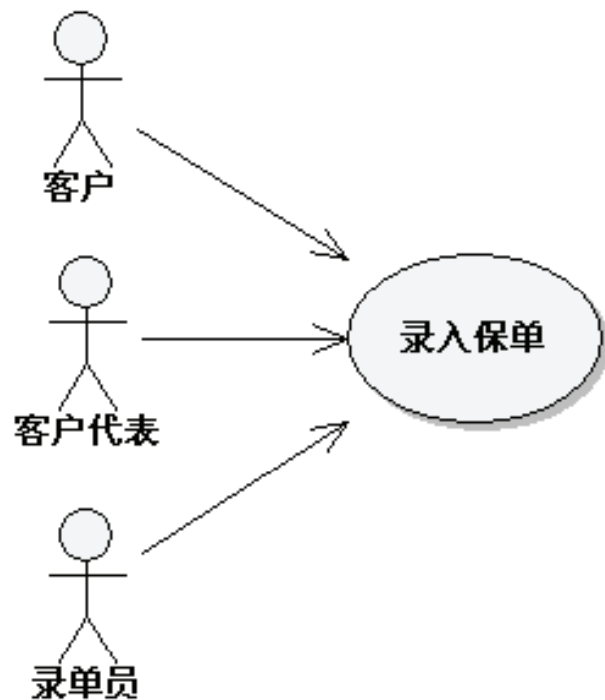


哪一个好一点？

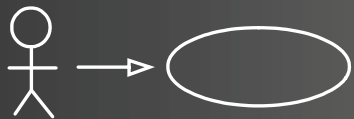


系统用例

——提示：看起来很相似



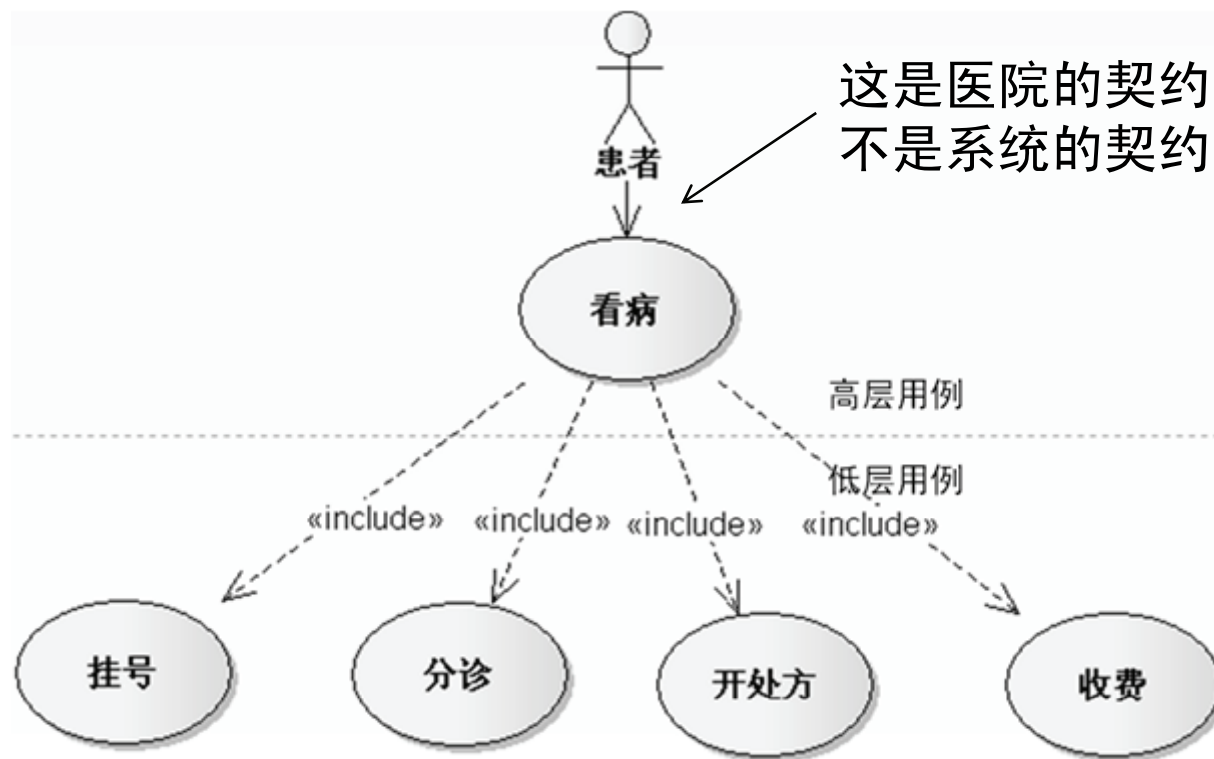
契约、价值



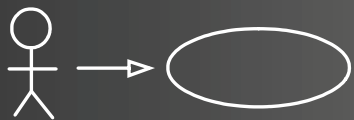
系统用例



——“高层”用例



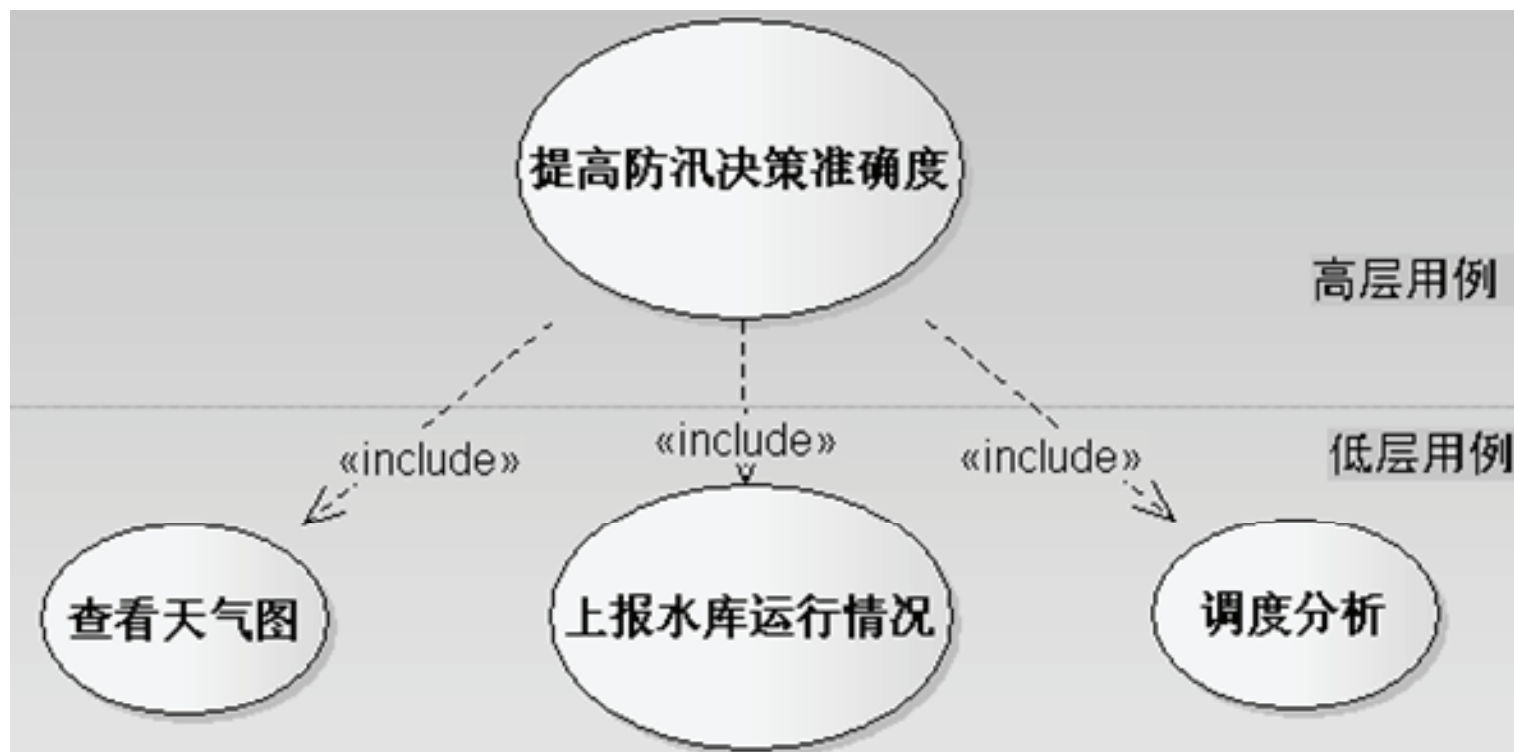
错误根源——偷换了研究对象



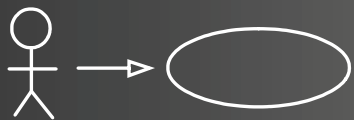
系统用例



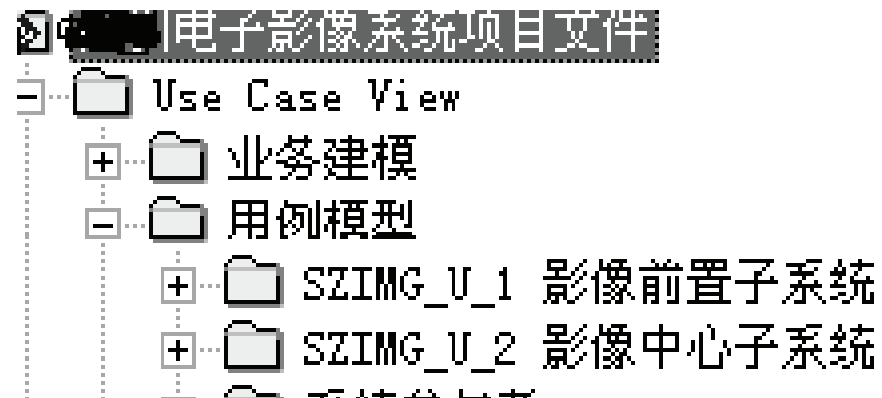
——“高层”用例



错误根源——非契约当成契约



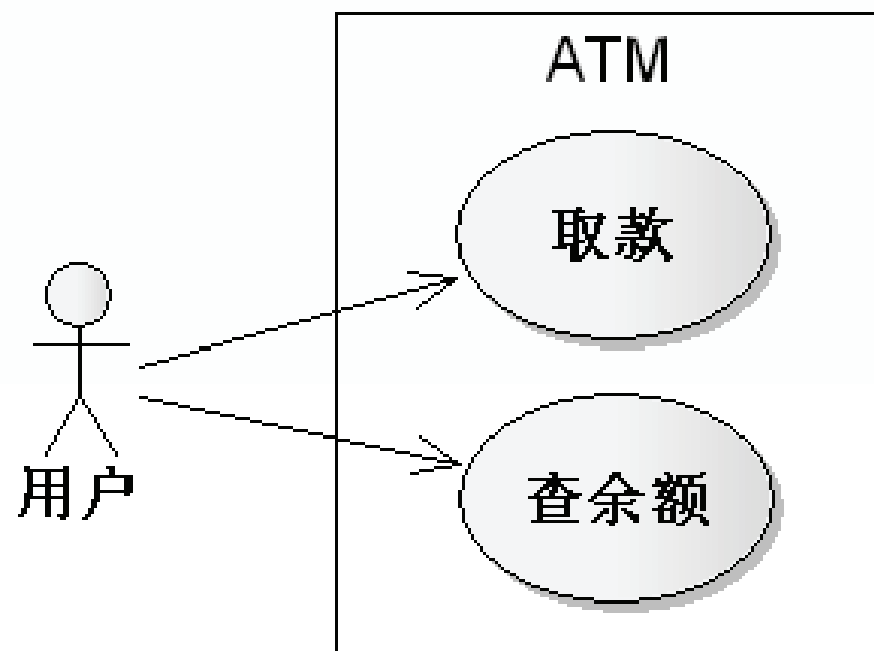
系统用例



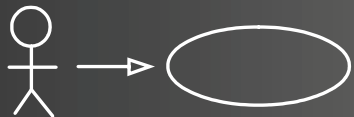
不存在“子系统”的用例



系统用例



大用例无妨小用例



核心 workflow

*愿景

*业务建模

选定愿景要改进的业务组织

业务用例图

现状业务序列图

改进业务序列图

*需求

系统用例图

书写用例文档

提升
销售

*分析

类图

序列图

状态图

*设计

建立数据层

精化业务层

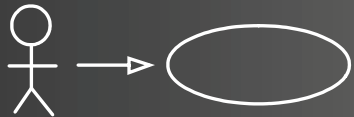
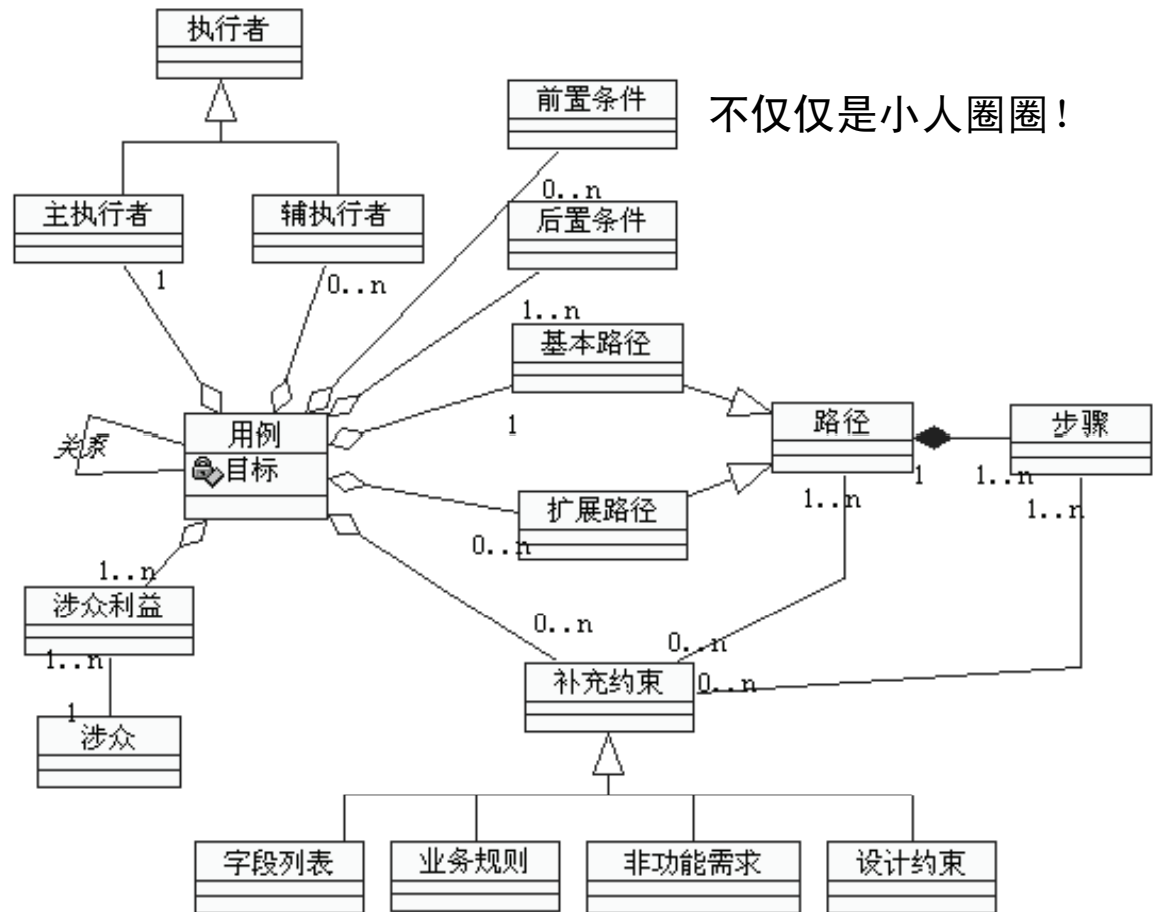
精化表示层

降低
成本



书写用例文档

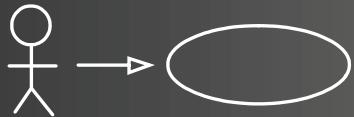
- ❖ 用例编号：用例名
- ❖ 执行者
- ❖ 前置条件
- ❖ 后置条件
- ❖ 涉众利益
- ❖ 基本路径
 - ❖ 1..... × × × × ×
 - ❖ 2..... × × × × ×
 - ❖ 3..... × × × × ×
- ❖ 扩展
 - ❖ 2a. × × × × ×:
 - ❖ 2a1.... × × × × × ×
- ❖ 字段列表
- ❖ 业务规则
- ❖ 非功能需求
- ❖ 设计约束



书写用例文档

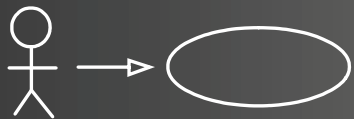
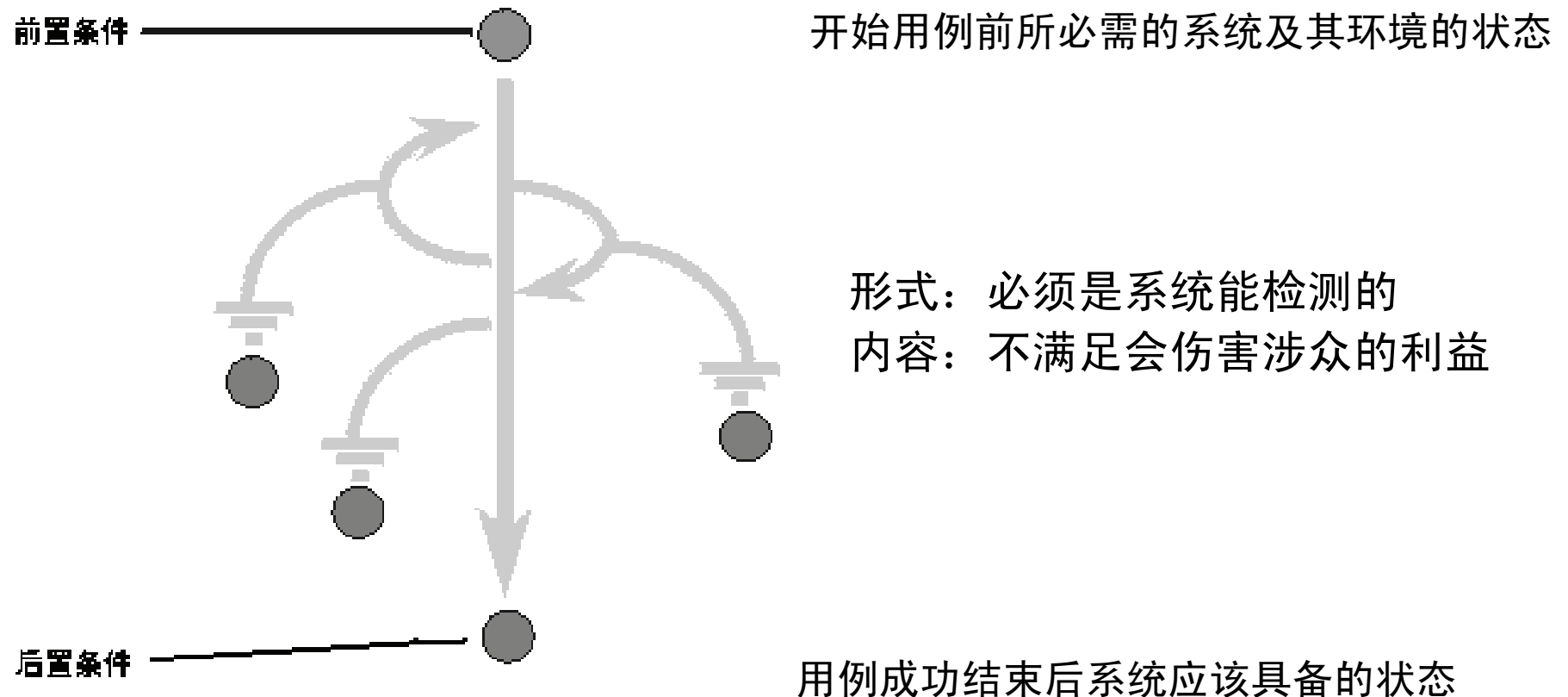
如果涉众不能理解和验证，它就不是需求
(如果删掉它，会不会有涉众的正当权益受侵害?)

总原则



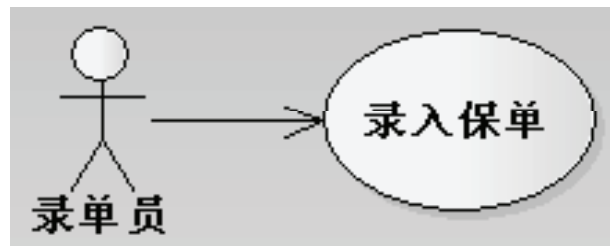
书写用例文档

——前置、后置条件



书写用例文档

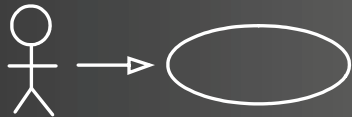
——前置、后置条件



业务代表已把保单交给录单员 ~~✗~~

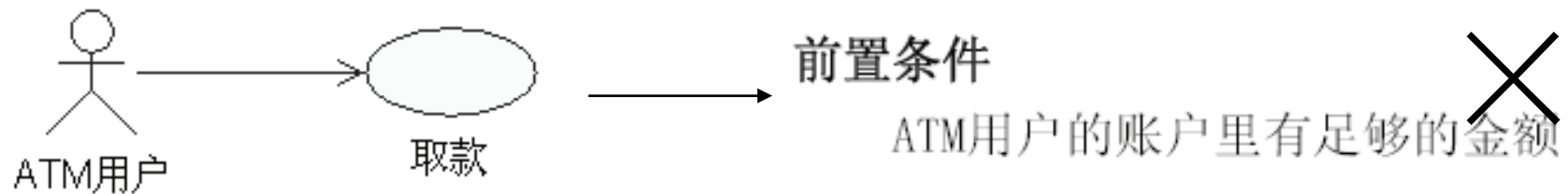
录单员已经登录 ✓

必须是系统能检测到的

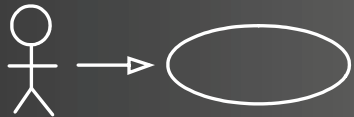


书写用例文档

——前置、后置条件



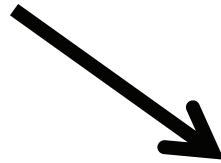
前置条件必须是系统在用例开始前能检测到的



书写用例文档

后置条件

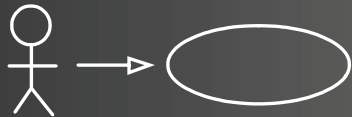
→ 系统记录痕检员的鉴定结果



后置条件

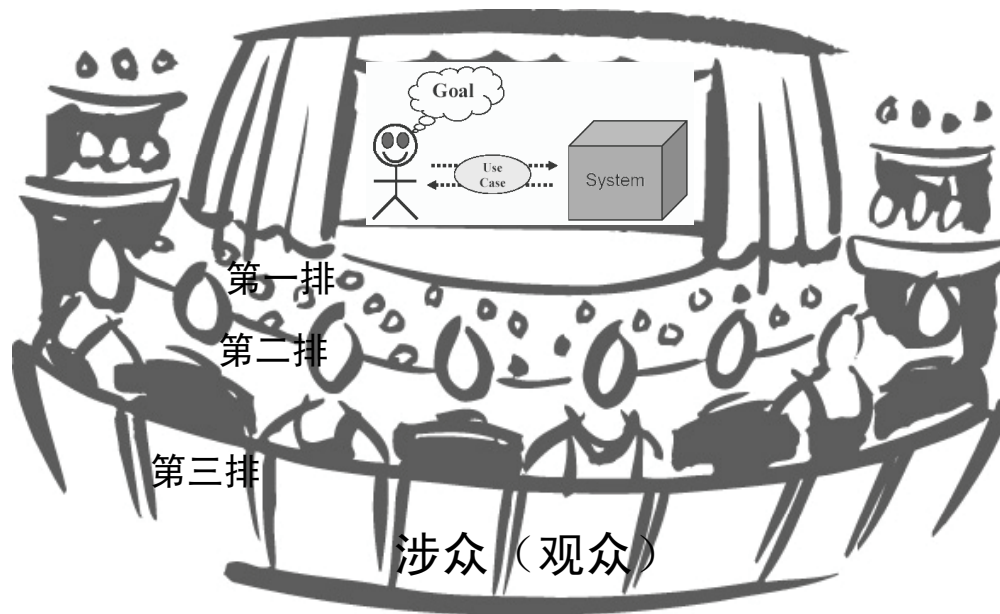
→ 系统已记录痕检员的鉴定结果

前置后置条件



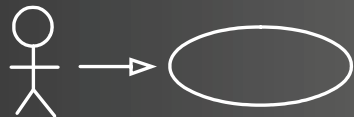
书写用例文档

——涉众利益



西门庆和潘金莲？

投资人、导演、演员、观众...



书写用例文档

——涉众利益



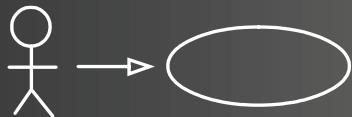
同样是“取钱”：

为什么家里的抽屉不用密码，取款机要用？

为什么银行不放一个箩筐在门口？

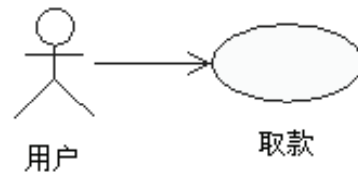
为什么取了钱以后要写“系统扣除账户金额”？

同样的目标，不同的过程...



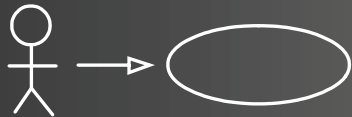
书写用例文档

——涉众利益



用户: 希望方便
银行: 希望安全
法律: 保护财产

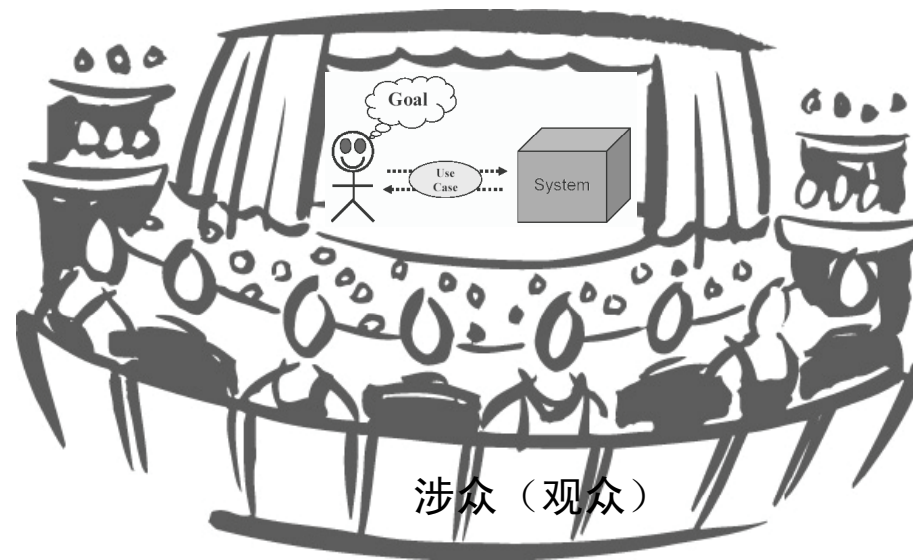
用例平衡涉众之间的利益



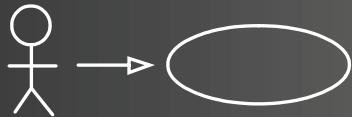
用例

——进阶定义

❖ Cockburn: 用例是涉众之间达成的契约，以执行者为达成特定目标和系统交互的方式演绎



达到“契约”的理解层次，才能保证用例内容的正确



书写用例文档

——涉众利益

涉众利益

用户——希望方便

银行——希望安全；希望节约运营成本。

法律——保护个人财产

利益的冲突

基本路径

1. 用户插入 ATM 卡

2. 系统要求输入合法的密码

3. 用户输入密码

4. 系统验证密码合法 正确

5. 系统提示用户输入取款金额

6. 用户输入取款金额并确认

7. 系统检测取款金额合法

8. 系统从帐户扣除取款金额

9. 系统吐钞

10. 系统提示用户“打印收据”或者“不打印收据”

11. 用户要求不打印收据

12. 系统显示“交易结束”，退卡

银行的

用户的

法律的

谁的？

业务规则

4. 密码为 6 位数字

7. 金额必须为 100 元的倍数

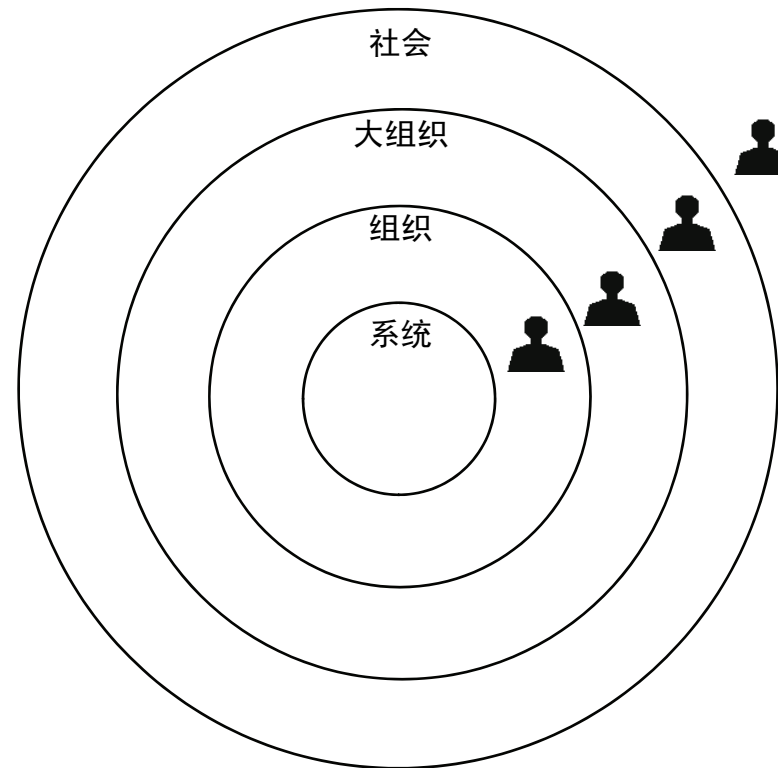
7. 当日取款金额不能超过 5000 元



书写用例文档

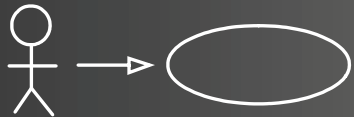
——涉众利益

- ✦最终用户
- ✦客户
- ✦政府
- ✦法律
- ✦文化
- ✦开发人员
- ✦管理人员
- ✦竞争对手
- ✦...



- ✦获得权力
- ✦失去权力
- ✦获得金钱
- ✦失去金钱
- ✦...

谁关心这个系统？会涉及到他的什么利益？



书写用例文档

——涉众利益

优秀的 VC++ 软件工程师,
Autodesk 为你提供展示才华的机会

Autodesk®

“英特尔杯”

请您选择一种注册方式：



短信注册 hot

联通和移动手机用户发短信至指定客服号码注册。
该方式注册用户自动激活，无须填写EMAIL地址，可以直接修改用户资料。方便快捷，推荐使用。

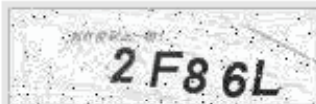


普通注册

通过网络填写注册表单注册。注意：邮箱必须填写正确，你将通过填写的邮箱来收到激活码，未激活帐户无法使用CSDN的各项功能！

注册条款：☒ 我已仔细阅读并接受CSDN的注册条款。 [CSDN注册条款](#)

校验码 *



输入校验码 *

用户登录昵称 *

检查用户是否存在

密码 *

重复输入密码 *



lchina.com

书写用例文档

——涉众利益

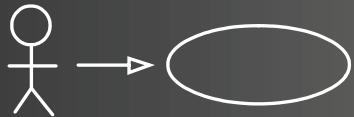


醉酒法

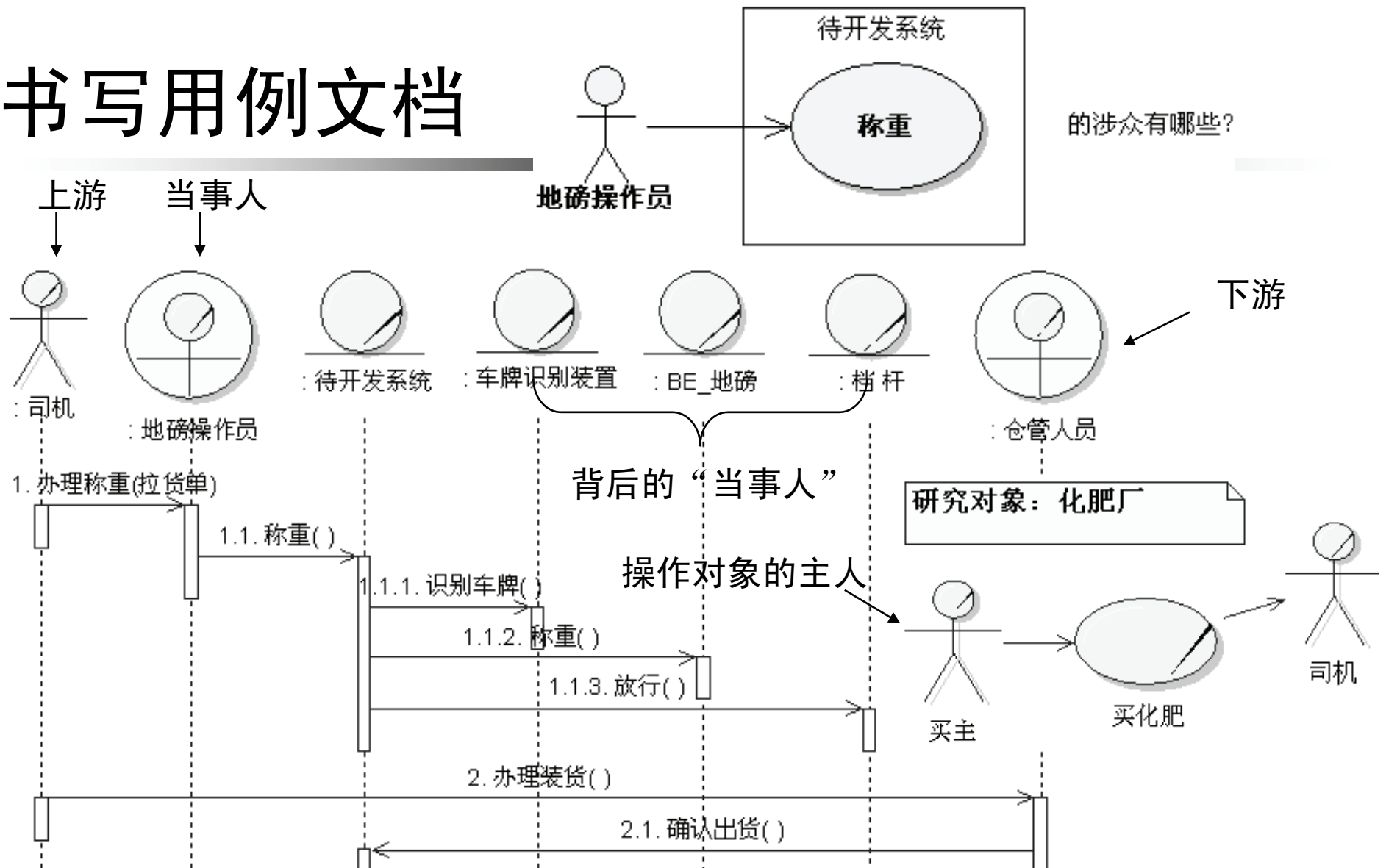


公安法

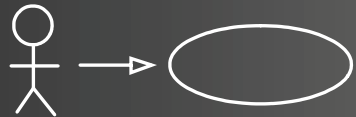
- ❖ 当事人
- ❖ 上游
- ❖ 下游
- ❖ 操作对象的主人
- ❖ ...



书写用例文档



业务建模对识别涉众的作用





书写用例文档

化肥公司老板（1）——希望杜绝老鼠
开发公司（2）——希望开发成本要低
地磅工作人员（3）——担心承担责任，担心系统坏掉；
希望操作简便
仓库装货员（3）——担心称不准导致返工
货主（4）——担心空车时称轻了，出来时称重了；
司机（5）——担心等候时间太长；

涉众 “各怀鬼胎”



书写用例文档

领导

客户

开发公司老板

用户

.....

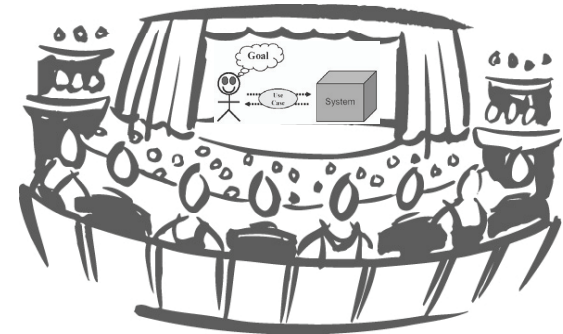
开发人员

前排涉众

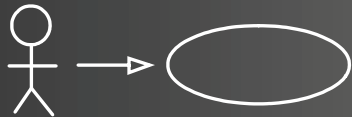
越是一线用户，排位越低

最后排涉众

开发人员花在前排涉众身上的时间往往不够！

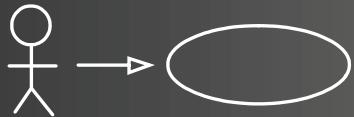
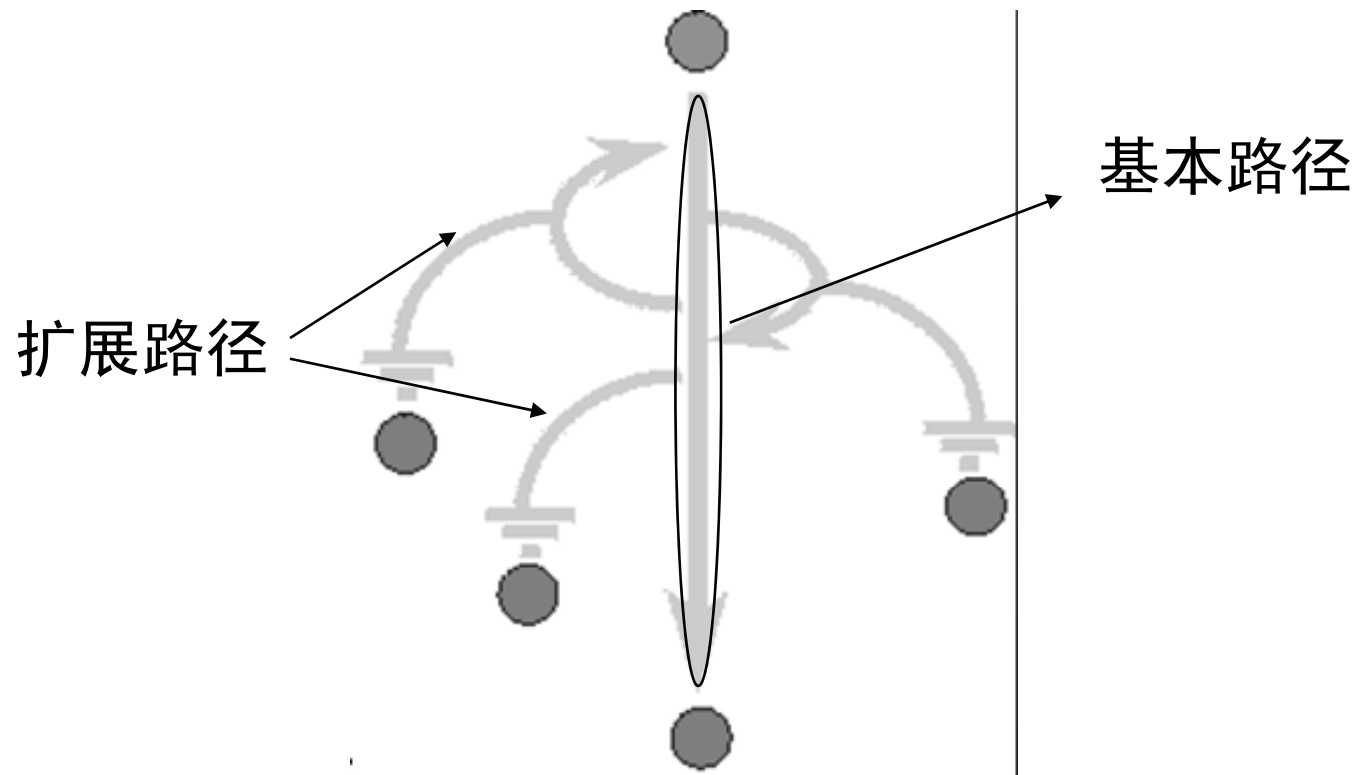


涉众其实指前排涉众



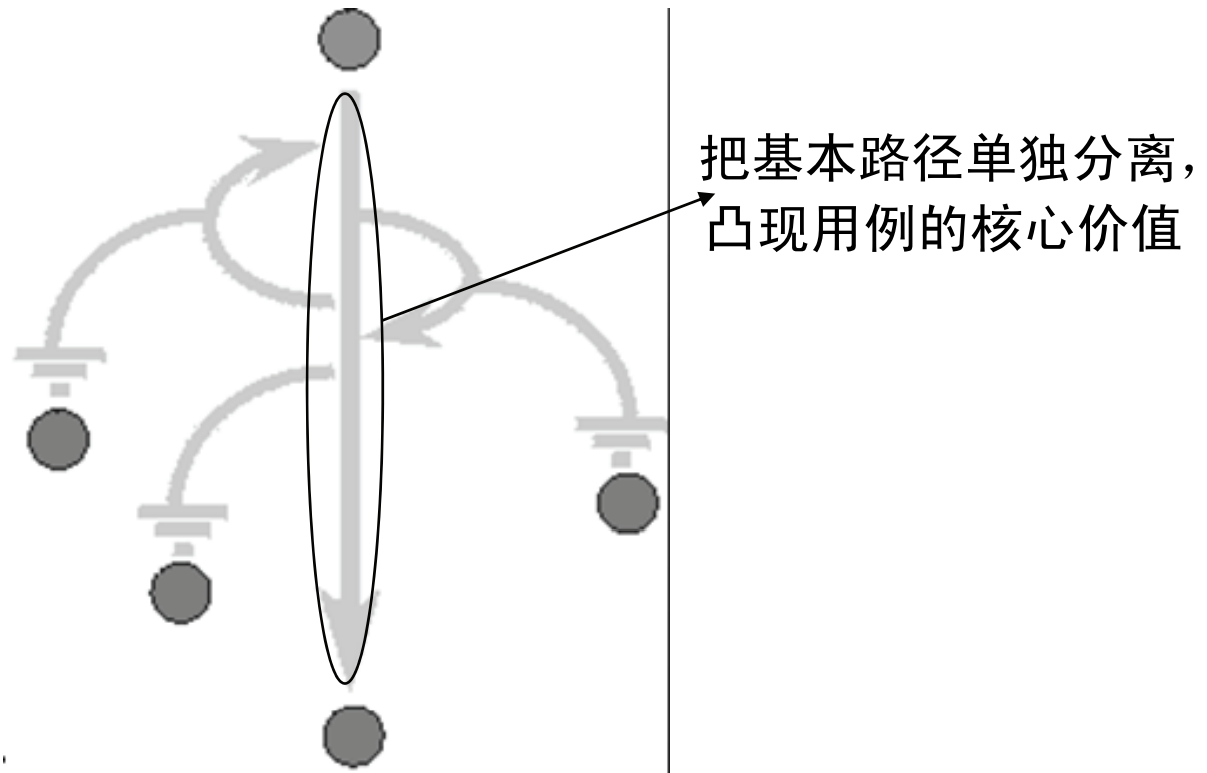
书写用例文档

——路径步骤

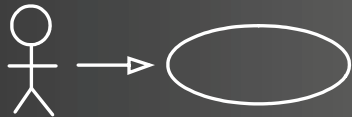


书写用例文档

——路径步骤：基本路径

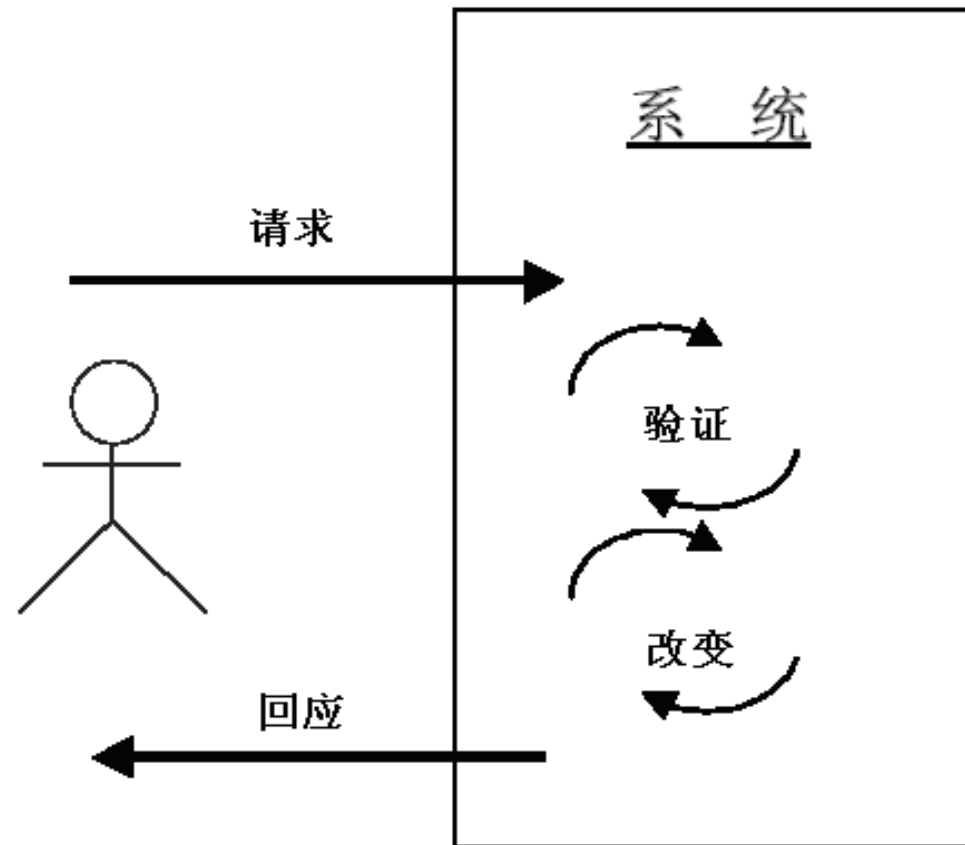


客户最想看到、最关心的路径

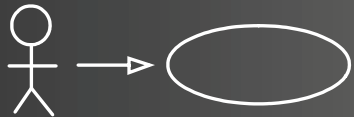


书写用例文档

——路径步骤



交互四步曲



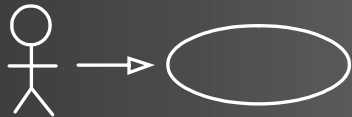
书写用例文档

——路径步骤

什么时候需求
里可以出现SQL字样?

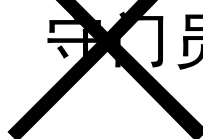

- ❖ 系统建立连接，打开连接，执行SQL语句，从“零件”表查询...
- ❖ 系统按照查询条件搜索零件

可理解可验证（说人话）



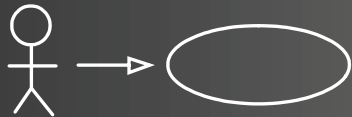
书写用例文档

——路径步骤

- ❖ 德罗巴从兰帕德处得到传球，守门员... 
- ❖ 兰帕德传球，德罗巴射门，守门员扑救.... 



使用主动语句——理清责任

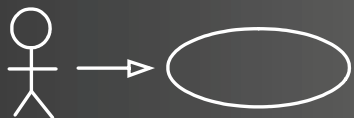


书写用例文档

——路径步骤

- ❖ 系统从会员处获取用户名和密码
- ❖ 会员提交用户名和密码
- ❖ 用户名和密码被验证
- ❖ 系统验证用户名和密码

主动语句

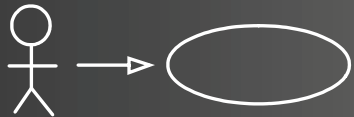


书写用例文档

——路径步骤

- ❖ 执行者 × × × × ×
- ❖ 系统 × × × × ×
- ❖ 系统 × × × × ×
- ❖ 执行者 × × × × ×

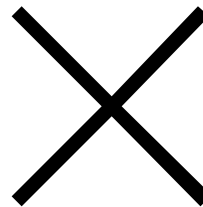
主语只能是执行者或系统



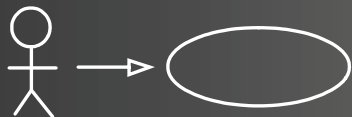
书写用例文档

——路径步骤

- ❖ 执行者填写用户名
- ❖ 系统验证用户名未被使用
- ❖ 执行者填写密码
- ❖ 系统验证密码合法
- ❖ ×××××



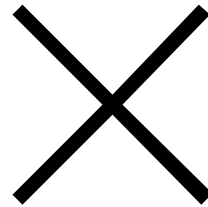
聚焦于输入输出处理



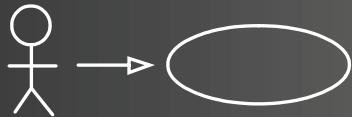
书写用例文档

——路径步骤

- ❖ 执行者填写姓名
- ❖ 执行者填写电话
- ❖ 执行者填写联系地址
- ❖ 执行者提交
- ❖ ×××××



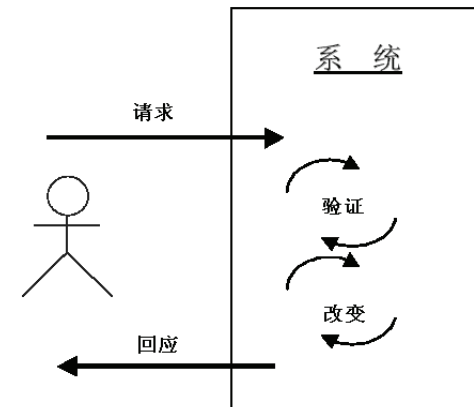
遵循四步曲



书写用例文档

基本路径：↵

- 1、→痕检员录入案件描述信息↵
- 2、→痕检员用扫描仪从指纹卡采集指纹图像↵
- 3、→痕检员编辑指纹特征信息↵
- 4、→痕检员提交查找任务↵
- 5、→系统将案件信息发送给 AFIS 系统进行查找↵
- 6、→AFIS 系统返回比中结果↵

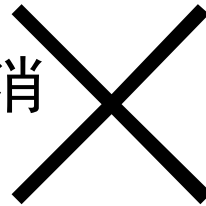


请求、（验证）、（改变）、回应

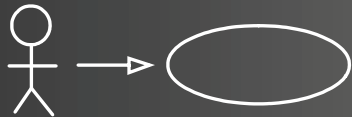


书写用例文档

1. 系统显示订单列表
2. 会员选择订单，请求取消
3.



请求、（验证）、（改变）、回应



书写用例文档

——路径步骤

基本路径

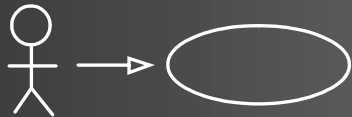
1. 潜在会员请求注册。
2. 系统显示注册界面。
3. 潜在会员提交注册信息。
4. 系统验证注册信息充分。
5. 系统生成用户名和密码，保

没有“如果”

扩展

- 4a. 信息不充分：
 - 4a1. 系统提示输入剩余信息

分支：基本和扩展分开

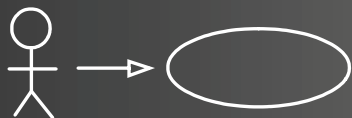


书写用例文档

基本路径

1. 群众请求评分。
 2. 系统显示同组被考核对象。
 3. 群众选择被考核对象，输入评分结果，请求保存。
 4. 系统验证对该考核对象已经输完所有评分结果。
 5. 系统保存对该考核对象的评分结果。
- *3-5 循环，直到全部被考核对象都已评分。

循环

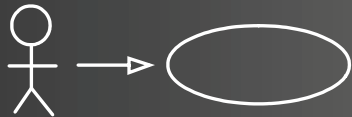


书写用例文档

——路径步骤

- ❖ 会员从下拉框中选择类别
- ❖ 会员在相应文本框中输入查询条件
- ❖ 会员点击“确定”按钮
- ❖

不要涉及界面组件



书写用例文档

低精度，稳定

❖ 用例（取款）

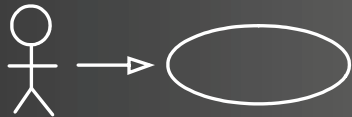
❖ 路径（正常取款）

❖ 步骤（系统验证取款金额合法）

❖ 补充约束（取款金额必须为100元的倍数）

高精度，不稳定

不同级别的需求





书写用例文档

3e. 收银

3e1. 系统收集客层分析数据。

3e2. 系统显示收银界面。

3e3. 系统计算并显示应收总金额。

~~3e4. 顾客付款。~~

3e5. 用户录入付款方式和金额。

3e6. 系统计算并找零。

3e7. 系统打开钱箱并打印小票。

~~3e8. 用户找零~~并关闭钱箱。

不要假想系统不能负责的事情



书写用例文档

——路径步骤

❖ ○ ○ ○ ○ ○ ○

❖ 经理审查支票

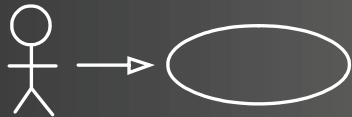
❖ 经理确认支票通过审查

❖ ○ ○ ○ ○ ○ ○

牢记四步曲
即可避免

模糊了系统的契约
抹杀进一步改进的可能

不要越界——假想系统不能负责的事情



书写用例文档

基本路径: ↵

- 1、→痕检员录入案件描述信息↵
- 2、→痕检员用扫描仪从指纹卡采集指纹图像↵
- 3、→痕检员编辑指纹特征信息↵
- 4、→痕检员提交查找任务↵
- 5、→系统将案件信息发送给 AFIS 系统进行查找↵
- 6、→AFIS 系统返回比中结果↵



辅助执行者

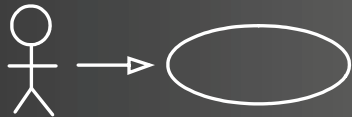


书写用例文档

系统请求 × × 做某事 (不用写 × × 系统做某事)

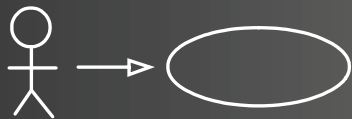
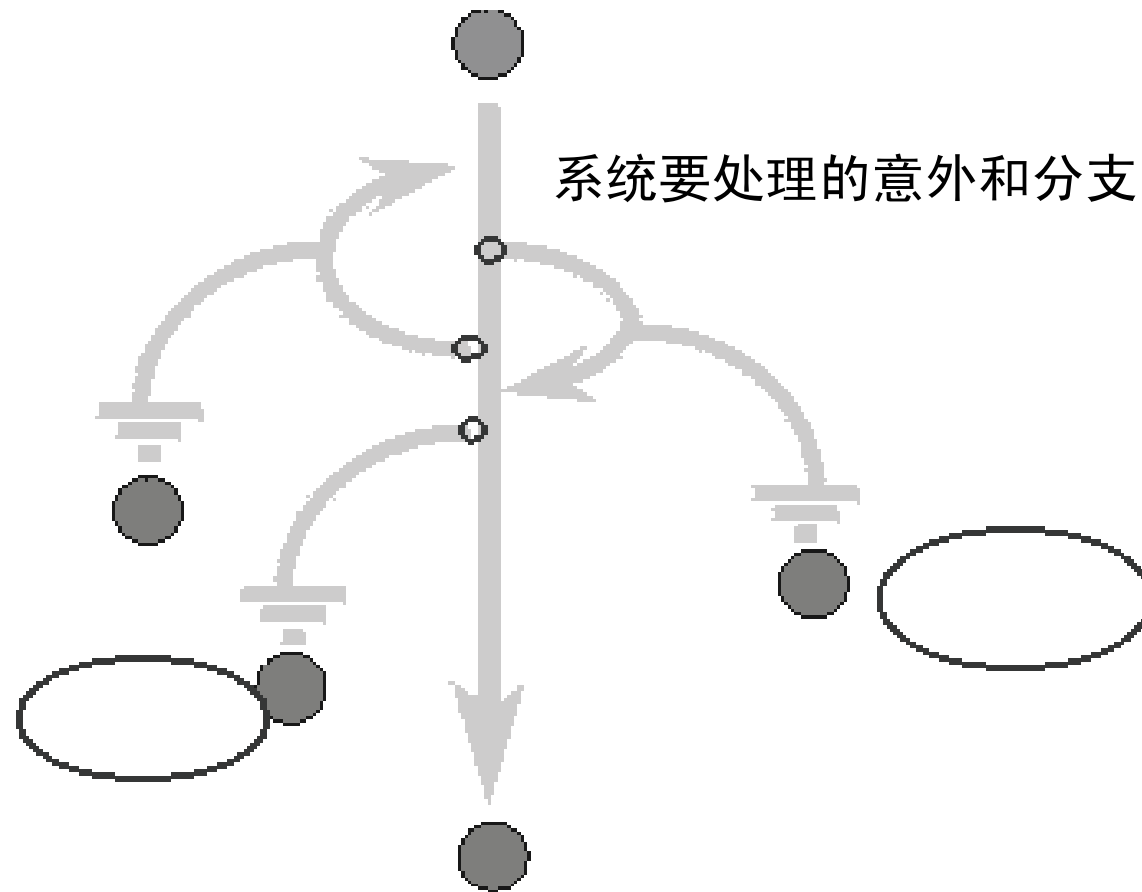
- 3、→痕检员编辑提交指纹特征信息、案件描述信息。
- 4、→痕检员提交查找任务。
- 5、→系统将案件信息发送给请求 AFIS 系统进行查找比对。

辅助执行者



书写用例文档

——路径步骤：扩展



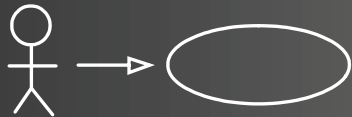
书写用例文档

——路径步骤：扩展

- ❖ 执行者的选择
- ❖ 系统验证
- ❖ 步骤失败
- ❖ ○ ○ ○ ○ ○ ○

必须是系统能
感知和要处理的

识别扩展点思路



书写用例文档

4. 用户选择如下操作：

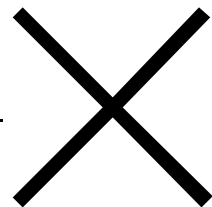
不让利

单条商品折扣

单条商品折让

削价

5. 系统计算单条商品价格。



3. 评议专家可以选择以下动作：

给出“5分”的评分

给出“4分”的评分

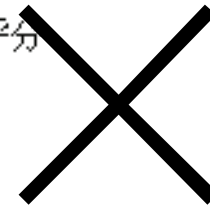
给出“3分”的评分

给出“2分”的评分

给出“1分”的评分

给出“0分”的评分

扩展：



3a. 评议专家给 5 分：

3a1. 评议专家给出“5分”的评分

3a2. 系统保存该学员本题分数为 5

扩展！=选项，看交互行为变化



书写用例文

1. 会员请求查看订单
2. 系统显示会员的订单列表
3. 会员可以选择以下动作：

取消一张订单

4. 会员请求查看某张订单
5. 系统显示该订单明细
6. 会员可以选择以下动作：
从订单中删除订单项
修改某个订单项的购买数量
修改订单的送货地址

扩展

扩展点长在执行者处

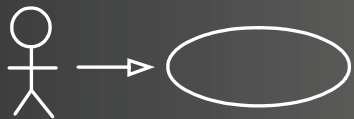
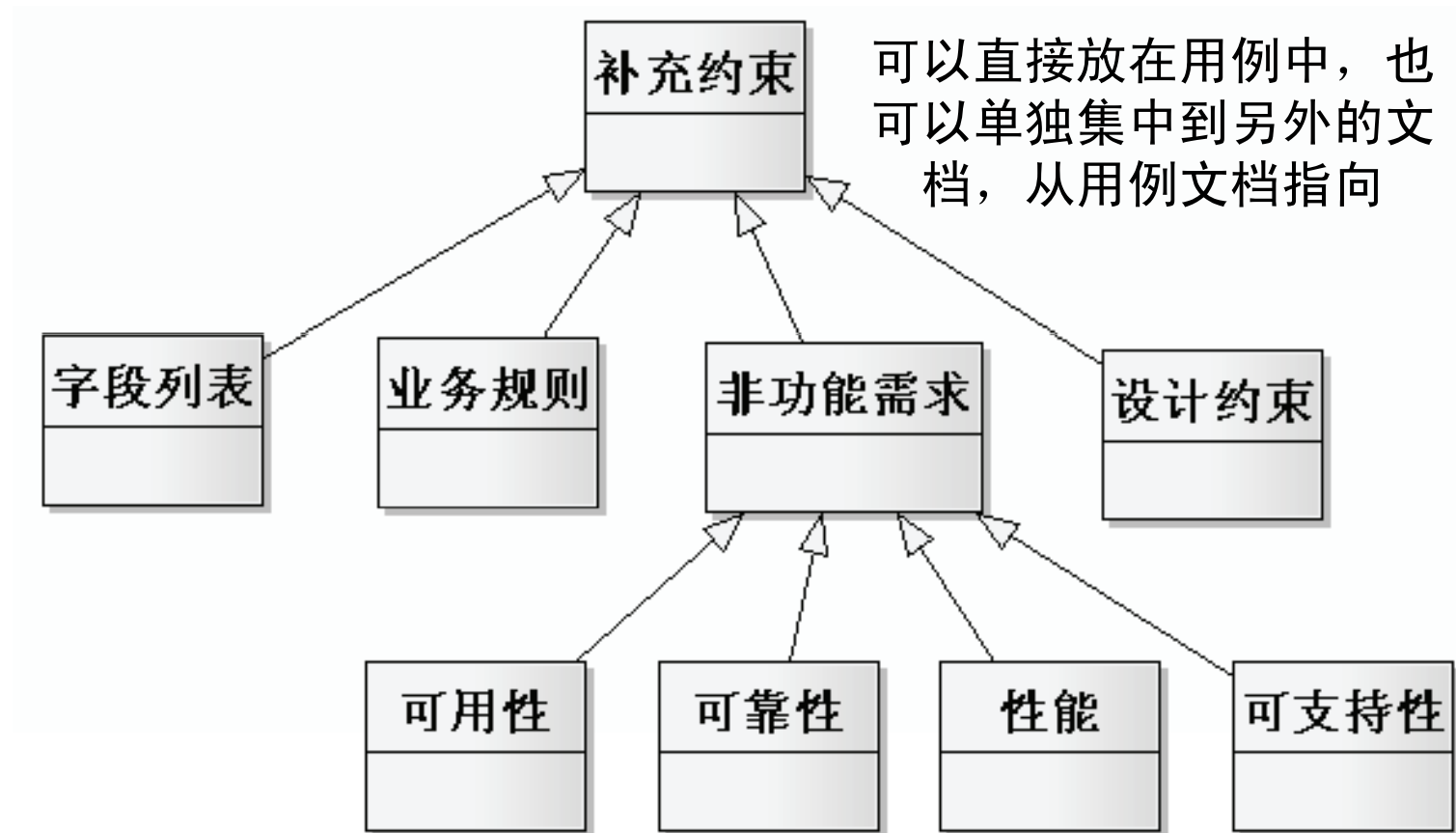
- 3a. 会员取消订单：
 - 3a1. 会员请求取消一张订单
 - 3a2. 系统删除该订单
 - 3a3. 返回 2
- 6a. 会员从订单中删除订单项：
 - 6a1. 会员请求从订单中删除某个订单项
 - 6a2. 系统删除该订单项
 - 6a3. 返回 5

CRUD用例



书写用例文档

——补充约束

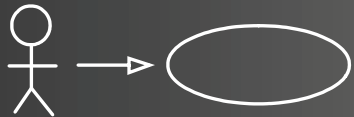


书写用例文档

——补充约束：字段列表

- ❖ + → 数据序列
- ❖ [] → 可选项
- ❖ { } * → 多个
- ❖ { | | | } → 可能取值
- ❖ A=B → 把B的结构赋给A

可以用自然语言，也可以用表达式



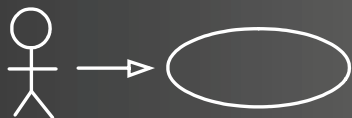
书写用例文档

——补充约束：字段列表



- ❖ 注册信息=公司名+联系人+电话+{联系地址}*
- ❖ 联系地址=州+城市+街道+邮编
- ❖ 保存信息=注册信息+注册时间
- ❖ 客房状态={空闲|已预定|占用|维修中}

用表达式表示

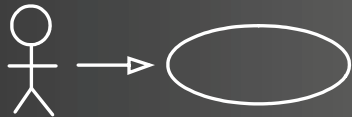


书写用例文档

——补充约束：字段列表

- ❖ 不同于数据模型——只是一部分
 - ❖ 可以用E/R图或业务对象图作为辅助说明，但不宜直接作为需求
- ❖ 不等于数据字典——容易过早把时间花在细节上
 - ❖ 一开始好像做了很多事情，其实却回避了困难的业务问题

注意



<http://www.umlchina.com>

书写用例文档

——补充约束：业务规则

❖ 事实

- ❖ 设备是资产的一种

❖ 推理

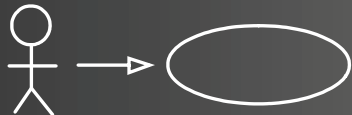
- ❖ 如果过了计划中的交付日期，货物还没有送到，即为“未按时送货”

❖ 约束

- ❖ 合同的总金额不能超出买方的信用额度

❖

各种业务规则



书写用例文档

——补充约束：业务规则

- ❖ 文字说明
- ❖ 决策表
- ❖ OCL
- ❖ 行业上适用的任何方式
- ❖

业务规则的各种表示方法

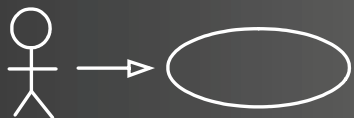


书写用例文档

——补充约束：业务规则

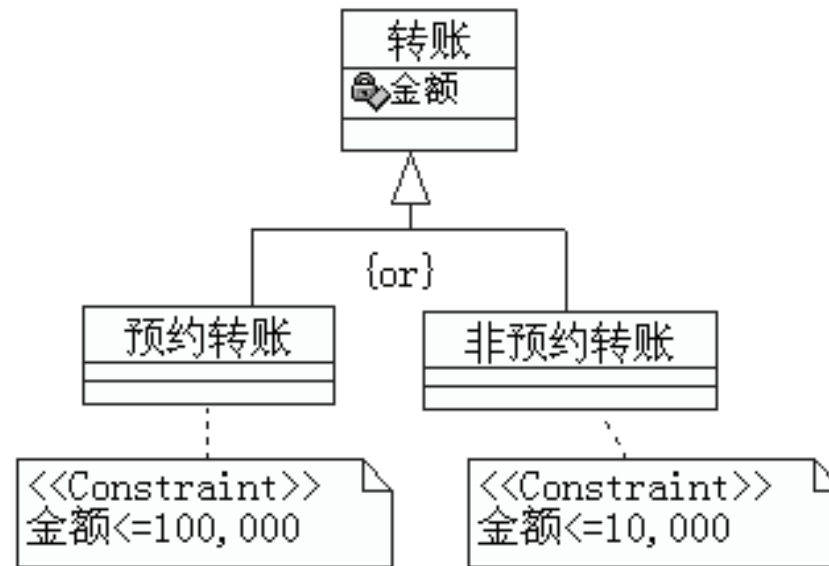
条件	1. 双人房用作单人房	Y	N	(Y)		
	2. 家庭额外客房	N	Y	Y		
	3. 立即入住				Y	Y
	4. 18:00 之前				Y	Y
	5. 预订率低于 50%				Y	Y
	6. 晴天				N	Y
行为	a. 25%	√				
	b. 10%		√			
	c. 立即入住 25%					√
	d. 立即入住 0%				√	

决策表

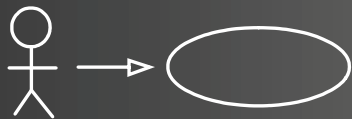


书写用例文档

——补充约束：业务规则



OCL



书写用例文档

——补充约束：业务规则

❖ 系统将语音输入翻译为文字

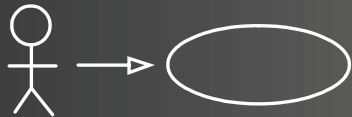
❖ 采用××识别算法

❖ 背景噪音强度为××的情况下，识别率应在××以上



警惕误把设计当成规则

业务规则！ = 实现算法

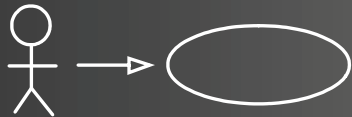


书写用例文档

——补充约束：非功能需求

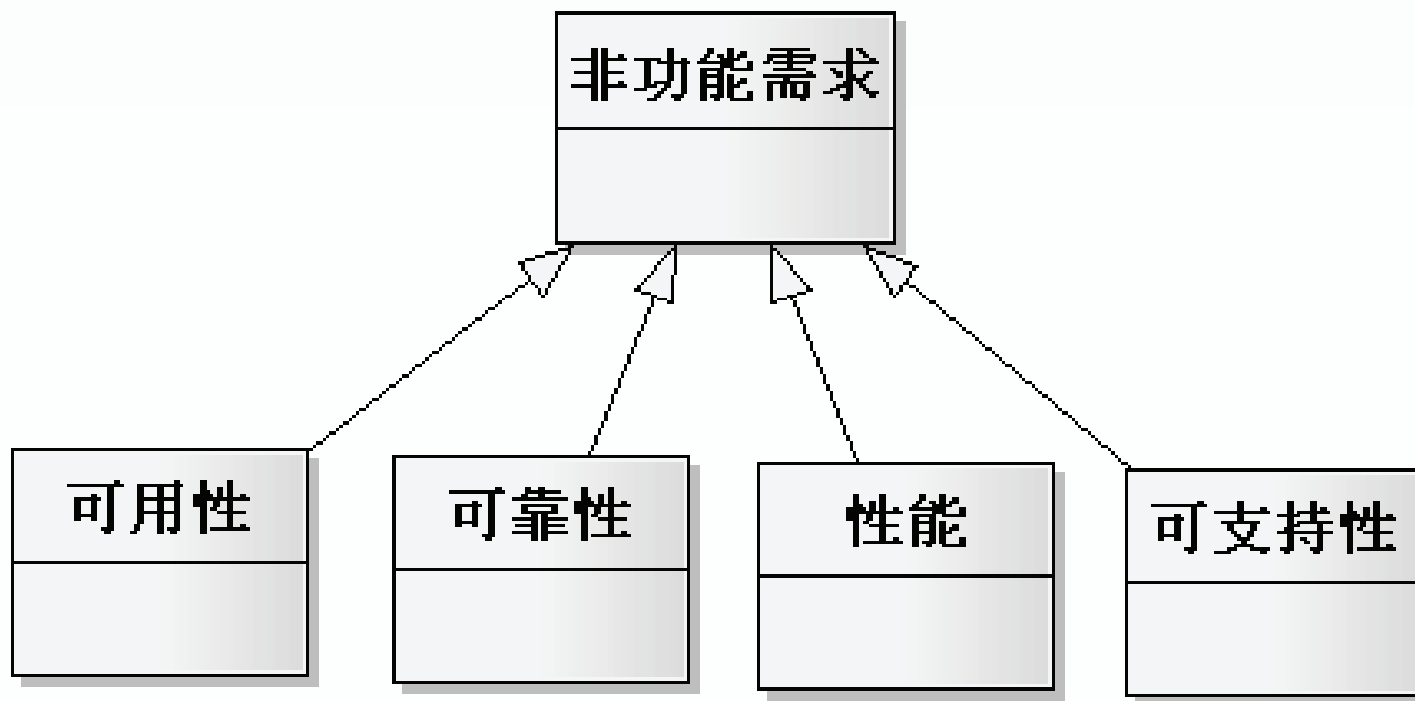


激烈竞争的决胜点

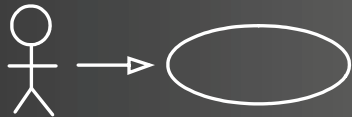


书写用例文档

——补充约束：非功能需求



URPS

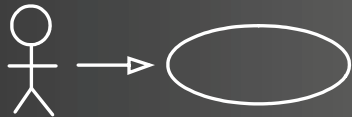


书写用例文档

——补充约束：非功能需求：可用性



- ❖ 系统没有按程序员的意图工作→程序错误
- ❖ 系统无法执行一项任务→功能需求遗漏
- ❖ 系统能按照程序员意图工作，并且支持任务→但用户仍然不知道如何使用系统执行任务或者不喜欢使用系统执行任务→可用性问题

“可用性”！=用例



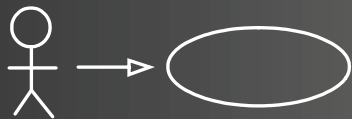
书写用例文档

——补充约束：非功能需求：可用性

- ❖ 系统应易于使用 
- ❖ 人事专员第一次使用时30分钟内能学会添加新员工（任务时间）
- ❖ 前台5次击键能完成客人入住服务，不需要使用鼠标（操作次数）
- ❖ 80%的用户认为系统易学，并且使用效率高（用户调查）
- ❖ 系统界面应如××附件所示的屏幕图像（小心！） 



可用性需求的表达



书写用例文档

——补充约束：非功能需求：可靠性

- ❖ 系统应能防范磁盘故障（安全）
 - ❖ （对照）系统应采用冗余磁盘阵列
- ❖ 系统应保证收到的数据和发送的数据一致（完整）
- ❖ MTBF（Mean Time Between Failures）
- ❖ MTTR（Mean Time To Repair）（稳定）

各种可靠性需求

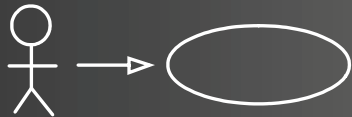


书写用例文档

——补充约束：非功能需求：性能

- ❖ 系统应在0.5秒之内拍摄超速车的照片（速度）
- ❖ 系统应允许1000个用户同时使用（容量）
- ❖ 在标准工作负荷下，系统的CPU占用率应少于50%（能力）

各种性能需求

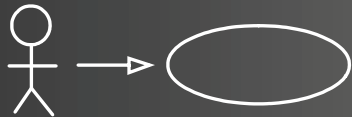


书写用例文档

——补充约束：非功能需求：可支持性

- ❖ 95%的紧急错误应能在30工作小时内修复
- ❖ 在修复故障时，未修复的相关缺陷平均数应小于0.5
- ❖ 在两年内，以每功能点××的价格升级系统
- ❖ 升级新版本时，应保存所有系统设置和个人设置

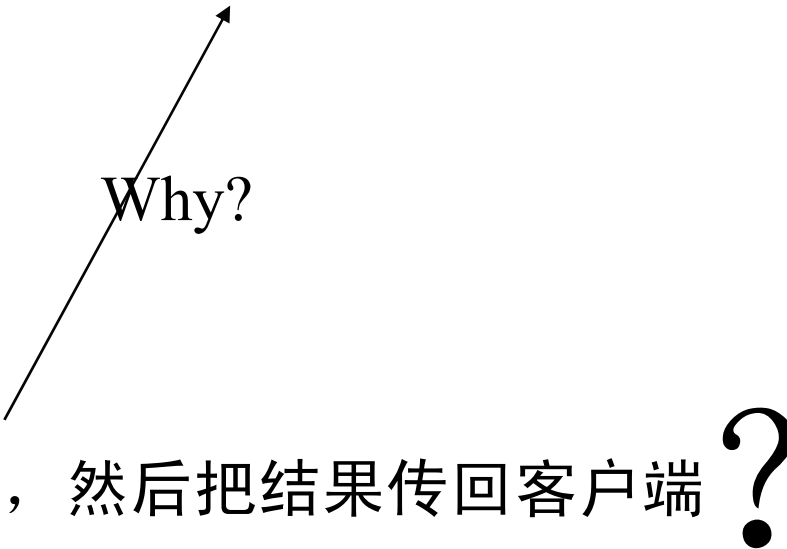
各种可支持性需求



用例文档

系统应在3秒内向用户反馈应收款项

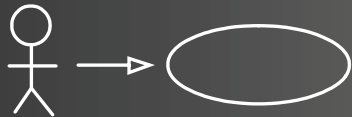
Why?

A diagram consisting of a diagonal arrow pointing from the bottom-left towards the top-right. The word "Why?" is written in the middle of the arrow. At the end of the arrow, pointing towards the top-right, is a large, bold question mark.

系统在服务器端计算应收款项，然后把结果传回客户端

?

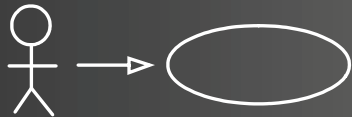
混入需求的设计——往往隐含非功能需求



书写用例文档

——补充约束：设计约束

- ❖ 界面样式
- ❖ 报表
- ❖ 平台
- ❖ 语言
- ❖ 外系统接口
- ❖ 行业规定
- ❖



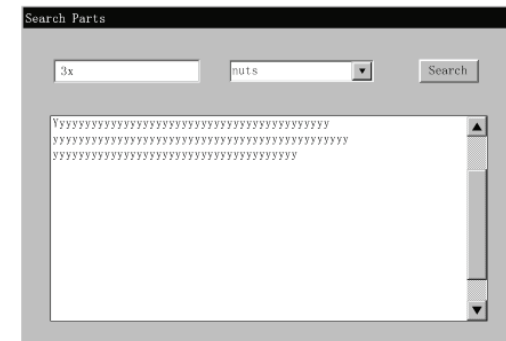
书写用例文档

——补充约束：设计约束

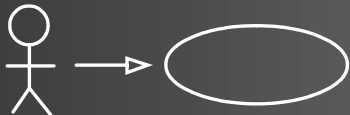
❖ 5. 系统显示订单明细

必须来自涉众！

❖ 5. 应采用附件××所示的屏幕格式.....



界面样式



书写用例文档

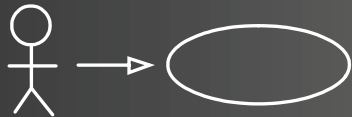
——补充约束：设计约束

❖ 5. 系统打印工资单

必须来自涉众！

❖ 5. 工资单的格式如附件××所示.....

报表

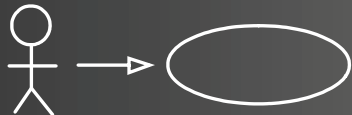


书写用例文档

——补充约束：设计约束

- ❖ 由于公司目前大约有200台PC在使用，还需要使用若干年，所以系统应能在这些PC上运行（Pentium 166，128M RAM）
- ❖ 由于×公司的IT人员有Oracle专长，而且目前已有的其他应用系统也使用Oracle数据库，所以系统应使用Oracle数据库。

平台：一般针对整个系统



书写用例文档

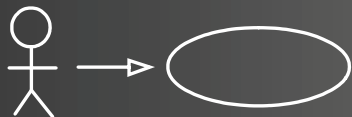
——补充约束：设计约束

❖ 5. 系统读取探测仪读数



❖ 5. 通过RS-232接口读取

技术接口



书写用例文档

——补充约束：设计约束

- ❖ 系统应采用三层架构方式搭建

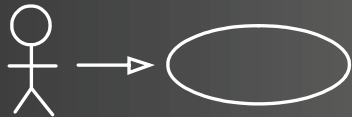
?

什么是涉众
能够理解和验证的？

- ❖ 系统应允许分布在各地的多个用户并发使用（非功能需求）



小心伪装成“设计约束”的“设计”



书写用例文档

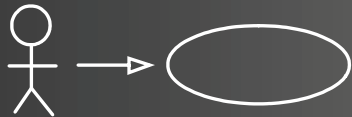
——补充约束：设计约束

- ❖ 录单界面应分为3个页面，每个页面填写完毕？点击“下一步”，出现下一页面。



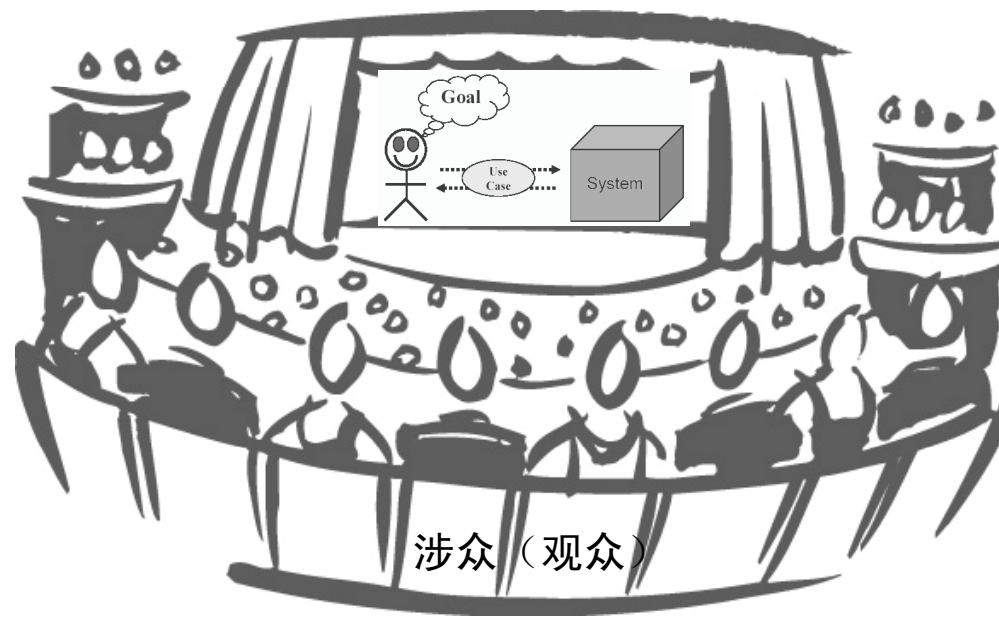
- ❖ 系统应在××秒内显示录单界面（非功能需求）

小心伪装成“设计约束”的“设计”

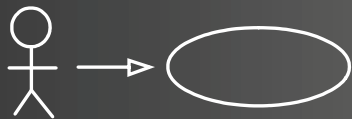


书写用例文档

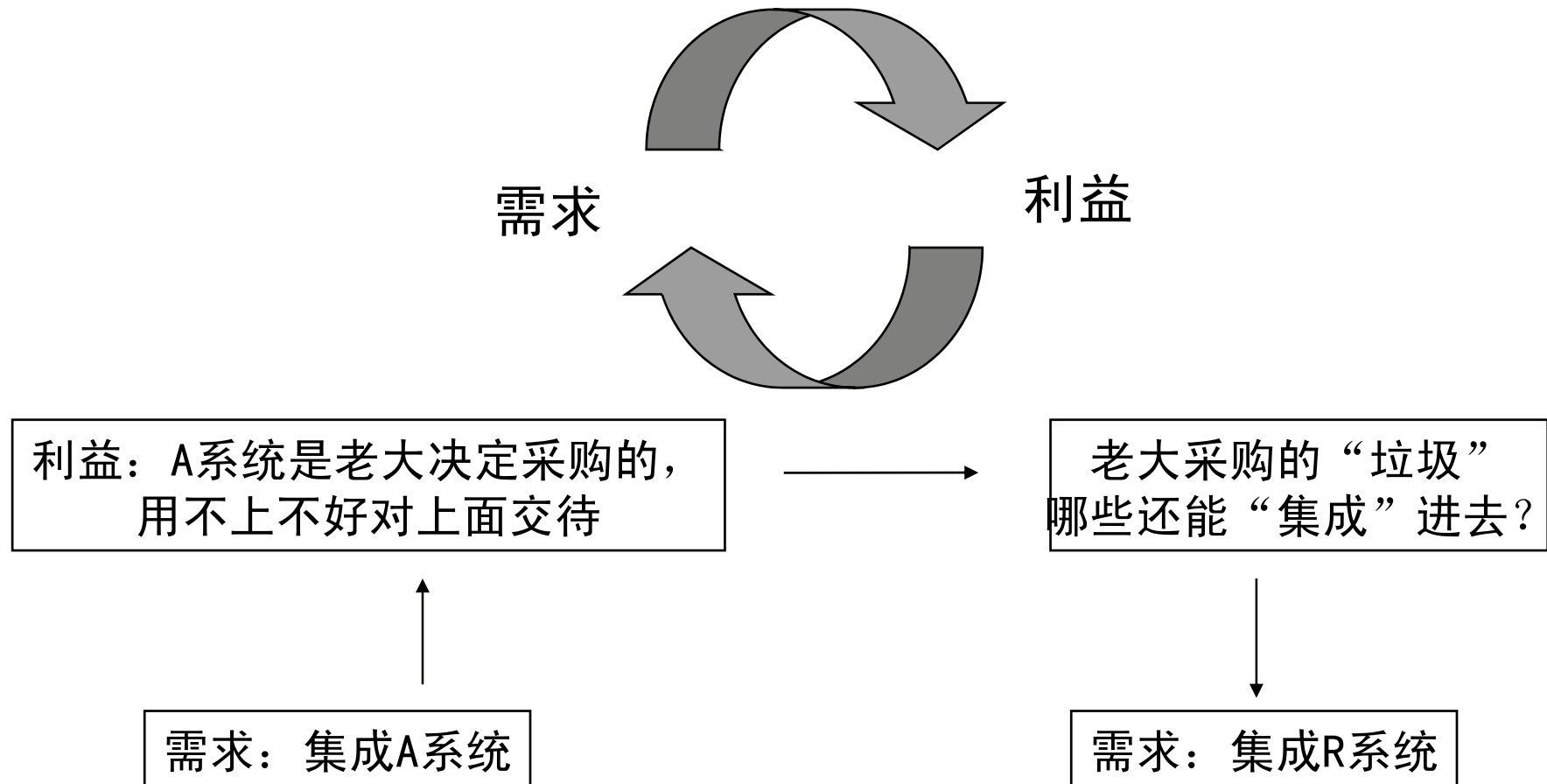
——最后，再看一遍！



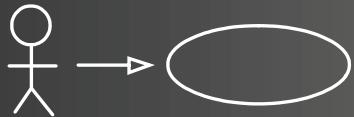
写对了吗？别忘了和涉众利益对照！



书写用例文档



涉众利益和需求



书写用例文档

❖ 涉众利益

CEO——希望佣金能灵活按照公司佣金政策调整准确计算；

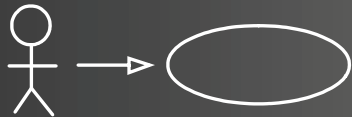
经纪——担心少算；担心拖延；

❖ 设计约束

使用C#和.NET开发

?

涉众利益和需求



书写用例文档

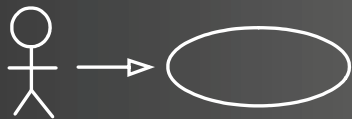
❖ 每个用例都要写文档吗？

❖ 用例文档一定要规范完整吗？

❖ 心里有观众胜过整齐的剧本

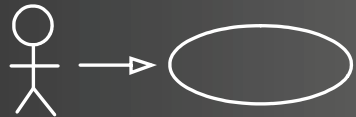
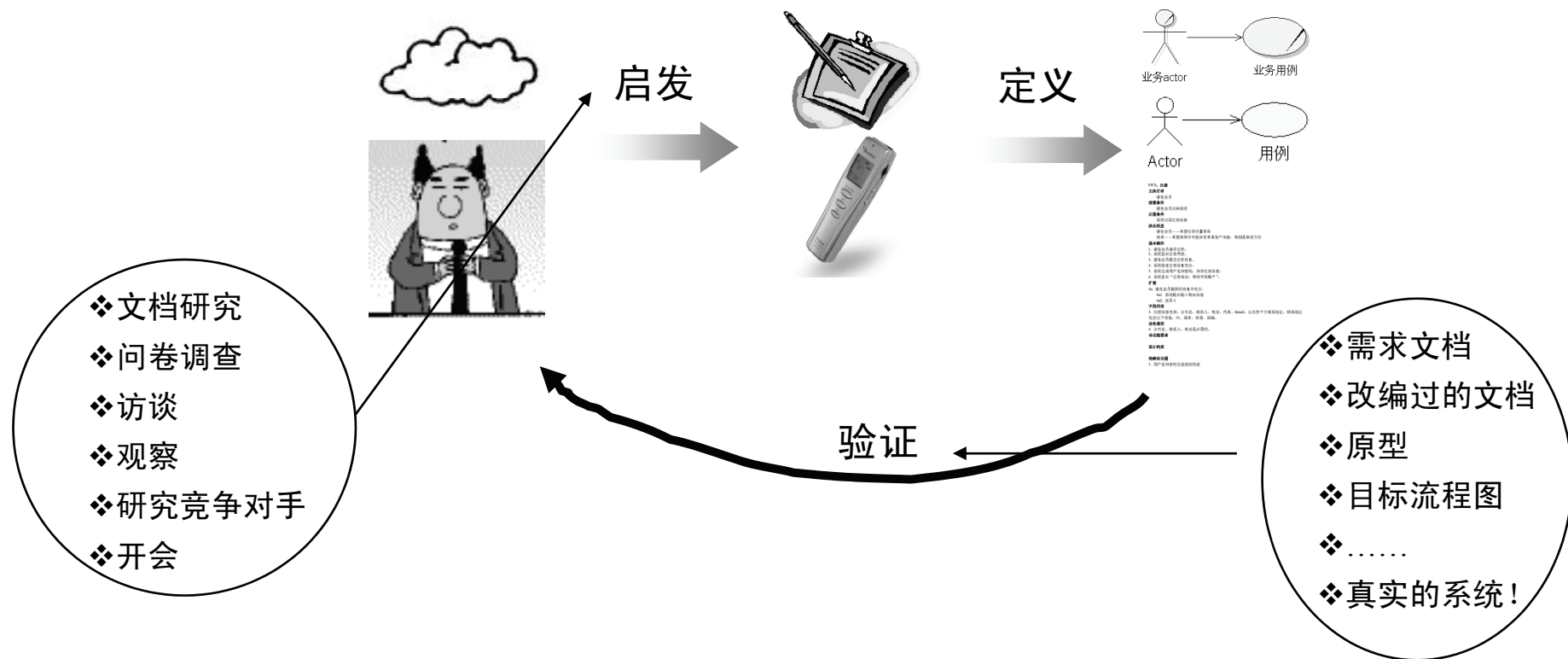


不浪费原则



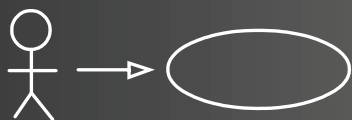
书写用例文档

——用例的验证



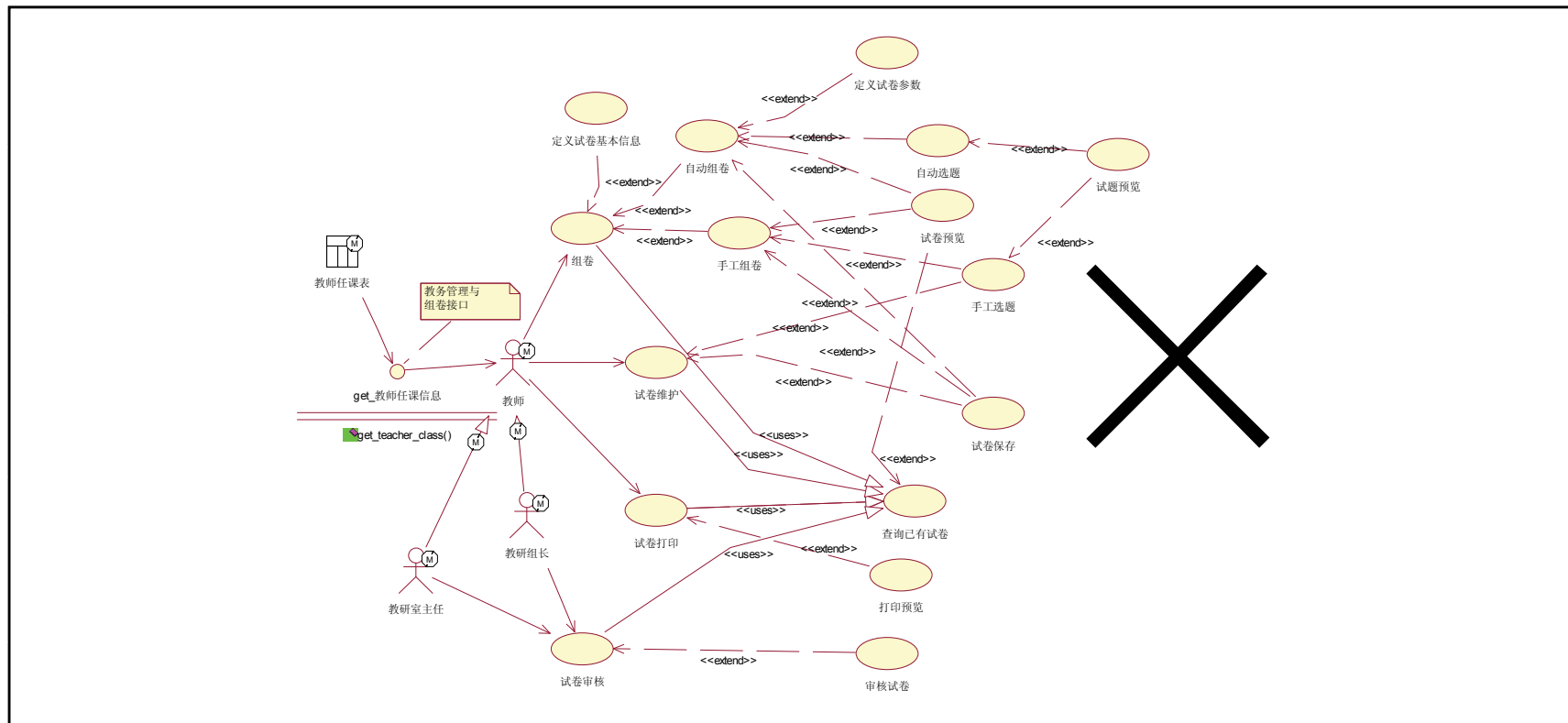
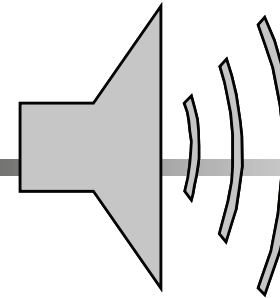
书写用例文档

——讨论与练习、项目实作

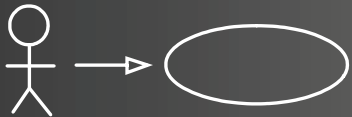


识别用例的关系

——注意

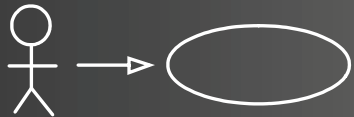
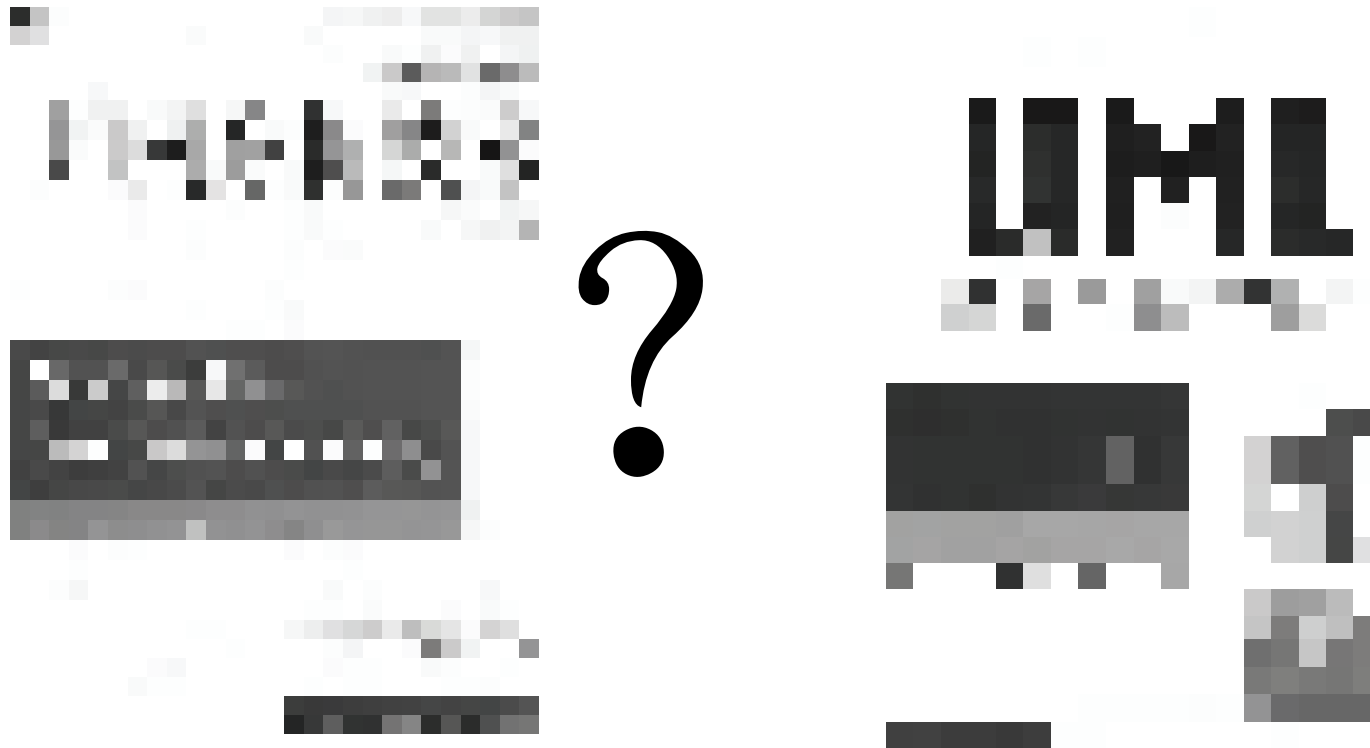


不要滥用用例关系！！！！



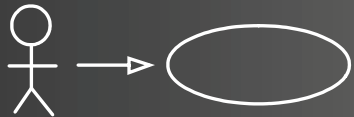
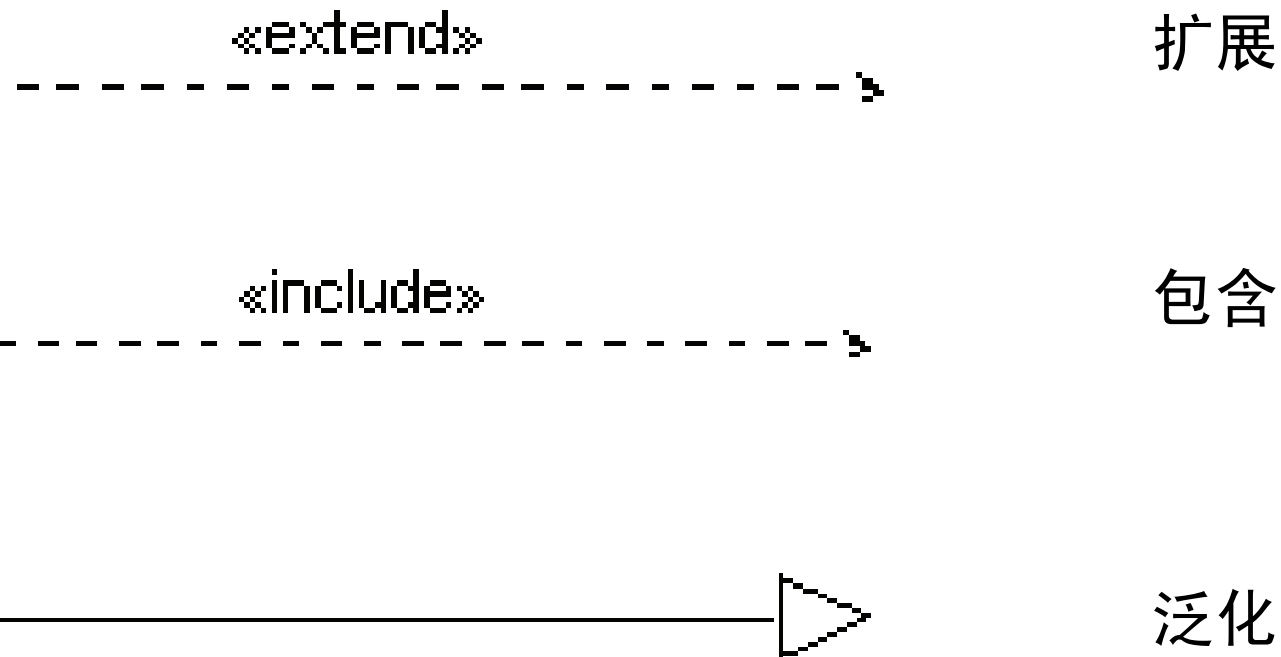
识别用例的关系

——某些书也有误导



识别用例的关系

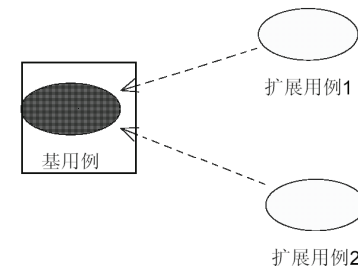
——用例的关系



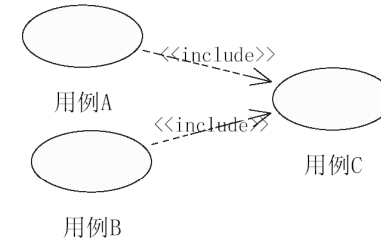
识别用例的关系

——通过关系整理用例文档

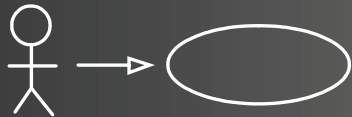
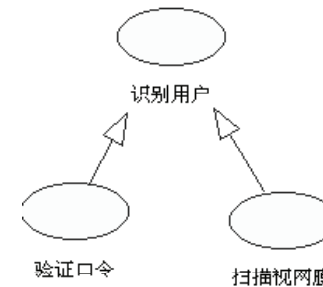
❖ 扩展：分离扩展路径



❖ 包含：提取公共步骤集合，便于复用



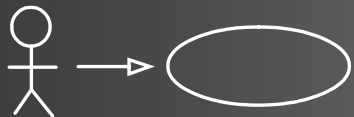
❖ 泛化：同一业务目的的不同技术实现



识别用例的关系

——何时使用扩展关系

- ❖ 扩展路径步骤多
- ❖ 扩展路径内部还有扩展点——扩展之扩展
- ❖ 扩展路径未定或容易变化——分离以“冻结”基用例



识别用例的关系

——扩展关系举例



6. 会员可以选择以下动作：

从订单中删除订单项

修改某个订单项的购买数量

修改订单的送货地址

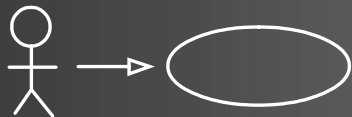
扩展

6a. 会员从订单中删除订单项：

6a1. 会员请求从订单中删除某个订单项

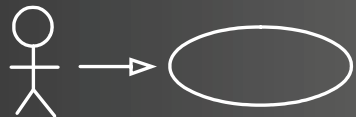
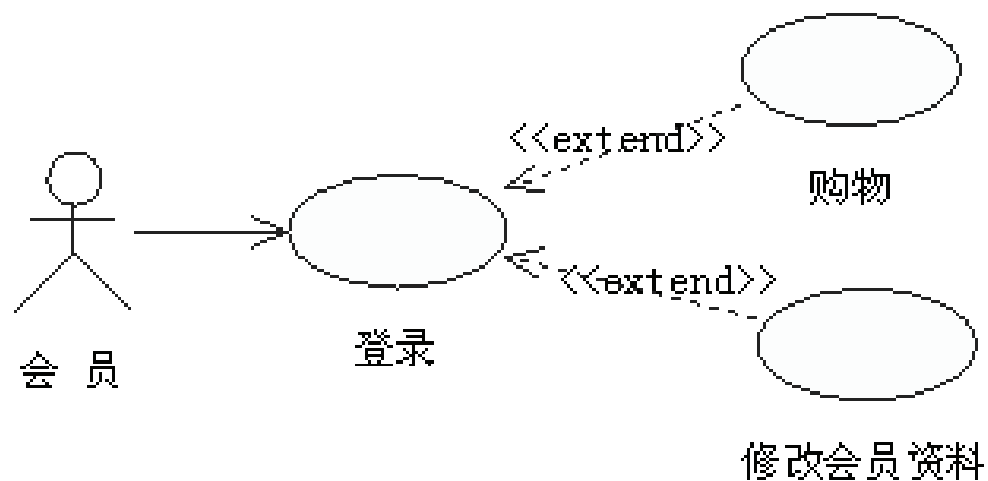
6a2. 系统删除该订单项

6a3. 返回 5



识别用例的关系

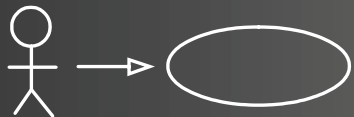
——扩展关系的误用



识别用例的关系

——何时使用包含关系

- ❖ 步骤集合在多个用例重复出现，且单独形成价值
- ❖ 用例的步骤较多时，可以用Include简化（慎用）



识别用例的关系

——包含关系举例



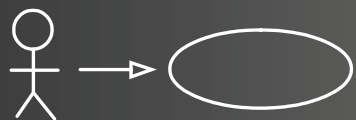
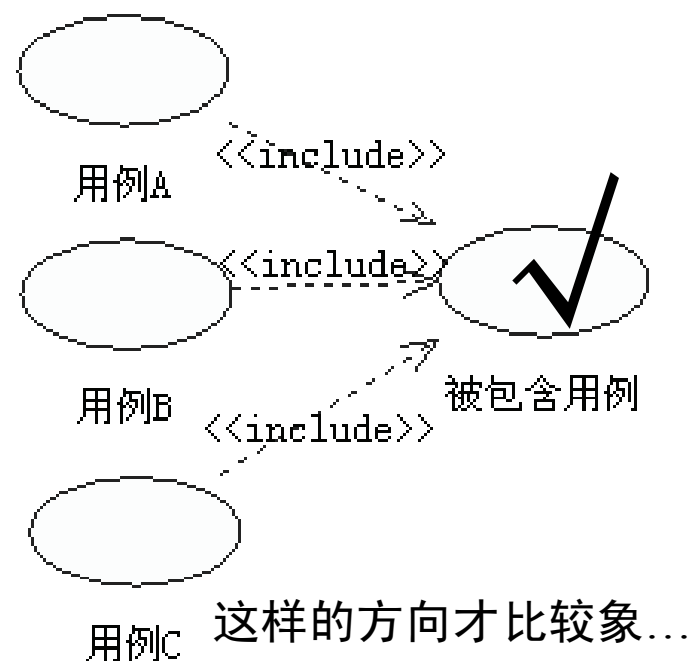
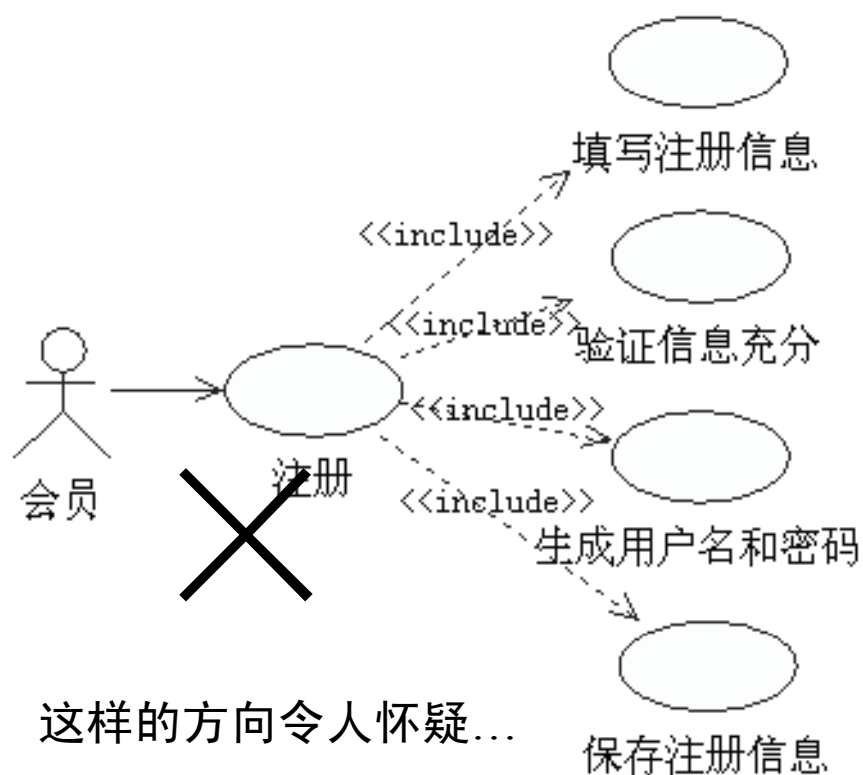
基本路径

1. 会员登录
2. 会员指定零件和购买数量
2. 系统显示会员订单列表
3. 会员可以选择：
 - 添加到新订单
 - 添加到已有订单
4. 系统显示当前订单



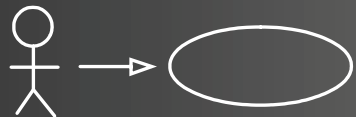
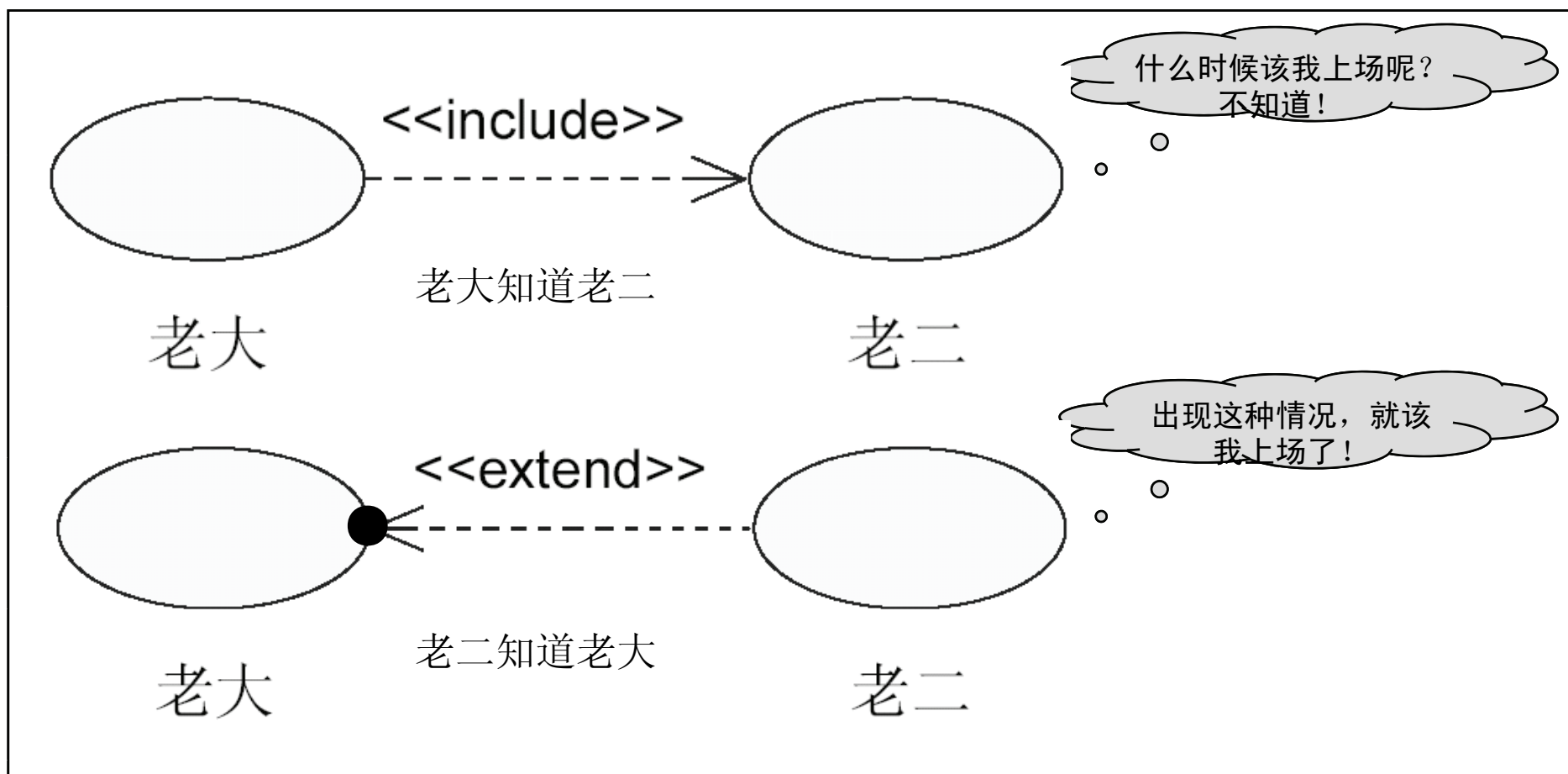
识别用例的关系

——包含关系的误用



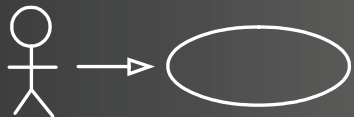
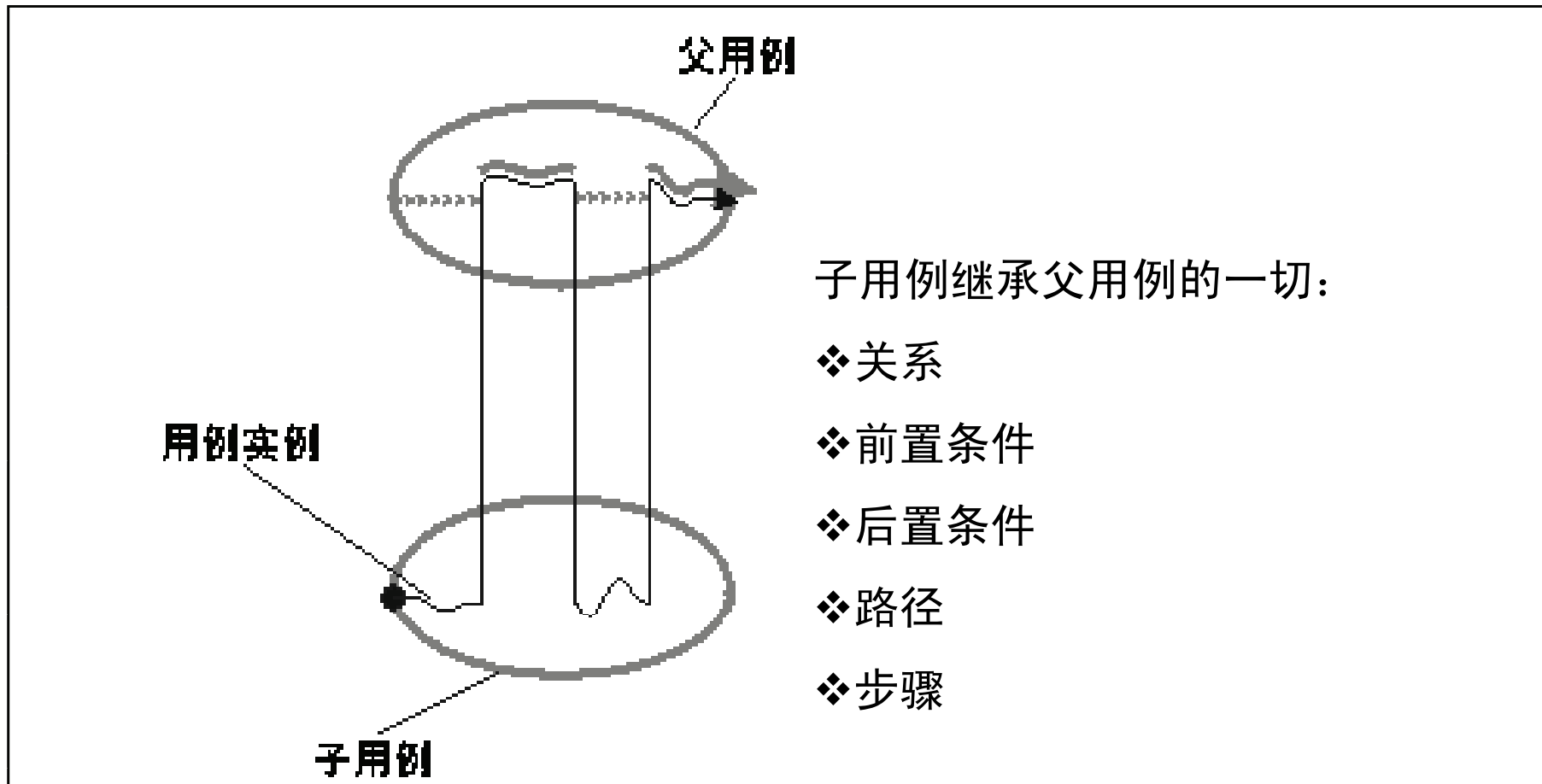
识别用例的关系

——扩展 vs. 包含的可见性



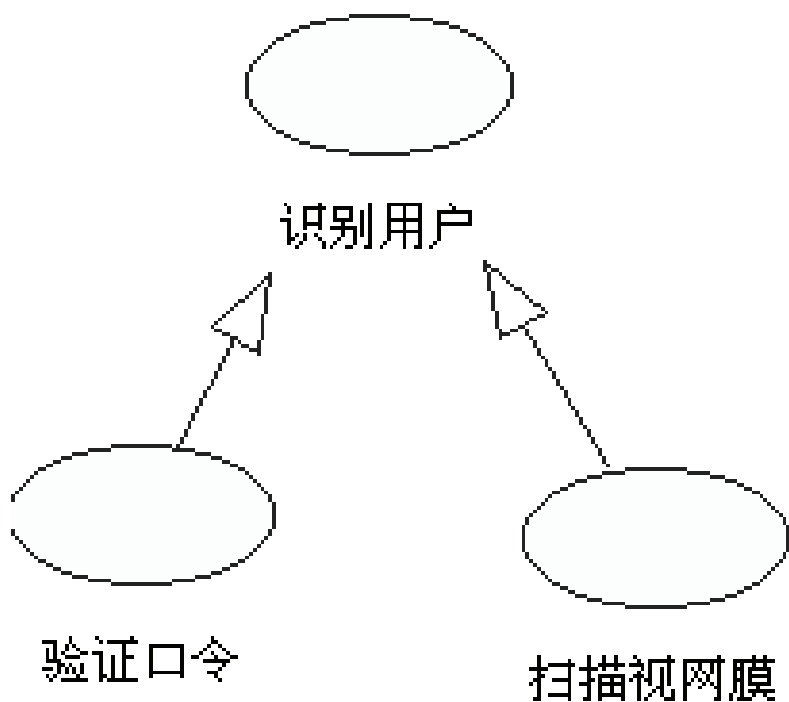
识别用例的关系

——泛化



识别用例的关系

——泛化关系举例



同一业务目的不同技术实现

UC20 检索螺钉

父用例

UC03

主参与者

潜在会员

前置条件

参与者访问系统

后置条件

参与者查询到所要的零件

基本路径

1. 参与者提交查询条件
2. 系统按查询条件检索螺钉
3. 系统显示搜索到螺钉的编号、类别、价格
4. 参与者选中某个螺钉
5. 系统显示该螺钉的详细信息

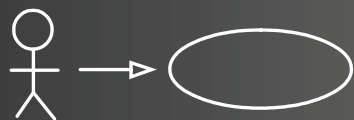
扩展点

- 2a. 系统没有检索到所需螺钉:
- 2a1. 系统显示“没有找到适合条件的螺钉”
 - 2a2. 用例结束

补充说明

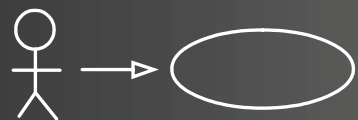
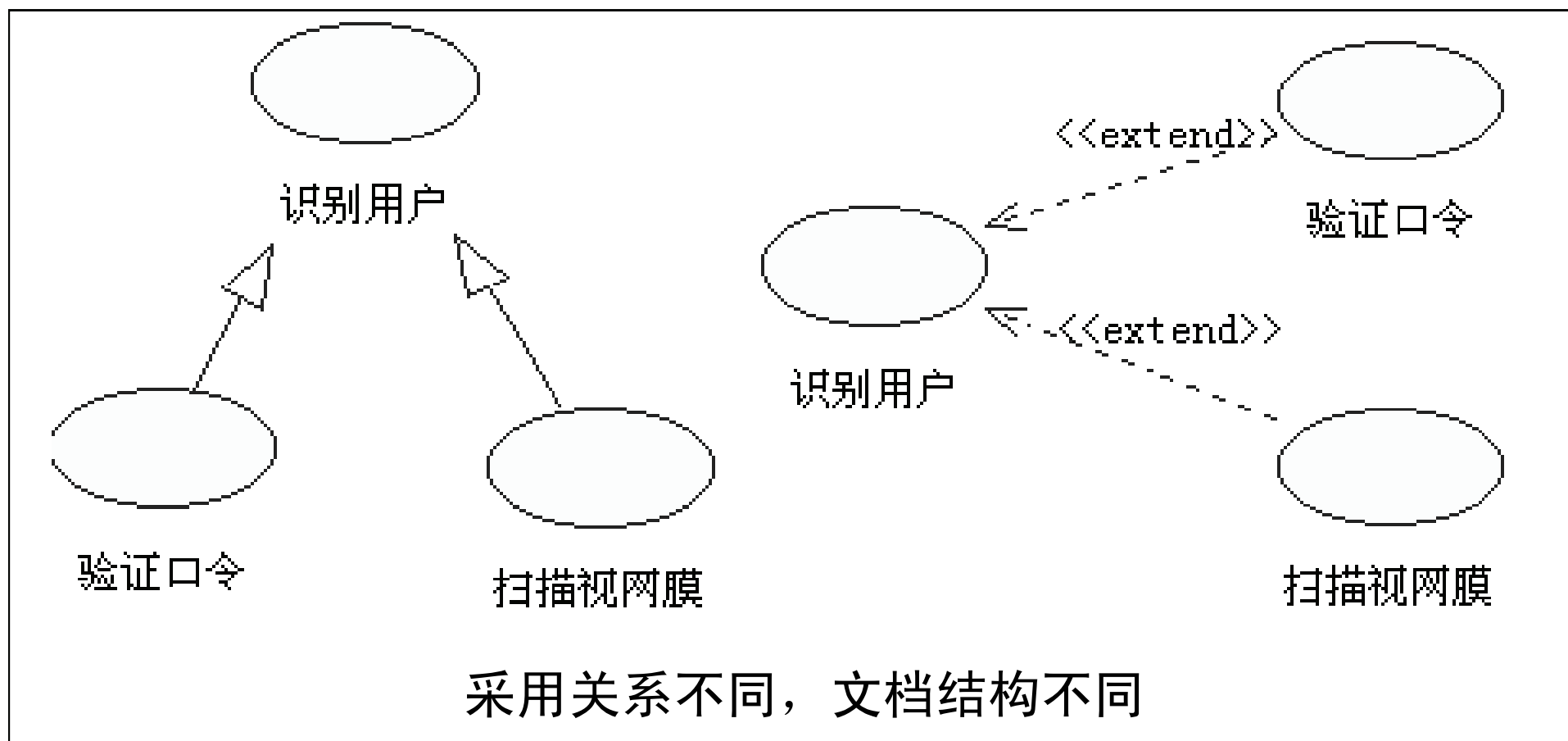
.....

泛化

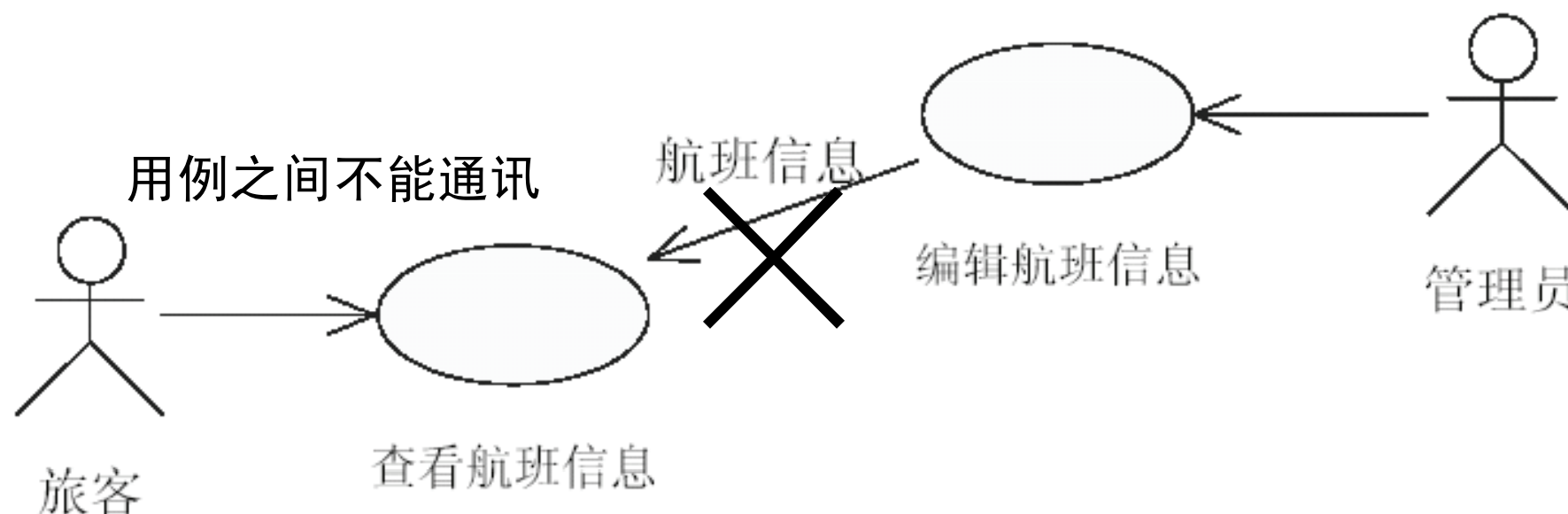


识别用例的关系

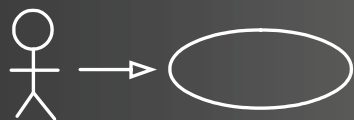
——泛化？扩展？



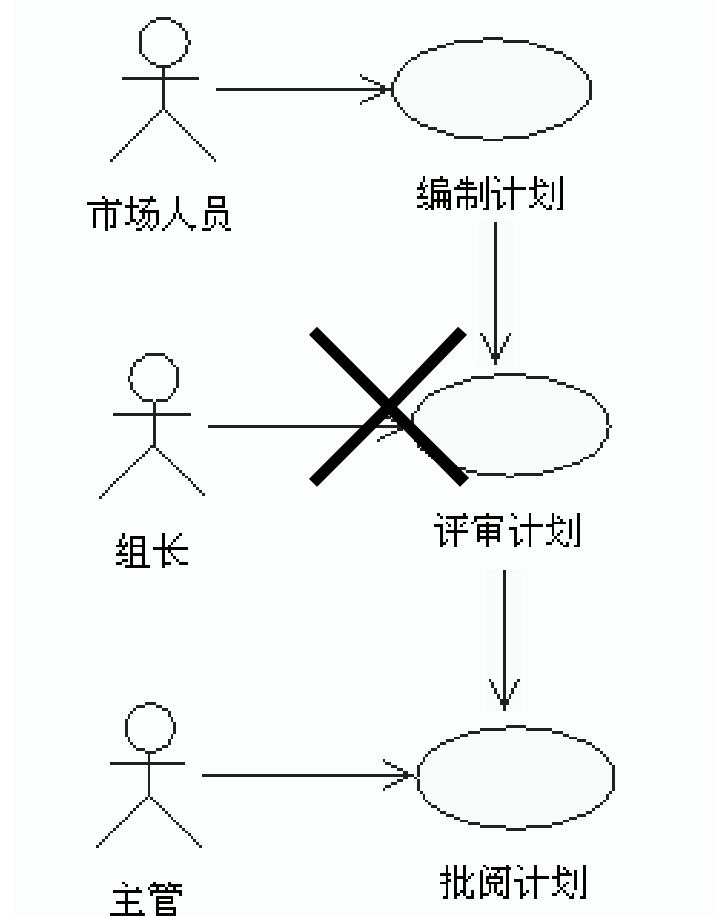
识别用例的关系



除此之外，不能有别的关系

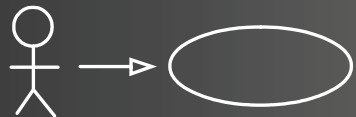


识别用例的关系

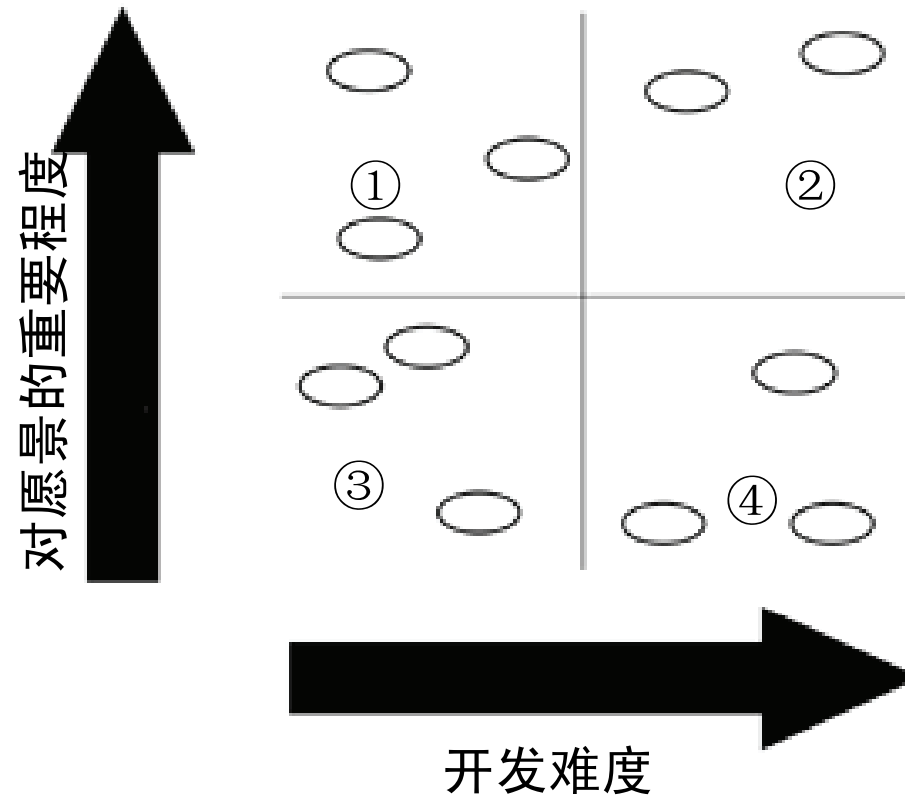


小心“羊肉串”！

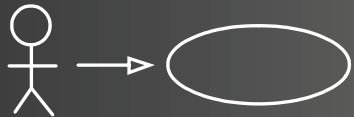
用例是执行者一份一份地和系统签订的契约



用例排序

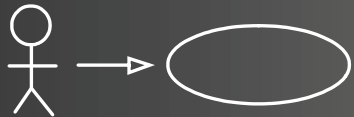


用例散点图：四个象限

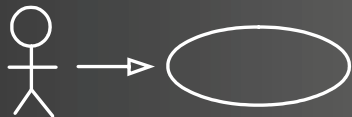
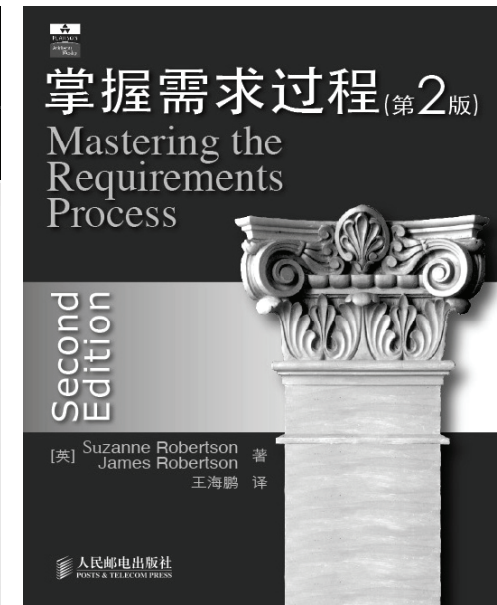
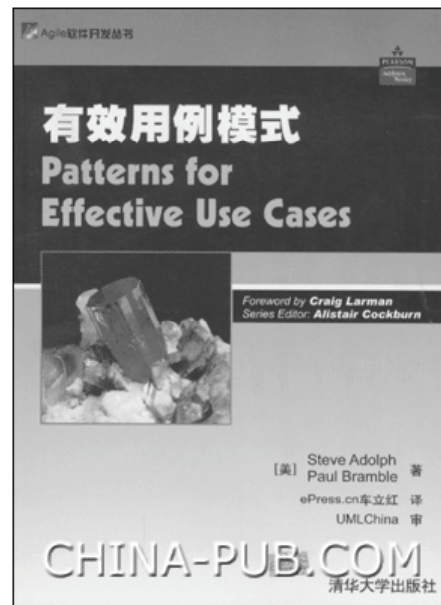
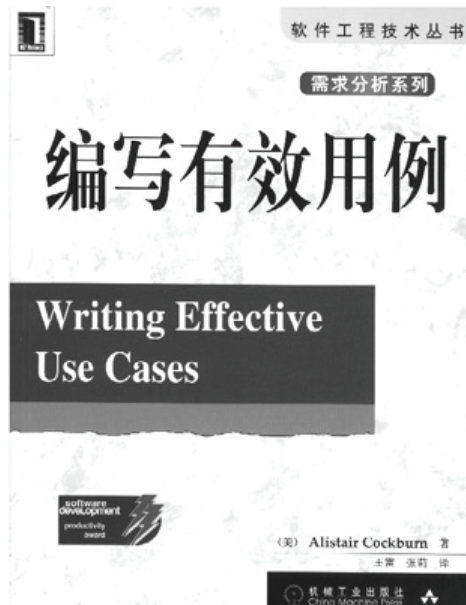


大量用例时的组织

- ❖ 按执行者分包
- ❖ 按主题分包
- ❖ 按开发团队分包
- ❖ 按发布情况分包



用例书籍

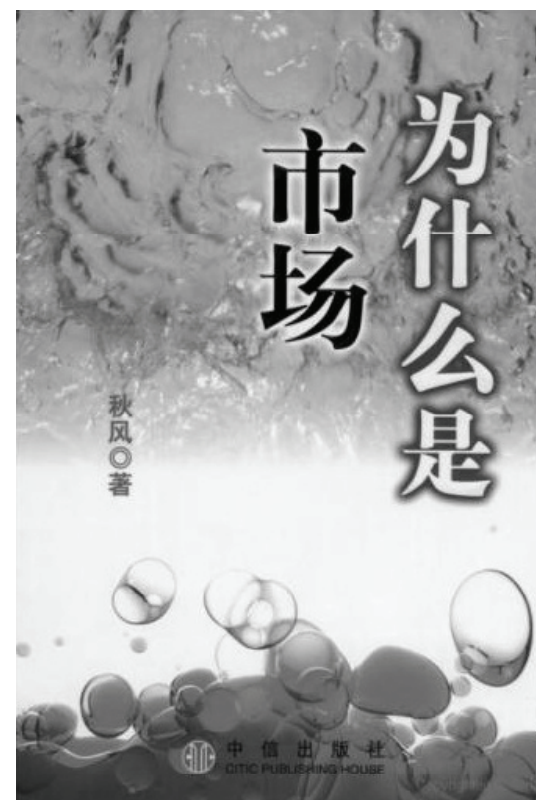


<http://www.umlchina.com>

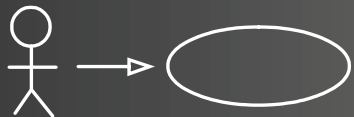
用例书籍——扩展读物



用于软件——软件需求
用于组织——业务建模
用于社会——????



帮助理解“涉众”、“利益”、“妥协”、“契约”



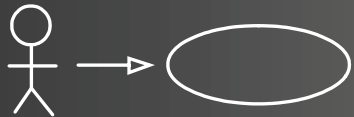
<http://www.umlchina.com>

用例

用例是否用对了？判断标准：

是否加强了和涉众的联系

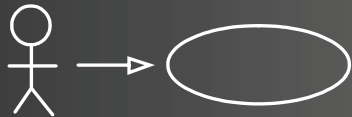
最后总结



下面呢？



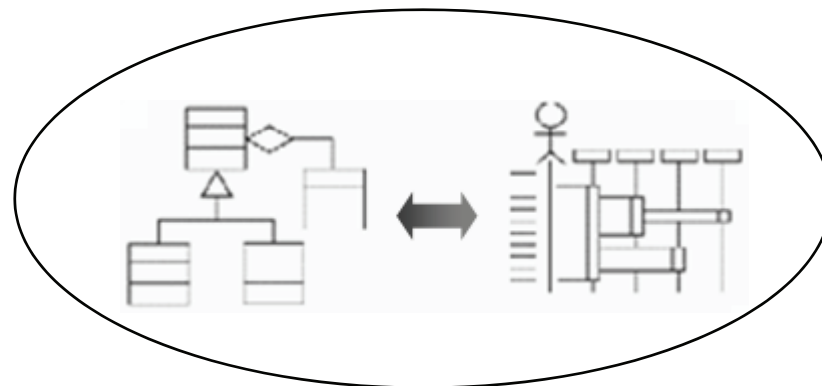
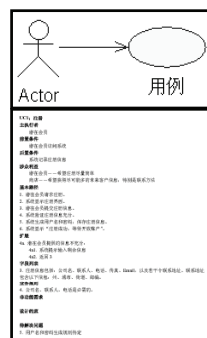
从前，有一个太监...



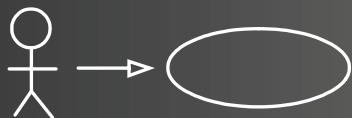
<http://www.umlchina.com>

[illegible]

用例与分析设计



用例如果用OO方法实现



<http://www.umlchina.com>