基于用例的需求建模

Chapter 4

北京航空航天大學
SCHOOL OF SOFTWARE BEIHANG UNIVERSITY 软件学院

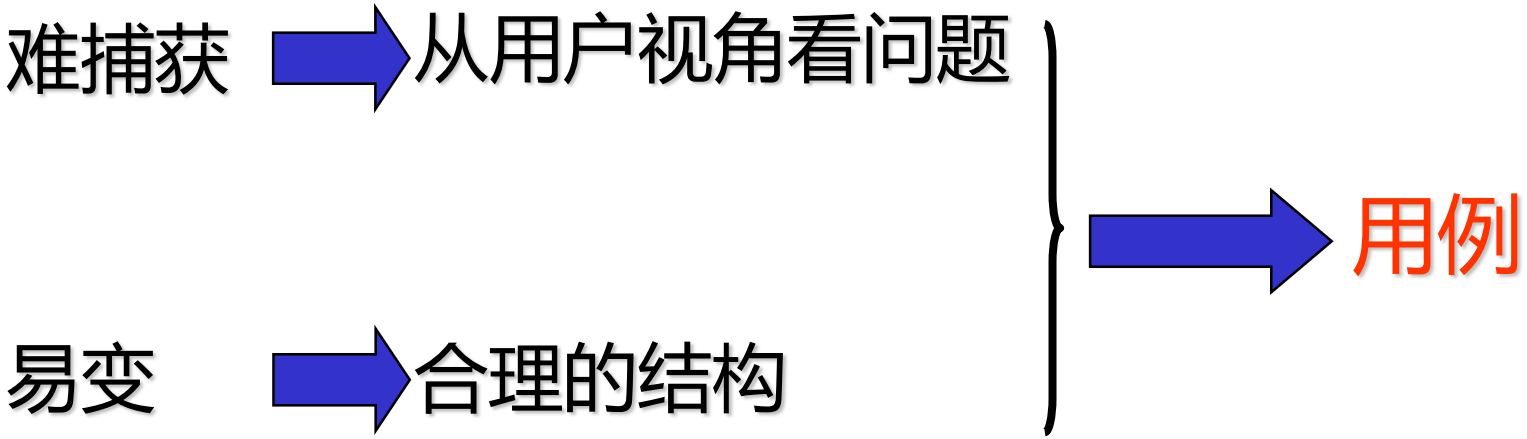
内容安排

1. 用例建模基础
2. 文档化用例
3. 重构用例模型

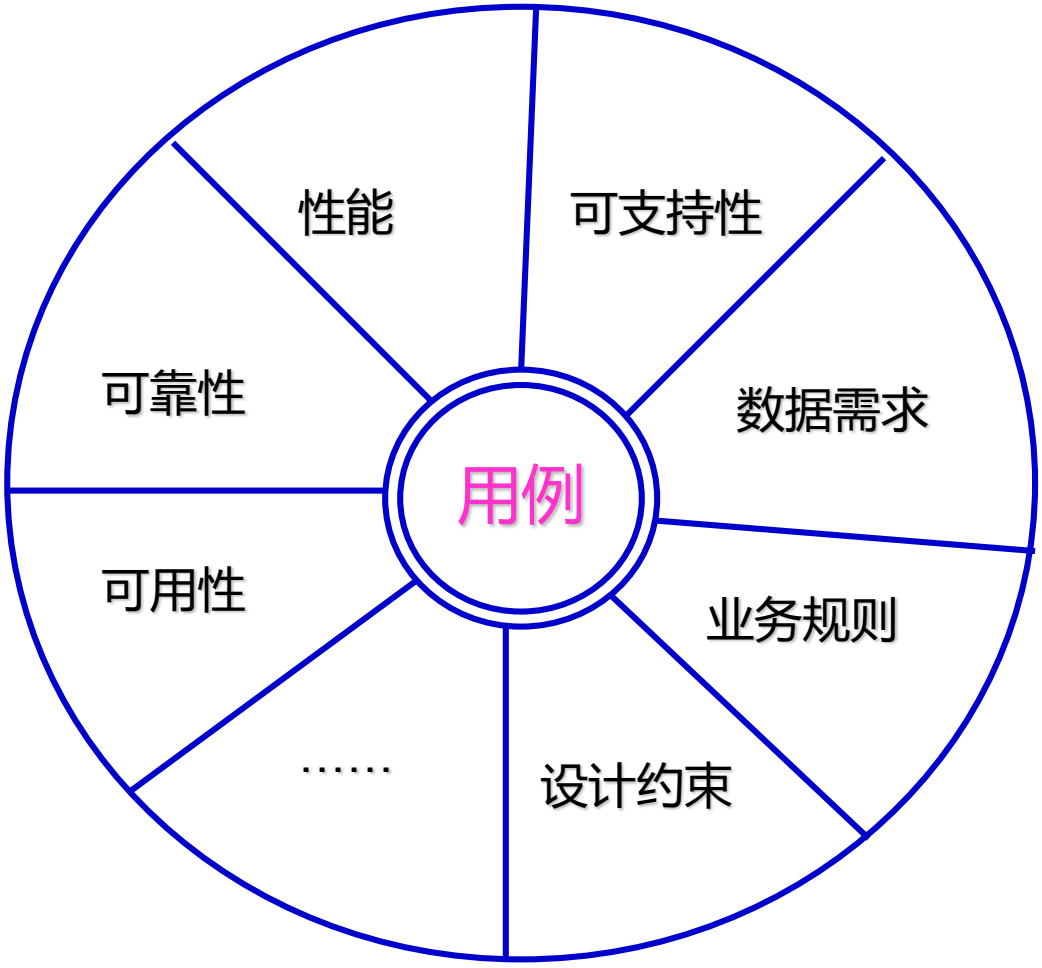
内容安排

1. 用例建模基础
2. 文档化用例
3. 重构用例模型

需求问题：对策



以用例为中心的需求开发

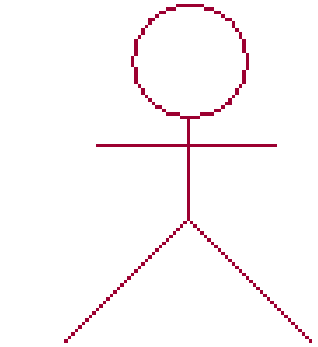


用例模型

- 用例模型和用例驱动的开发方法
  - 一种系统功能需求建模方法
  - 借助于用例、场景等概念建模系统功能需求
  - 基于用例驱动后续的分析和设计过程
- 核心概念
  - 用例：系统对外部用户提供的价值（功能）
  - 参与者：系统外部用户、接口或环境

参与者

- 关键词: 边界
- 参与者
  - 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

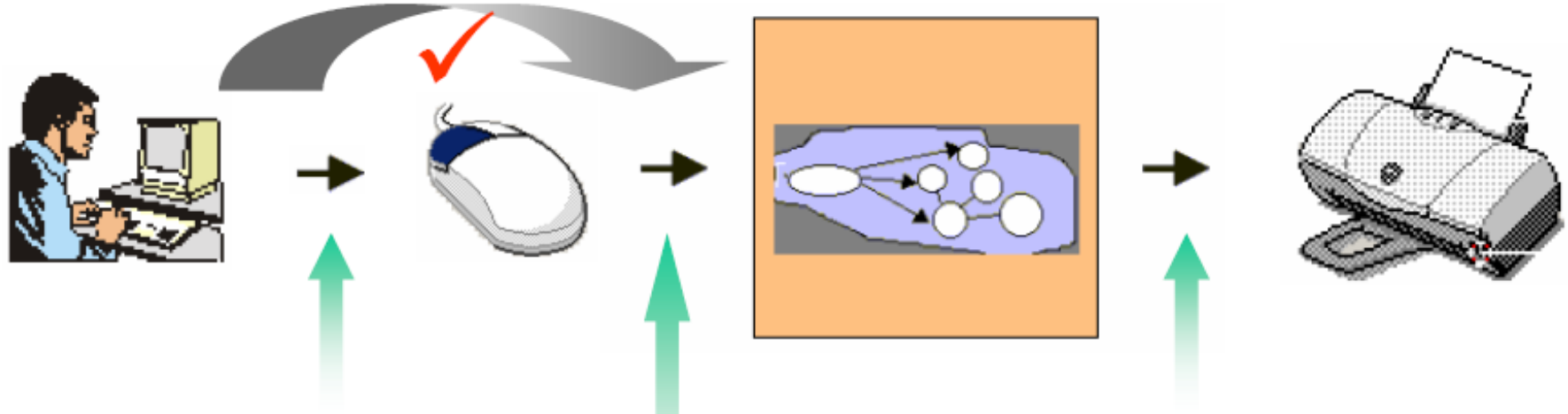


见过我吗？
我是参与者

要点分析

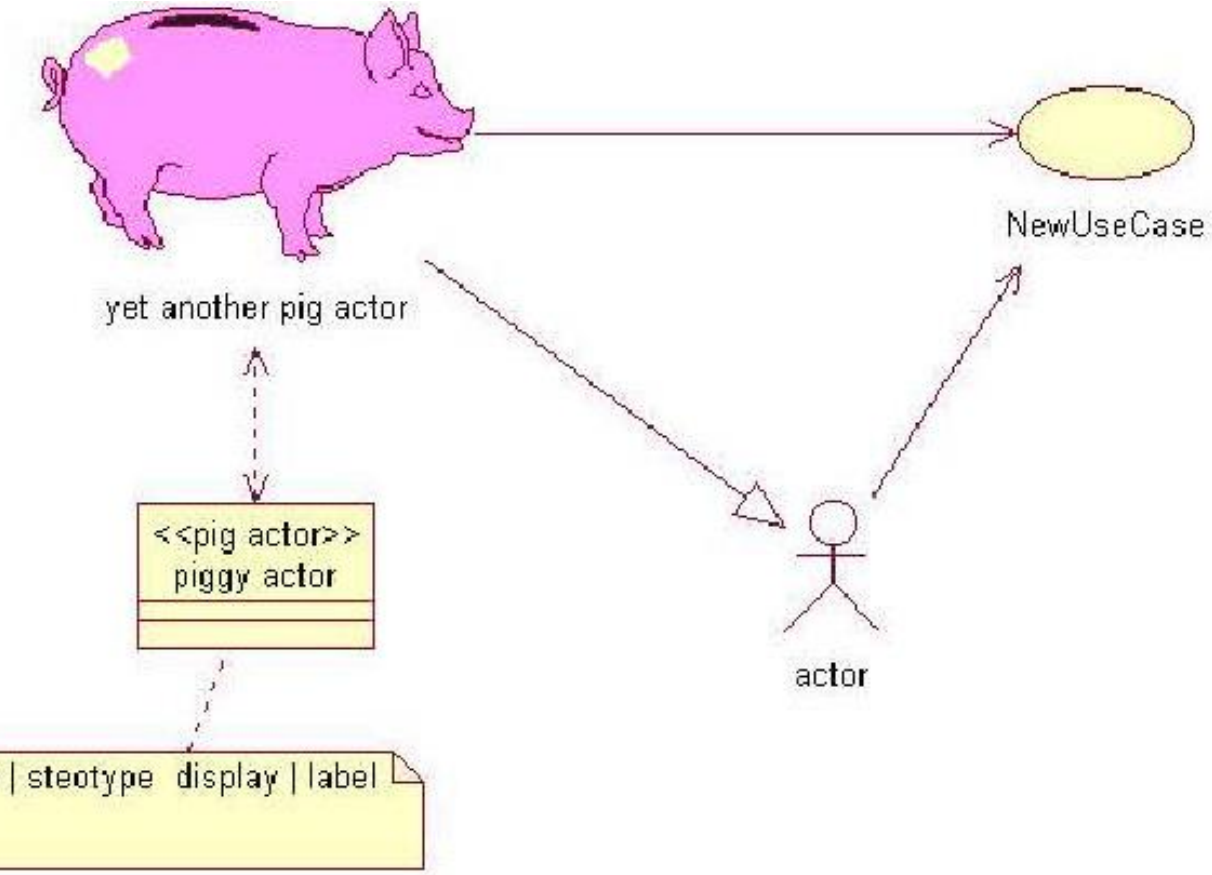
- 系统外
  - 参与者不是系统的一部分，处于系统的外部
- 系统边界
  - 参与者透过系统边界直接与系统交互，参与者的确定代表系统边界的确定
- 与系统交互
  - 系统需要处理其交互过程，即系统职责
- 任何事物
  - 人、外系统、外部因素、时间
- 系统角色
  - 参与者与使用系统的物理人和职务没有关系
  - 需要从参与系统的角色(作用)来寻找参与者

要点：与系统进行信息交互



任何事物：小人与圣小猪

- 如果开发一个猪圈自动供食供水系统，猪的前蹄触发一个开关系统就供食或供水。显然，这里的参与者是小猪



思考：参与者与系统边界

- 某企业要求开发一个企业信息管理系统，并与原来已有的库存系统相连接
- 某企业要求开发一个企业信息管理系统，并把原来已有的库存管理系统加以改造，成为企业信息管理系统的一部分

识别参与者的思路

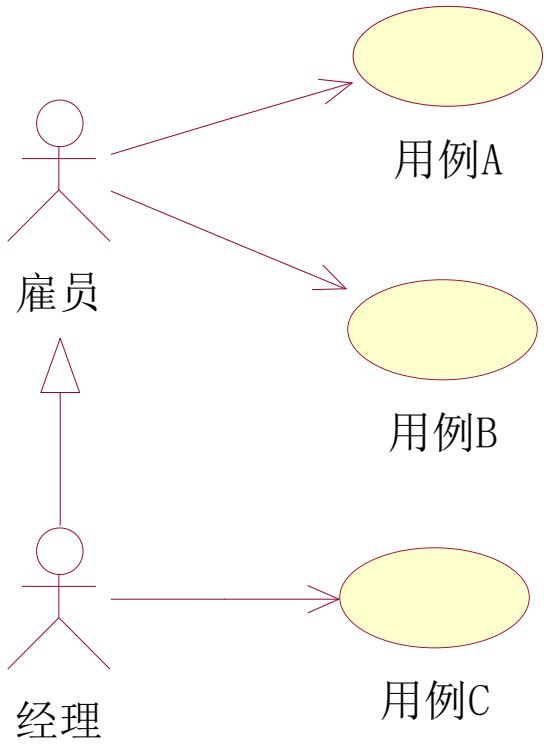
- 可以从以下要点来识别参与者
  - 系统在哪些部门使用
  - 谁向系统提供信息、使用和删除信息
  - 谁使用系统中的信息完成业务
  - 谁对系统进行维护
  - 与外部系统是否有关联
  - 时间参与者：一种习惯用法，用于激活那些系统定期的、自动执行的用例

命名参与者

- 对参与者赋予能更好地表达其角色(作用)的名称
  - 不好的参与者命名的例子：用职务名称和个人姓名来命名
    - 例如，张三、老李、校长、科长…
    - 若使用系统的人（职务名称）变化的话，就不是参与者了
  - 好的参与者命名的例子：用能知道其角色的名称来命名
    - 例如，学生、订单管理员、维护部门…
    - 即使使用系统的人改变，从系统来看，使用者的角色（作用）是相同的

参与者之间的关系：泛化

- 参与者可以通过泛化关系来定义
  - 在这种泛化关系中，一个参与者的抽象描述可以被一个或多个具体的参与者所共享
  - 如：系统中经理可以参加雇员的所有用例

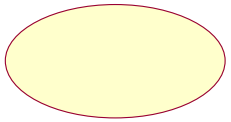


文档化参与者

- 参与者的文档没有固定的格式，但至少应该包含如下信息
  - 描述：为每一个参与者提供一个简要的描述，准确地描述该参与者的所扮演的角色和职责
  - 基本特征：针对参与者的职责范围、物理环境、使用习惯、用户数量和类型、使用系统的频率等特性进行说明
  - 相关的涉众和典型用户

用例

- 关键词：价值
- 简洁定义：参与者使用系统达到某个目标
- 定义
  - 一个用例定义一组用例实例（场景）
  - 用例实例是系统执行的一系列动作，这些动作将生成特定参与者可观测的结果值

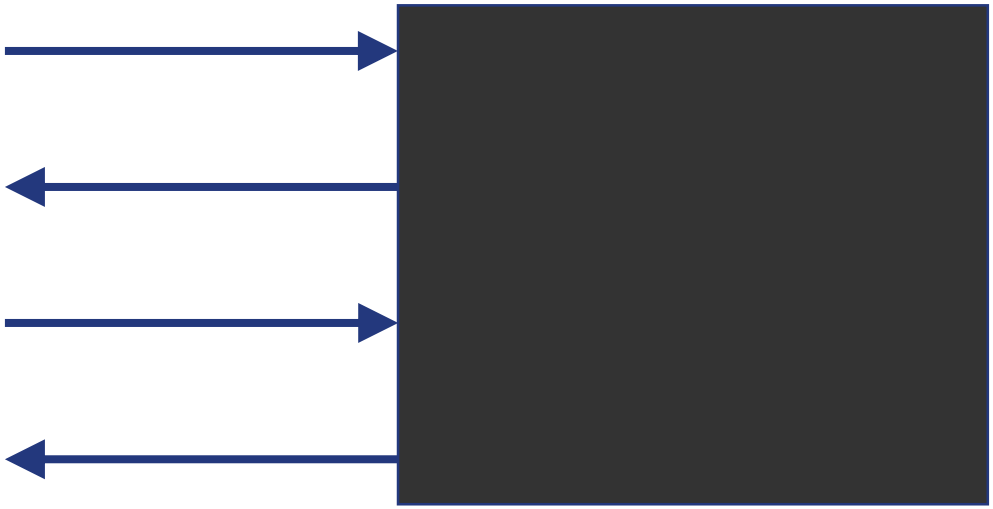


记住了，我是(系统)用例

用例要点

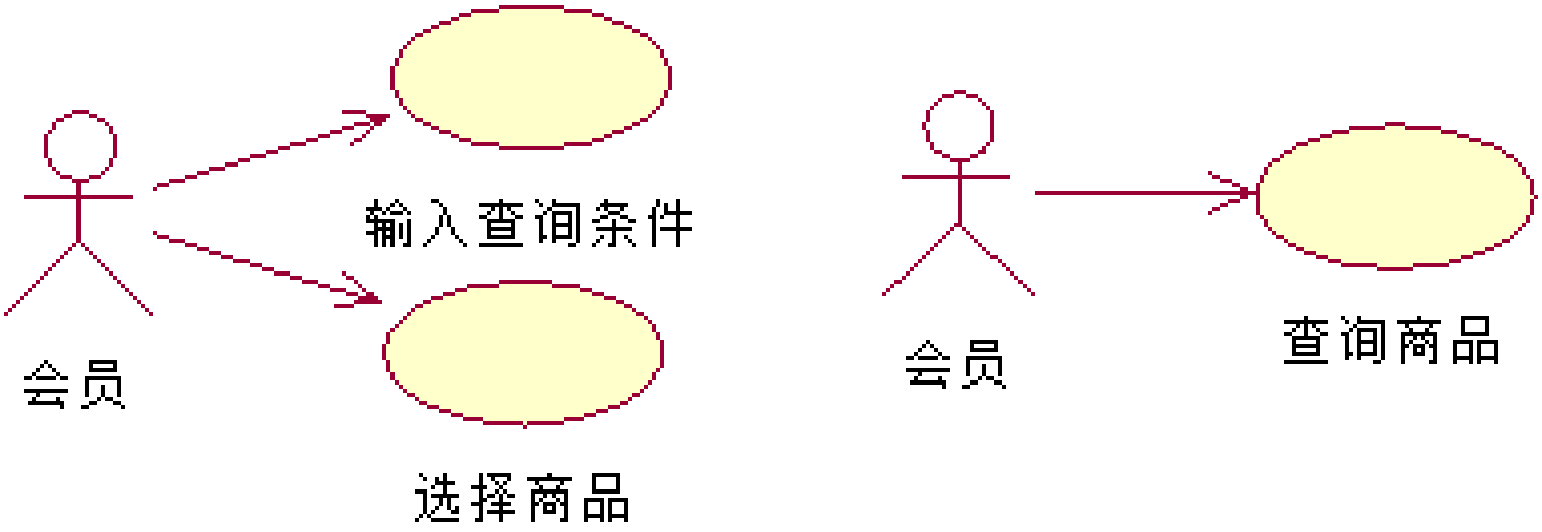
- 可观测→用例止于系统边界
- 结果值→用例是有意义的目标
- 系统执行→结果值由系统生成
- 由参与者观测→业务语言、用户观点

要点：用例止于系统边界



描述交互，而不是内在的系统活动

要点：有意义的目标



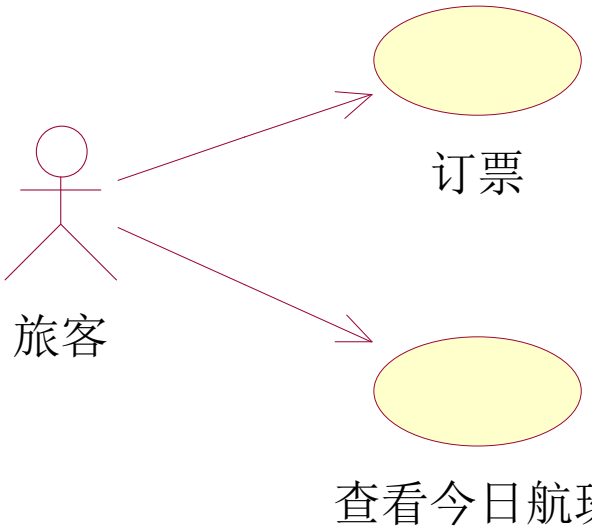
要点：结果值由系统生成

系统需要处理的，由系统生成

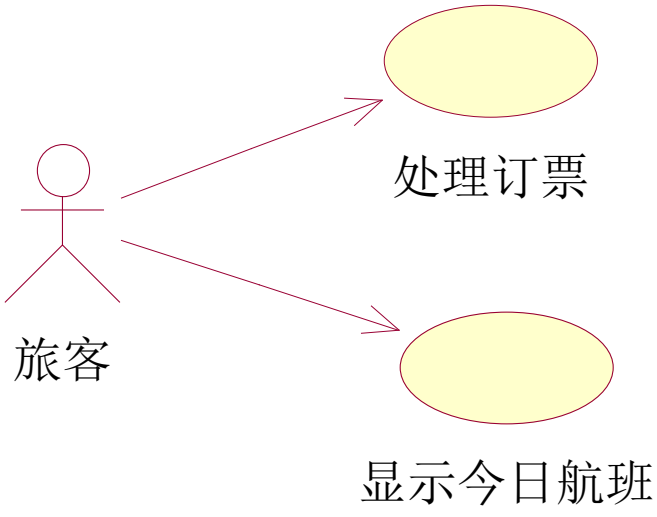
要点：业务语言而非技术语言

- 用户词汇，而不是技术词汇
  - 如：发票，商品，洗衣机
  - 而不是：记录，字段，COM，C++等

要点：用户观点而非系统观点



用户观点



系统观点

用例 VS. 功能

- 呼叫某人
- 接听电话
- 发送短信
- 记住电话号码
- ……

用户观点

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

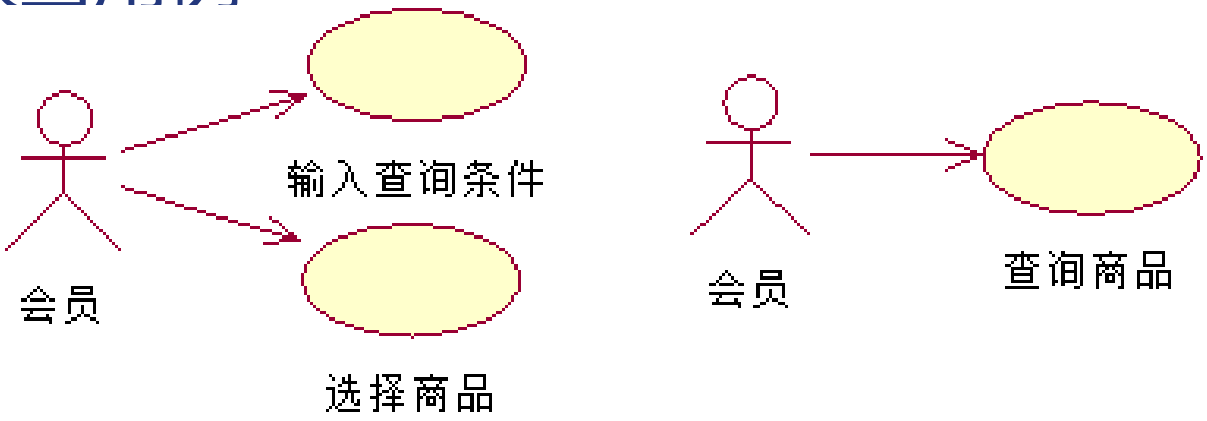
系统观点

用例粒度

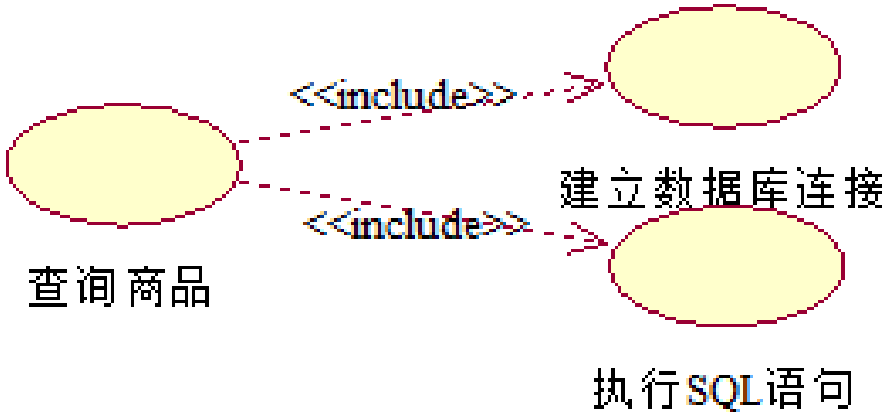
- 用例是一组用例实例的抽象；其内部要有路径，路径要有步骤
- 最常犯错误：粒度过细，陷入功能分解
  - 通过执行用例，参与者完成想做的事情(最终的目的)，并为参与者产生所需要的价值
  - 过细的粒度，一般都会导致技术语言的描述，而不再是业务语言

用例粒度（续）

- 把步骤当用例

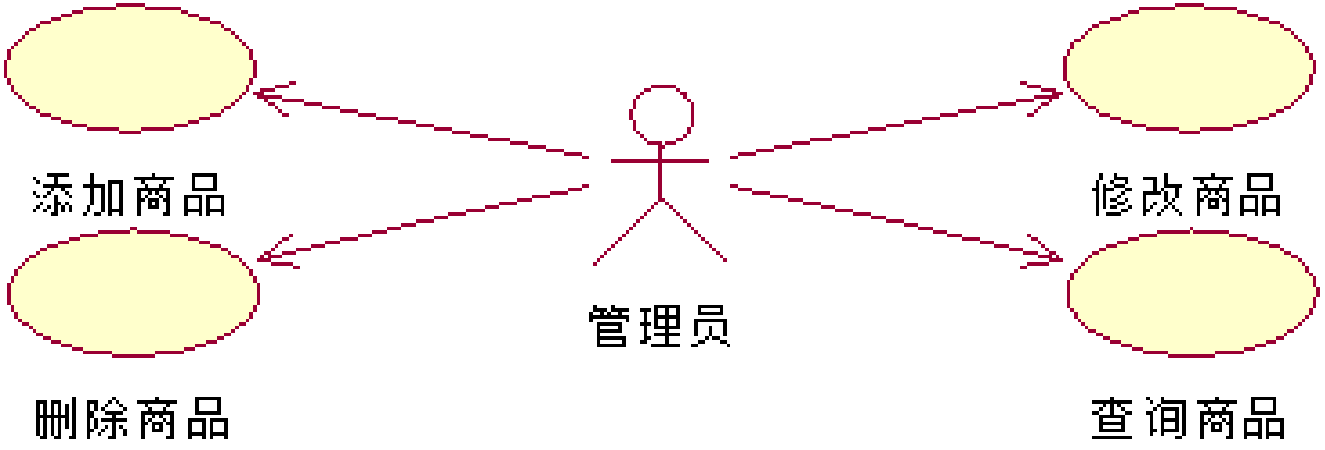


- 把系统活动当用例



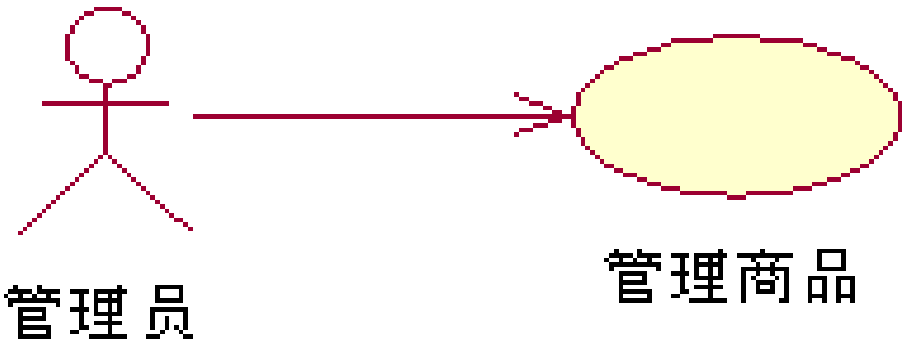
用例粒度：四轮马车

- 四轮马车CRUD：C(Create)、R(Read)、U(Update)、D(Delete)
  - 所有业务最终对会成为CRUD?
  - CRUD能为Actor提供价值?
  - CRUD掩盖业务，锐变成关系数据库的建模
    - “系统就是数据的增删改查”
    - 关心数据的存储和维护，反而忽略了用户的目的



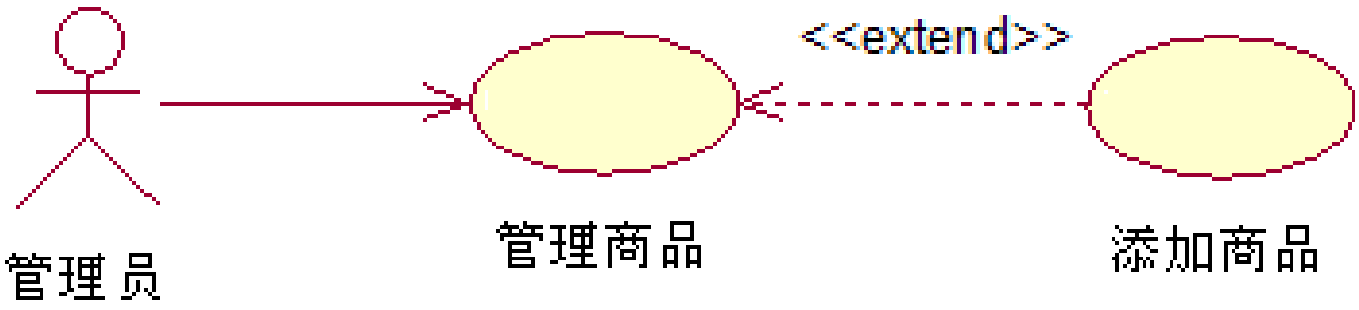
用例粒度：四轮马车（续）

- 如果确实是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”目标
  - 甚至很多种的基本数据管理都可以用一个用例表示



用例粒度：四轮马车（续）

- 灵活处理CRUD



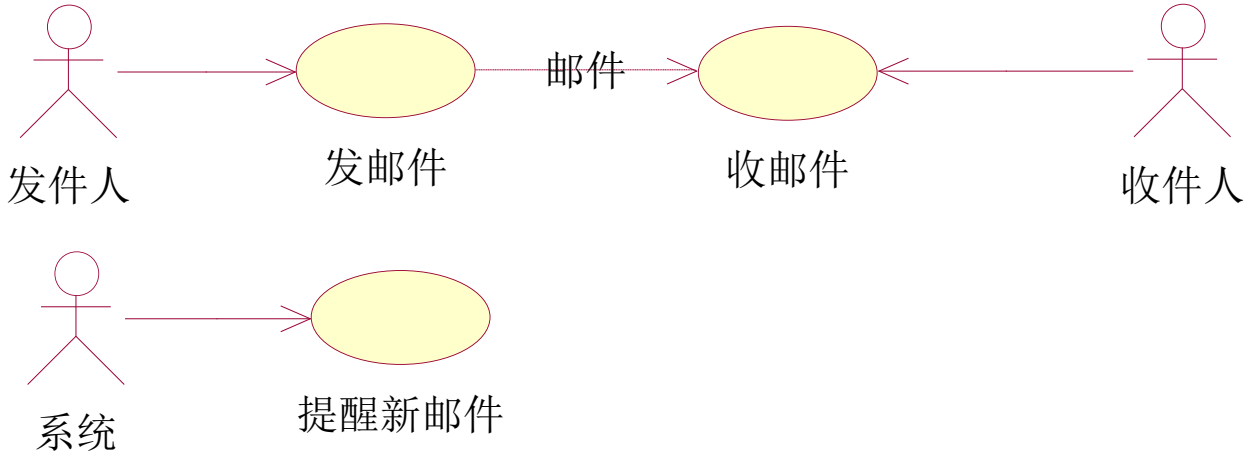
可以把包含复杂交互的路径独立出去形成用例

确定用例

- 从参与者的角度入手，通过分析参与者使用系统达成的目标来识别用例
  - 参与者的日常工作是什么
  - 参与者在业务中承担什么样的作用
  - 参与者是否生成、使用或删除系统相关信息
  - 参与者是否需要把外部变更通知系统
  - 系统是否需要把内部事情通知参与者，通知参与者过程就是系统用例的行为
  - 是否存在进行系统维护的用例
- 用例数量的参考基准
  - 1个系统中存在十几个用例(或更少)
  - 1个用例中有多个用例实例(场景)

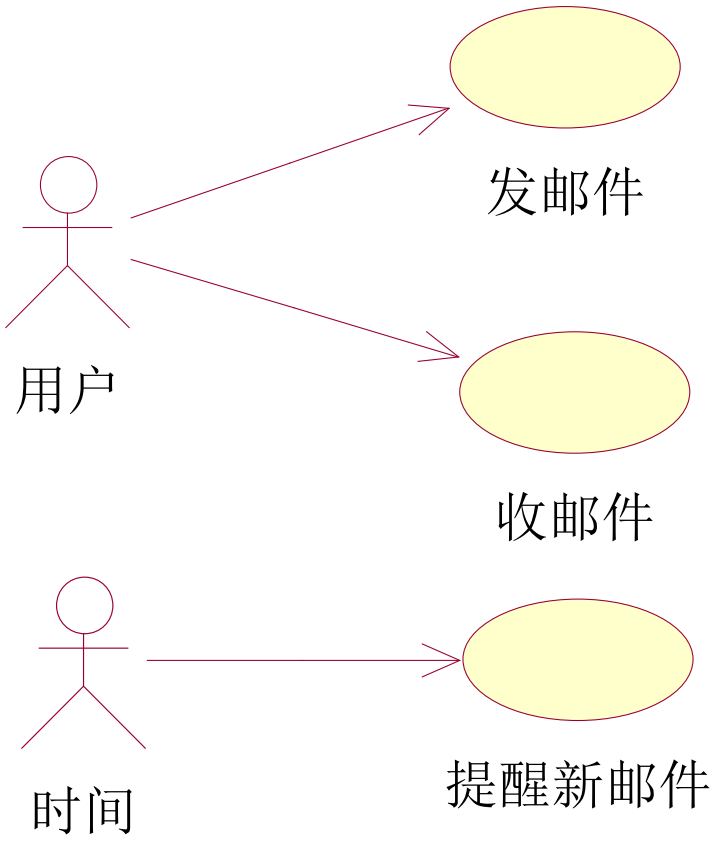
思考：识别用例

- Email客户端（如：outlook），A在北京发邮件给上海的B，系统提醒B你有“新邮件”，B收邮

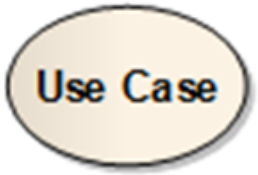
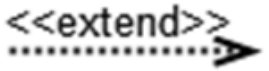
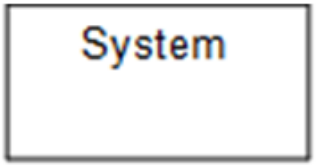


用例是一个完整的交互
用例之间没有顺序的关系

思考：识别用例（续）

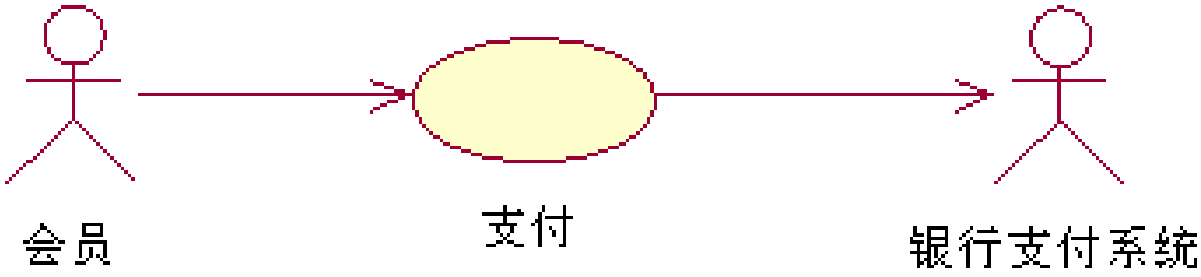


用例图

结构元素	图形符号	关系元素	图形符号
用例 (Use Case)		关联 (Association)	
参与者 (Actor)		扩展 (Extend)	
系统边界 (System Boundary)		包含 (Include)	
		泛化 (Generalization)	
注释 (Note)		注释连接 (Note link)	

关联关系：参与者和用例

- 关联关系：参与者参与用例的执行
  - 箭头（关联的方向）并不代表数据流或业务流的方向
  - 箭头代表通信的发起方
    - 箭头由通信的主动方指向被动方，或者说不带箭头的一方会受到带箭头一方的影响
    - 不带箭头意味着没有考虑这种影响的方向



建模实例：旅游业务申请系统

- 阅读“旅游业务申请系统”问题陈述
  - 识别系统参与者
  - 识别系统用例
  - 绘制用例图

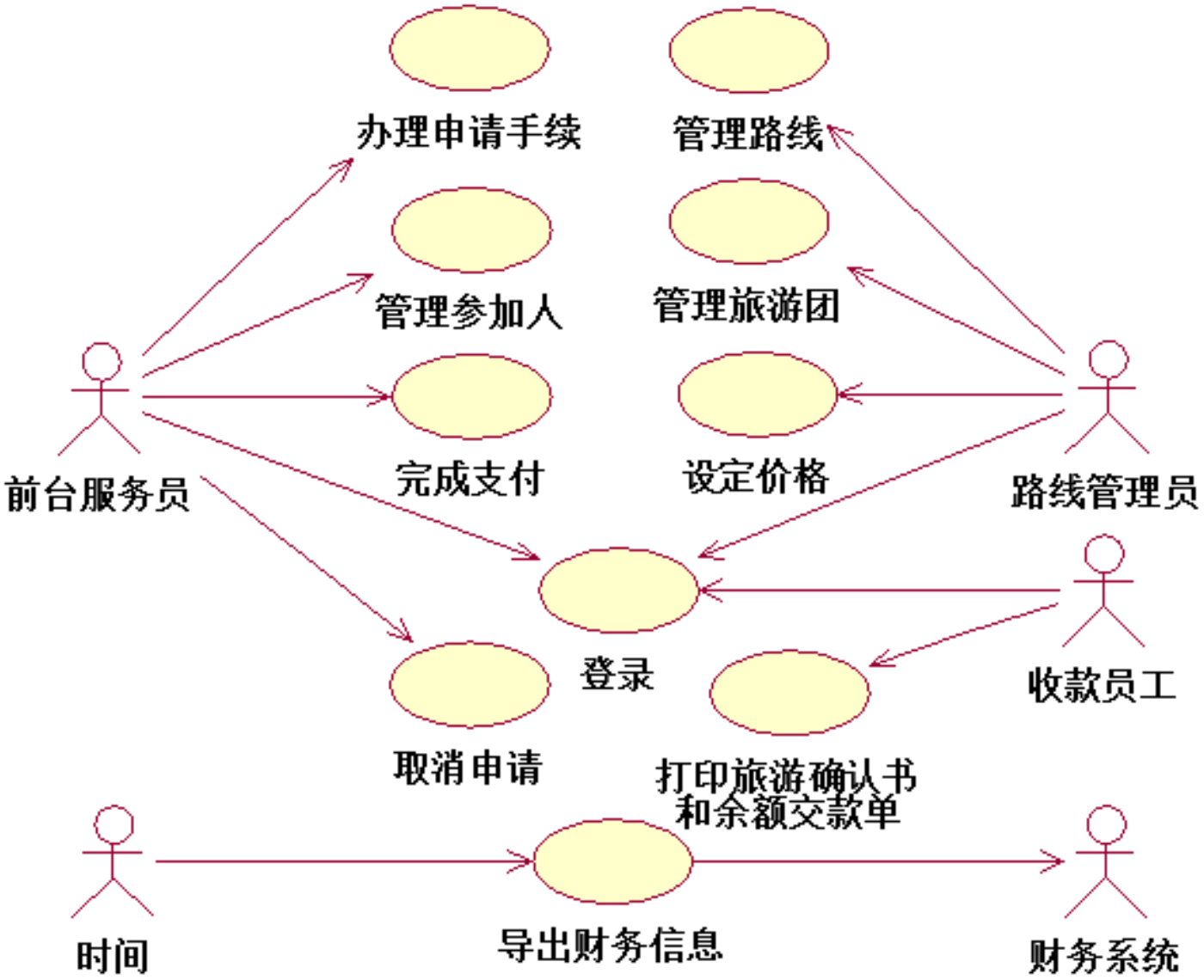
获取系统参与者

抽取角度	外部事物种类	主要日常工作	使用目标系统职责	参与者	典型代表
相关用户	前台招待顾客的员工	洽谈客户事宜并为客户办理各种申请和取消手续、完成费用支付	办理申请手续以及相关的取消、支付等后续业务	前台服务员	具体用户代表
	负责催款的员工	打印和邮寄旅游确认书和交款单	打印旅游确认书和交款单	收款员工	…
	旅行社内的会计人员	财务记帐	不使用本系统，不是参与者		…
	宣传和路线管理员工	制作宣传资料、定期维护旅游路线和活动	维护旅游路线和旅游活动	路线管理员	…
其他外部事物	财务系统	记帐等财务操作	接收本系统中与现金相关的财务信息	财务系统	…
	外部激励	关注或影响系统的运行	定期自动导出财务信息	时间	

从参与者的角度获取用例

参与者	主要工作	是否使用系统	用例
前台服务员	向申请人介绍申请情况	否	
	为申请人办理申请手续	是	办理申请手续
	对申请参加人的增、删、改、查等日常维护	是	管理参加者
	记录申请人支付信息	是	完成支付
	为申请人取消申请	是	取消申请
收款员工	打印旅游确认书和余额交款单	是	打印旅游确认书和余额交款单
	邮寄旅游确认书和余额交款单	否	
路线管理员	制作宣传资料	否	
	设计旅游路线	是	管理路线
	设计旅游团（活动）	是	管理旅游团
	调整旅游团价格	是	设定价格
财务系统	记账等财务操作	否	
	接收与现金相关的财务信息	是	导出财务信息（被动）
时间	定期导出财务信息	是	导出财务信息
辅助用例	系统要区分各种不同的用户身份，并提供不同的功能	是	登录

用例图


办理申请手续
管理路线
管理参加人
管理旅游团
完成支付
设定价格
前台服务员
路线管理员
登录
收款员工
取消申请
打印旅游确认书
和余额交款单
时间
导出财务信息
财务系统

内容安排

1. 用例建模基础
2. 文档化用例
3. 重构用例模型

用例文档

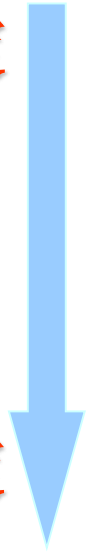
- 用例文档：描述用例与外界交互的规格说明书
  - 需求的核心内容，而用例图作为用例文档的索引图
  - 进一步的精度：有层次的文档
  - 文档中每一句话都有其价值

用例图是骨架，而用例文档则是其内在的肉

有层次的需求组织形式

- 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50元的倍数）

低精度，稳定

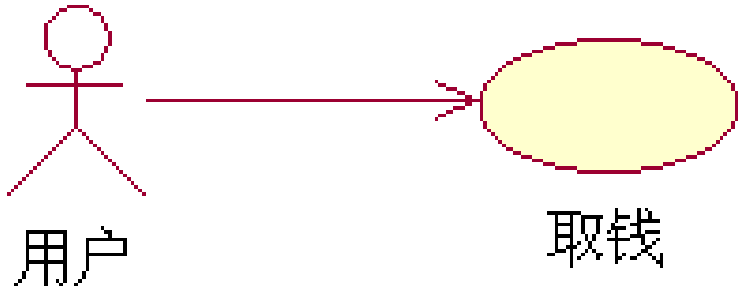


高精度，不稳定

用例文档的组成

- 用例名、简要描述
- 参与者与涉众
- 相关用例
- 前置条件、后置条件
- 事件流
  - 基本路径
  - 备选路径
- 补充约束
  - 数据需求、业务规则
  - 非功能需求、设计约束
- 待解决问题
- 相关图(用例图、活动图、类图)

参与者与涉众



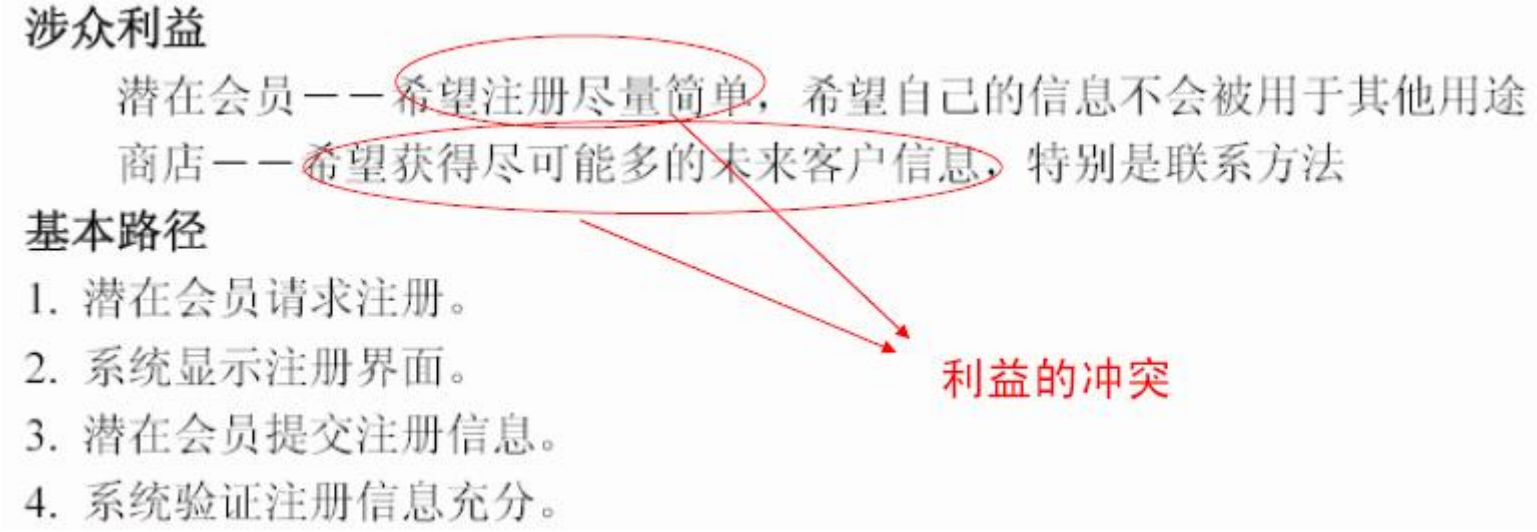
- 同样是用户“取钱”
  - 为什么家里的抽屉不用密码，取款机要用?
  - 为什么取了钱以后要“系统扣除帐户金额”
- 还有一些因素没有考虑…

涉众

- 涉众是受系统影响的，有自己主张的人或组织，可能的涉众有：
  - 最终用户、客户、政府、法律
  - 开发人员、管理人员、竞争对手、…
- 用例平衡涉众之间的利益
- 对于用户在ATM取钱的用例
  - 用户：希望方便
  - 银行：希望安全
  - 法律：保护财产

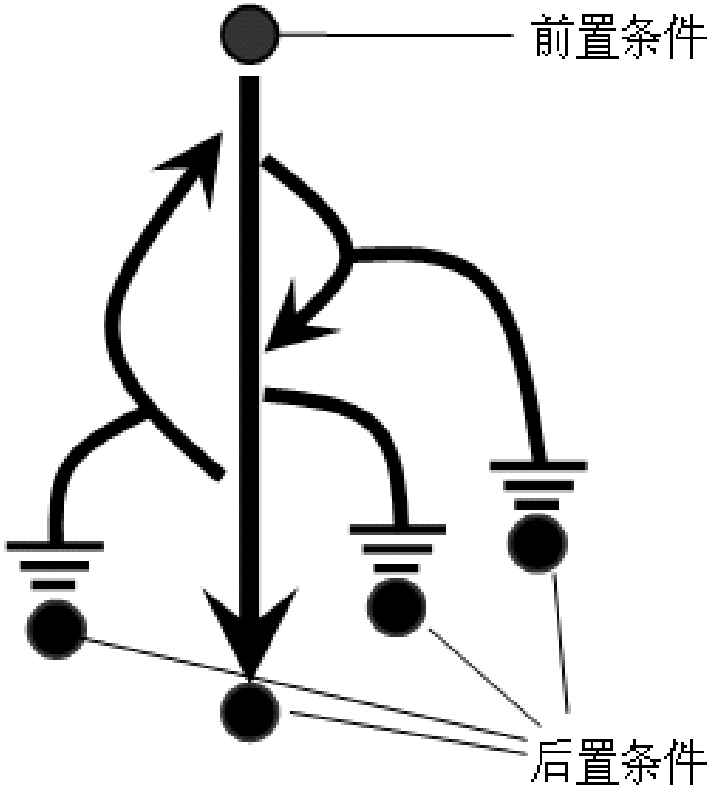
用例文档体现涉众利益

- 用例相当于参与者在台上表演，而最重要的是台下的观众(涉众)的利益
- 涉众有轻重缓急，甚至存在利益冲突
- 编写用例文档的过程就是描述如何满足涉众之间的利益，达到涉众利益的平衡



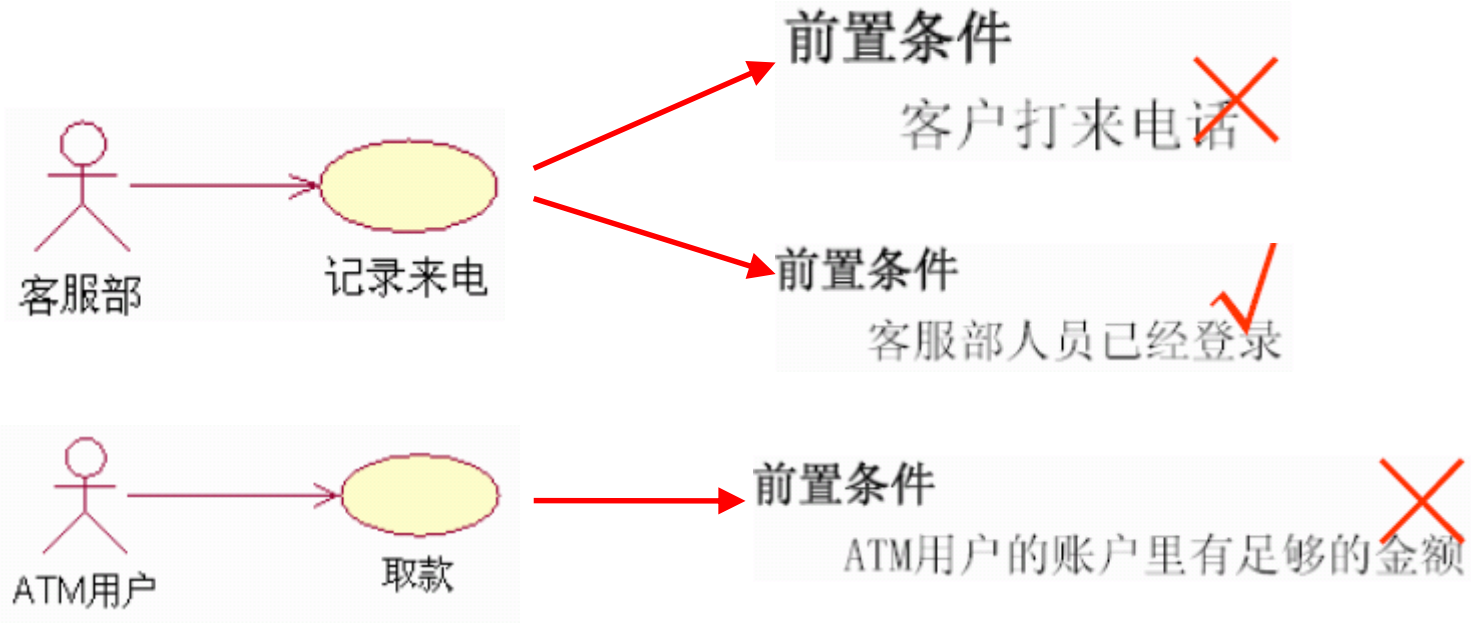
前置和后置条件

- 前置条件约束在用例开始前系统的状态
  - 作为用例的入口限制，它阻止参与者触发该用例，直到满足所有条件
  - 说明在用例触发之前什么必须为真
- 后置条件约束用例执行后系统的状态
  - 用例执行后什么必须为真
  - 对于存在各种分支事件流的用例，则可以指定多个后置条件

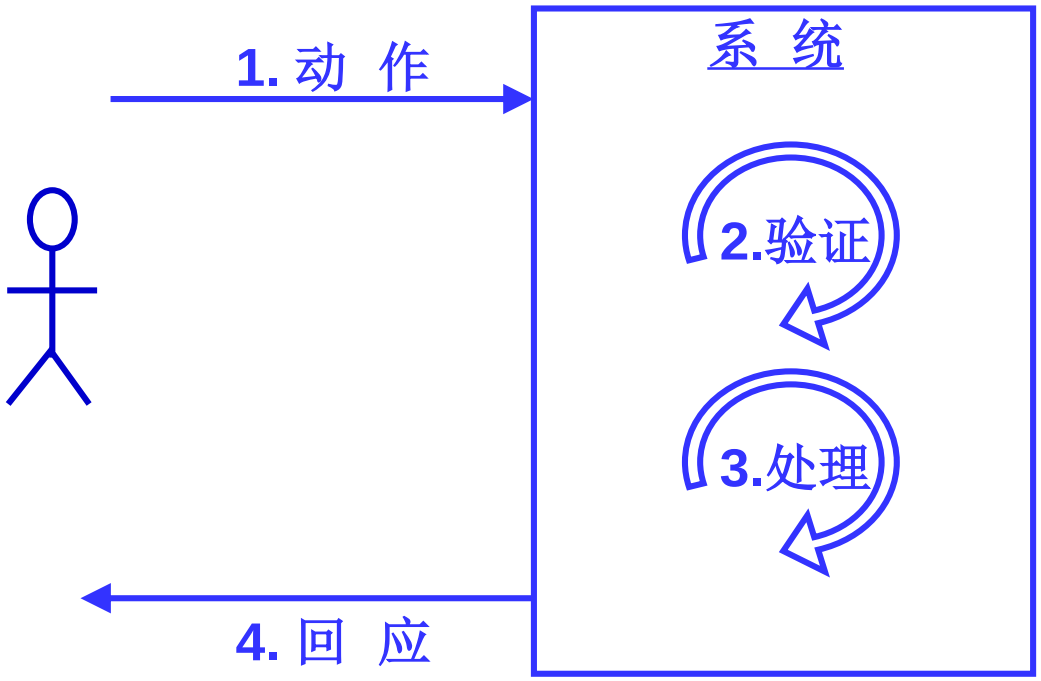


定义前置和后置条件

- 只有在用例的使用者将这些条件视为附加价值是才使用
- 条件必须是系统可以感知的
- 前置条件必须是在用例执行前就可以感知



事件流：描述用例交互



重点写：1和4（可观测的、体现客户利益的文字）

事件流要点

- 事件流描述要使用户和开发人员互相理解用例的功能，要注意以下几点：
  - 使用业务语言：使用用户平时所使用的语言进行描述
  - 重点描述参与者与系统交互的信息
  - 不使用“例如”、“等”不清晰的表达
  - 不要过多的考虑界面细节
  - 不要描述系统内部处理细节，要描述从系统外部所看到的活动
  - 要明确描述用例的开始和结束
  - 不仅需要描述基本事件流，还需要考虑备选事件流

例1：使用自然语言（业务语言）

- 技术语言：无法与用户沟通
  - 系统通过ADO建立数据库连接，传送SQL查询语句，从“商品表”查询商品的详细信息…
- 自然语言(业务语言、用户语言)
  - 系统按照查询条件搜索商品的详细信息

例2：描述参与者与系统交互过程

- 以参与者或系统作为主语描述
  - 参与者……
  - 系统……
- 示例
  - 出纳员接收顾客的付款—顾客的付款数可能高于商品总额
  - 出纳员录入顾客所付的现金总额
  - 系统显示出应找还给顾客的余额，打印付款收据

分支和循环的描述

- 分支：放到备选路径中
  - 参与者的选择
  - 另一条成功线路
  - 系统进行验证
  - ……
- 循环：直接描述

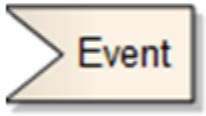
两种类型的事件流

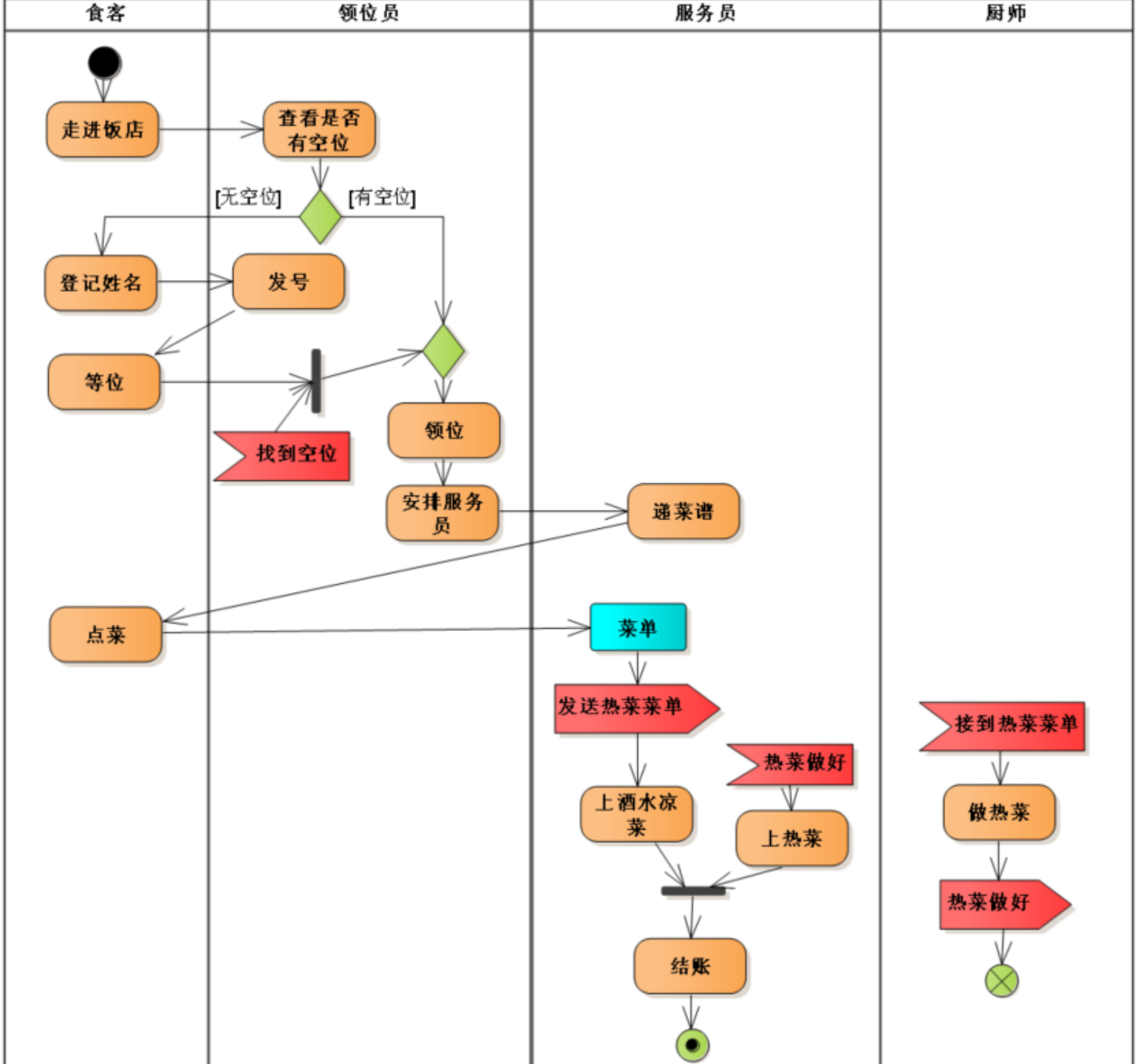
- 基本事件流
  - 用例的主路径、愉快路径（Happy Path）
  - 通常用来描述一个理想世界，即没有任何错误发生的情况
  - 复杂的基本流可以分解成多个子流
- 备选事件流
  - 基本事件流中的分支或异常情况
  - 注意如何与基本流衔接

利用活动图可视化事件流

- 活动图（Activity Diagram）
  - 一种行为模型，描述活动或动作之间的流程，强调行为的执行序列和条件
- 对于包含复杂场景的用例，可以利用活动图可视化业务流程

活动图

元素	图形符号	元素	图形符号
活动/动作 (Activity/Action)		起点 (InitialNode)	
对象 (Object)		终点 (FinalNode)	
发送事件 (SendEvent)		流结束 (FlowFinal)	
接收事件 (AcceptEvent)		分叉/合并 (Fork/Join)	
分区 (Partition)		控制流 (ControlFlow)	
决策点 (Decision)		对象流 (ObjectFlow) (UML1.x 中为虚线)	


食客
领位员
服务员
厨师
走进饭店
查看是否有空位
[无空位]
[有空位]
登记姓名
发号
等位
找到空位
领位
安排服务员
递菜谱
点菜
菜单
发送热菜菜单
热菜做好
上酒水凉菜
上热菜
结账
接到热菜菜单
做热菜
热菜做好

用例文档中的补充约束

- 用例重点在于描述功能需求，而其它方面的补充约束：
  - 与特定用例相关的补充约束，作为该用例文档中一部分来描述
  - 一些全局性的补充约束，单独形成一份独立的文档，如“补充需求规约”文档
- 补充约束
  - 数据需求
  - 业务规则
  - 非功能需求
  - 设计约束

场景（Scenario）

- 场景：用例的一次执行，按照特定条件执行事件流时的具体流程、执行示例
  - 用例中的每个分支潜在产生一个独立场景
  - 将用例的部分基本流和若干备选流结合起来，就可以构成不同的使用场景
  - 基本场景 + 辅助场景

内容安排

1. 用例建模基础
2. 文档化用例
3. 重构用例模型

重构用例模型

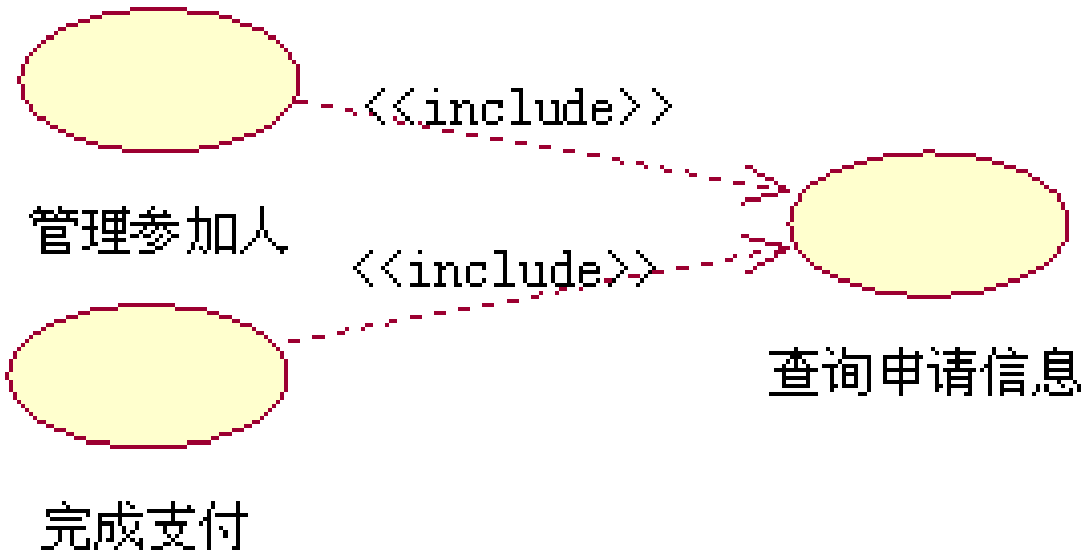
- 利用用例建模高级技术重构用例模型
  - 用例关系
    - 通过用例关系将复杂的用例进行适当的分解，以便于提高需求的复用性和可扩展性等，从而使用例模型的结构更合理
  - 用例分包
    - 将相关的用例打包，通过分包的方式可以将用例图分层表示，以用于大规模系统的用例建模
  - 用例分级
    - 可以根据用例的重要程度进行分级，以便后续迭代计划的制定，高级别的用例优先考虑

通过关系整理用例

- Include(包含) <<include>>
  - 基用例中复用被包含用例的行为
  - 提取公共步骤，便于复用
- Extend(扩展) <<extend>>
  - 通过扩展用例对基用例增加附加的行为
- Generalization(泛化)
  - 派生用例继承泛化用例的行为并添加新行为

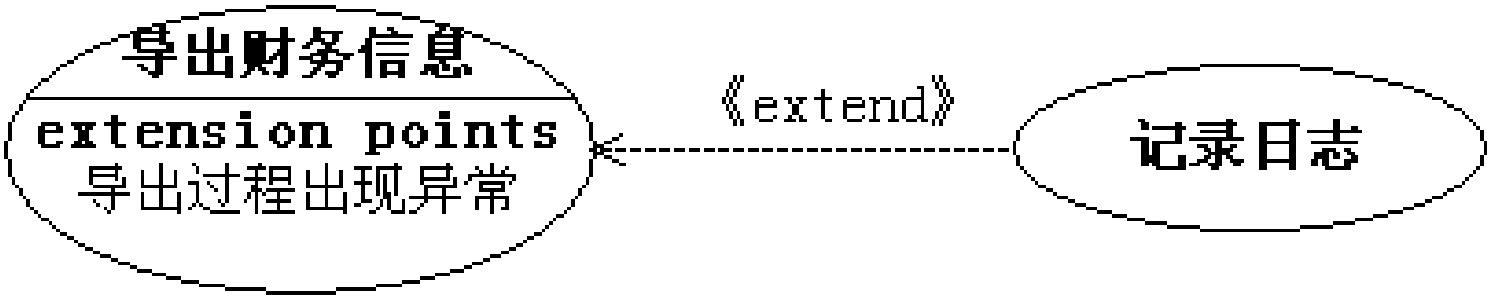
用例关系：包含

- 包含：表示某个用例中包含了其他用例的行为
  - 从两个或多个用例行为中提取公共部分的能力，主要用于支持用例行为的复用

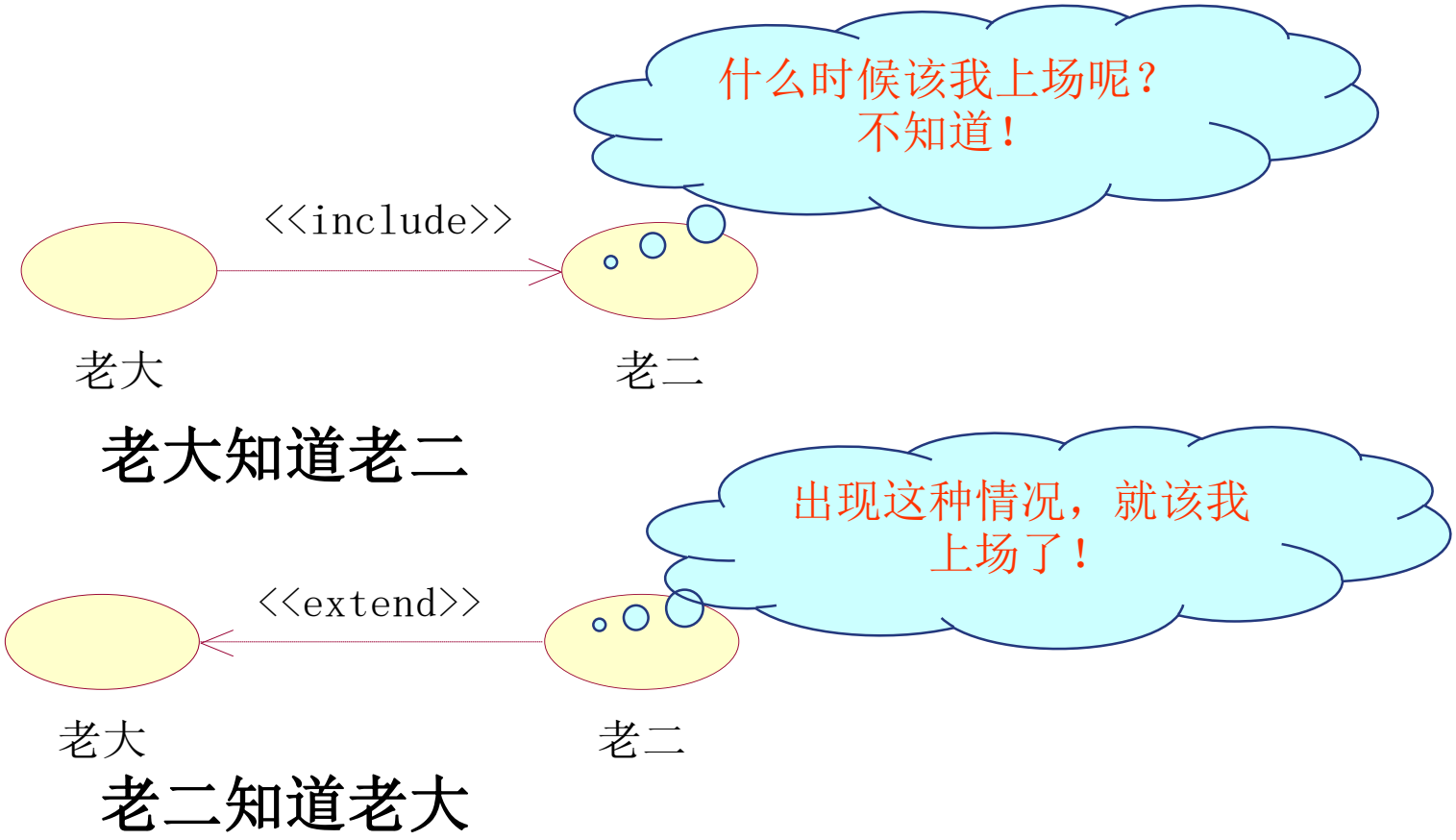


用例关系：扩展

- 扩展：某个用例在特定情况下，包含其他用例(扩展用例)的行为，表示功能被扩展
  - 为了将基用例的一些特殊情况分离出来，在保持基用例本身相对完整的情况下处理这些特殊行为
  - 即不改变基用例，对基用例的行为进行扩展



扩展 VS. 包含

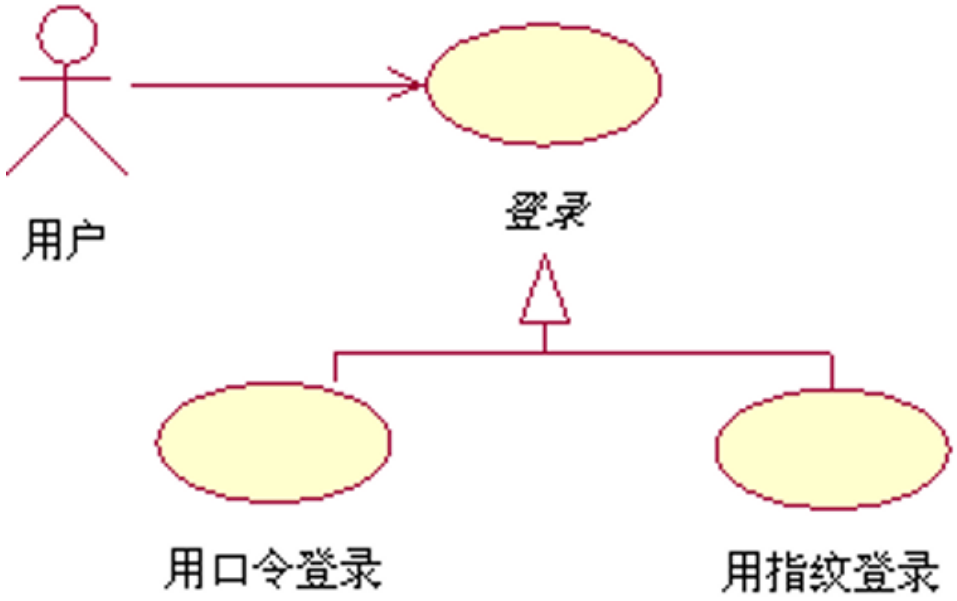


扩展 VS. 包含

出发点不同	包含关系	便于子用例流的复用
	扩展关系	通过扩展点在不影响基用例的情况下附加行为
达到的效果不同	包含关系	基用例中的一部分业务放在子用例中
	扩展关系	基用例处理一般情况，一些特殊业务放在子用例中
执行子用例方式不同	包含关系	基用例中直接引用子用例
	扩展关系	主用例达到一定条件触发扩展点，子用例通过扩展点触发
使用方式不同	包含关系	基用例不够完整，一般要联合子用例为参与者提供价值
	扩展关系	基用例相对完整，可以单独为参与者提供价值
依赖方向不同	包含关系	主用例依赖子用例，子用例相对独立
	扩展关系	子用例依赖主用例，主用例相对独立

用例关系：泛化

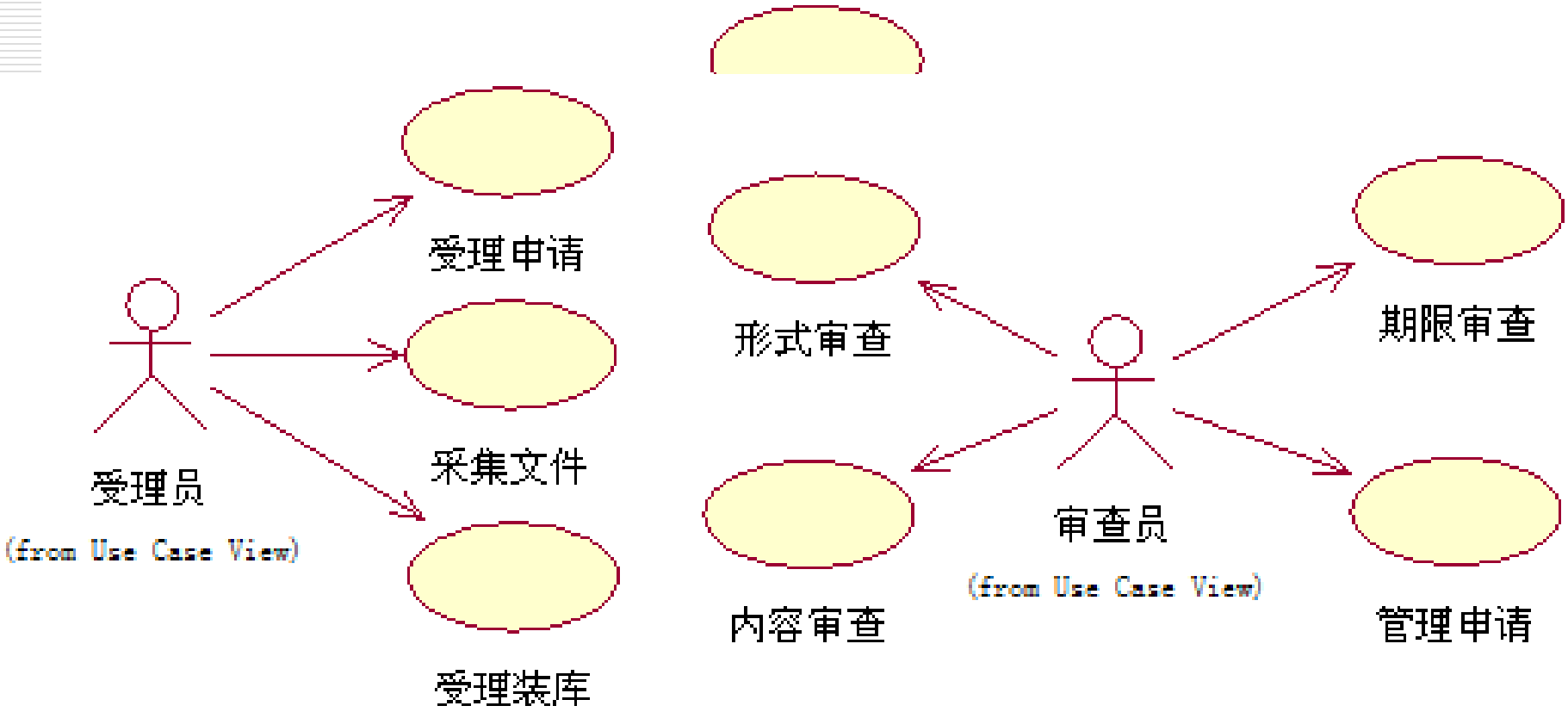
- 泛化：表示子用例继承了父用例
  - 用例间的泛化关系表明子用例继承父用例中定义的所有属性、行为序列和扩展点，并且参与父用例中所有的关系



用例分包

- 对于大规模系统，当用例数量很多时，难以在一个层次上一次性描述所有用例
- 用例分包
  - 让系统的用例图能够更为清晰的表现出系统的业务逻辑关系和层次
  - 对系统进行模块的分割，这种分割将影响到系统今后的开发、系统的最终表现形式
- 常见的分包方式
  - 基于业务主题的分包
  - 按照参与者分包
  - 基于开发团队的分包
  - 基于发布情况的分包

利用分包机制组织用例模型



用例分级

- 用例分级的一个基本原则
  - 高级别用例是那些对系统核心架构影响最大的用例
- 提高用例级别的特性：
  - (1) 对架构设计有重要影响的用例，如在领域层中增加多个类的用例或者需要持久化的用例
  - (2) 体现系统核心业务流程的用例
  - (3) 存在开发风险的用例
  - (4) 涉及新技术或需要创新的用例
  - (5) 能够尽快投入使用并带来直接经济效益的用例

示例：用例分级

- 可以使用一个简单的但是有些不精确的分类方法，如将用例划分成高、中、低三个等级

用　　例	优先级	分 级 原 因
预订房间	高	项目主要远景，代表系统核心业务流程
取消预订	中	重要流程，保证主流程的完整性
登录	中	影响系统权限机制，存在安全性架构机制
调整价格	低	独立于其他用例，且对系统架构影响较小

- 其他的分级方式
  - Must：必须做
  - Should：应该要做
  - Could：可以做
  - Would Not：不做

示例：用例分级

- 依照上述的影响用例级别的特性给用例打分（特性也可能带有权值）

用例	（1）架构	（2）核心业务	（3）风险	（4）新技术	（5）经济效益	总分
办理申请手续	5	5	4	1	5	20（高）
完成支付	4	3	5	3	4	19（高）
导出财务信息	5	3	5	1	4	18（高）
打印旅行确认书和余额交款单	2	4	1	1	3	11（中）
设定价格	1	2	0	1	4	8（低）
……						

题外话：用例与需求规约

- 用例模型由用例图、参与者文档和用例文档组成，通过这三部分来表示系统需求
  - 可以合并为一个文档，也可为每个用例单独编制文档
- 补充规约文档
  - 需求规约中还应该包含对数据要求、非功能需求、设计约束、用户界面、验收标准等内容的约定
  - 可以是单独的文档，可以与用例模型合并在一个完整的文档中
- 数据规约文档
  - 分析阶段编制，本次作业不包含

用例建模的适用场合

- 用例是从参与者角度捕获系统功能，当系统只有一个或没有参与者时，显然不是非常有效的
- 用例捕获功能需求，因此对于系统的非功能需求不是有效
  - 当遇到下述情况时，用例是需求捕获的最好选择
    - 系统由功能需求所主导
    - 系统具有很多类型的用户，系统对他们提供不同的功能
    - 系统具有很多接口
  - 当遇到下述情况时，用例是一个糟糕的选择：
    - 系统由非功能需求所主导（如：google）
    - 系统具有很少的用户
    - 系统具有很少的接口（非内部功能）
    - 如：嵌入式系统、算法复杂但接口少的系统等

从用例到用户故事

- 用户故事（Use Story）：一种轻量级的、有效的用户需求描述方式
- 都是从用户角度描述用户渴望得到的功能
- 目的不同
  - 用例：为了在用户与开发团队之间形成文档化的规约
  - 用户故事：是帮助发布和迭代计划的制定，是作为占位符，为有关详细用户需求的对话服务的
- 描述的粒度不同：
  - 用例：规范化、系统化，包含不同的用例场景
  - 用户故事：以用户使用场景为基础，轻量级的需求描述方式，是敏捷方法中主流的需求建模技术

用户故事

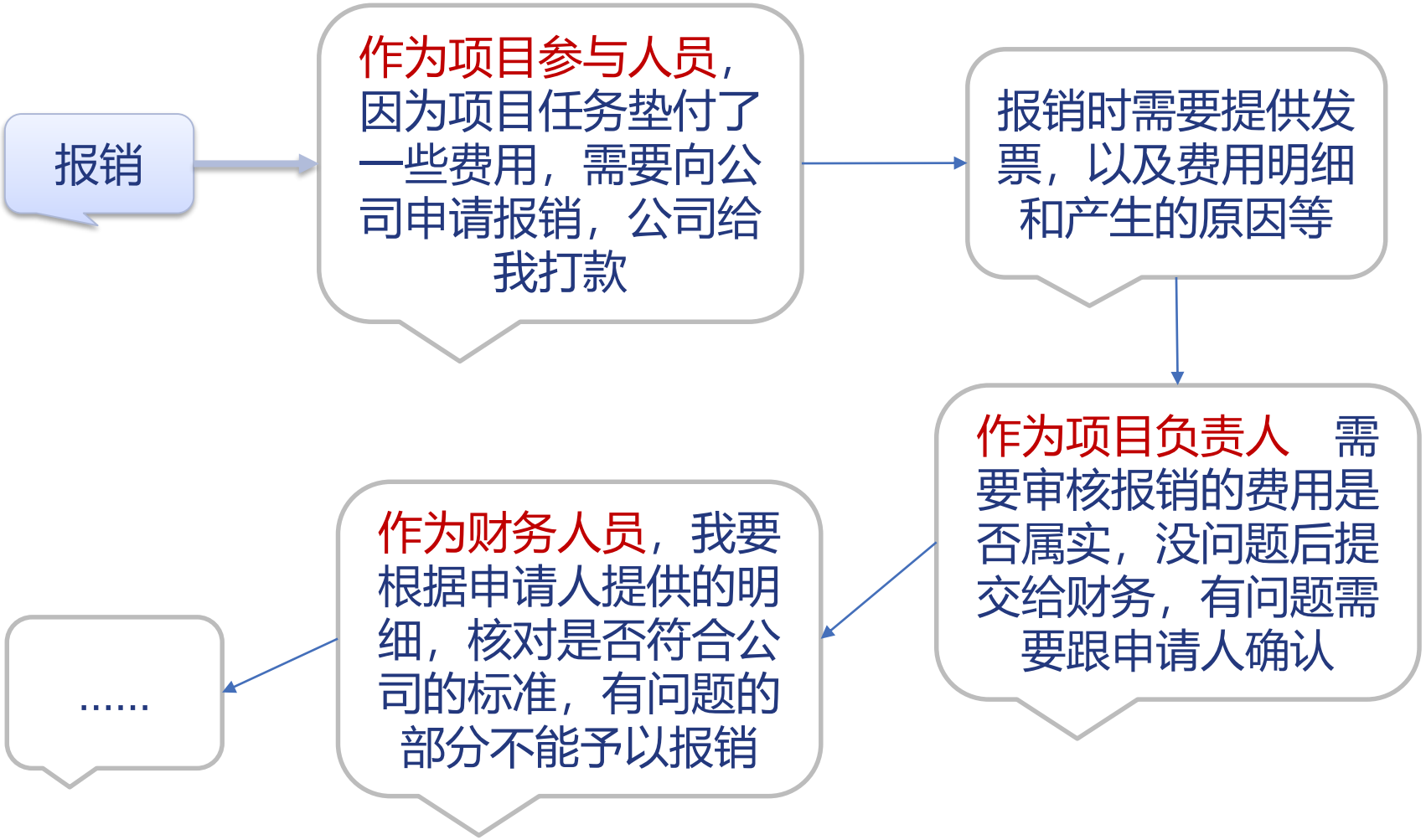
- 用户故事的描述方式

作为***（角色，Who），希望通过系统***（功能，What），以便达成***（商业价值，Why）。

  - 角色：谁要使用这个功能
  - 活动：需要完成什么样的功能
  - 商业价值：为什么需要这个功能，这个功能带来什么样的价值

- 3C
  - 卡片（Card）：故事的简短描述、工作量估算等
  - 交谈（Conversation）：细节来源于和客户的交流沟通
  - 确认（Confirmation）：通过验收测试确认用户故事被正确完成

提取用户故事



编写用户故事

编号	优先级	名称	描述	可接受条件
1		申请报销	作为项目参与人员，因为项目任务垫付了一些费用，我需要向公司申请报销，以便公司及时给我打款	1.报销时需要注明参与的项目信息 2.报销时需要注明具体产生哪些类型费用，产生的时间、原因和说明 3.需要提交附带费用的发票凭证，是否区分电子发票和纸件发票? 4.申请后提交项目负责人，一个项目是否有几个负责人？如果有，怎么处理?
2		审核报销	作为项目负责人，我需要审核大家提交的报销申请，检查报销申请中各项费用是否属实，以防止大家做错申请	1.检查属实的依据是：费用发生时间、参与人员，需要看这些明细 2.检查出不属实后，可通知申请人进行修改? 3.检查没问题后提交给财务核查
3		核准报销	作为财务人员，对项目负责人认可的申请，核实是否各项费用超标，是否超出项目预算，检查各类凭据的真实性。以便…	1.能够看到各类费用明细，并与标准核对 2.能够根据费用类型，检查是否超出项目预算费用 3.如何检查发票的真伪？对于纸质发票和电子发票如何处理?
4	…	…	…	…

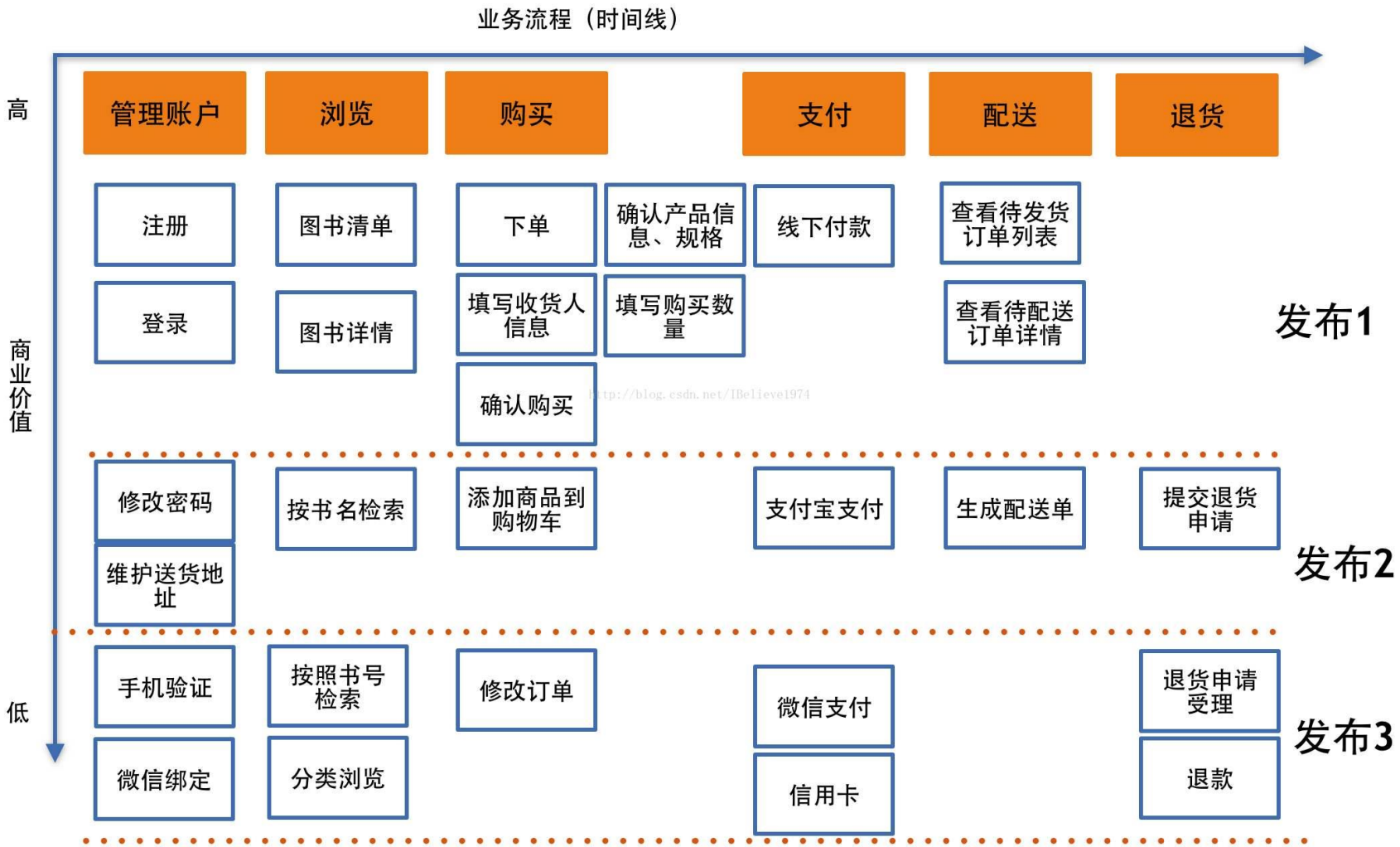
用户故事地图

- 用户故事地图（User Story Mapping）已经成为敏捷需求规划中的一个流行方法
- 用户故事地图可以将Backlog变成一张二维地图，而不是传统的简单列表
  - 参考文章：
    - http://www.woshipm.com/pd/2067818.html
- 可以尝试使用教学系统的中“实训”功能在线编制用户故事地图

用户故事地图

用户故事地图

故事地图示例 – 在线购书网站



Practice

- 作业2：需求建模
  - 进度安排
    - 9月20日之前：基于需求调研的成果，采用用例技术，构建系统用例模型（或者用户故事模型），编制需求规约
    - 9月20日：讲解需求模型，并针对其中的问题展开讨论
    - 9月20日之后：根据讨论结果，完成各个需求规则说明的细节
  - 提交
    - DDL：2022年9月23日晚上10:00
    - 交付物：系统需求规格说明书，以用例（或用户故事）为中心描述功能需求，单独描述系统的非功能需求