



IBM Software Group

## 定义清晰的系统功能 - 用例建模技术简介

IBM 软件部 华琦

**Rational** software



**ON DEMAND BUSINESS™**

© 2005 IBM Corporation

## 内容

- 需求开发体系
- 什么是用例建模
- 用例的开发过程
- 管理用例模型的复杂度



## 内容

- 需求开发体系
- 什么是用例建模
- 用例的开发过程
- 管理用例模型的复杂度



## Rational 需求管理解决方案解决的问题

- 需求开发
  - ▶ 如何方便收集客户需求
  - ▶ 如何准确理解客户所要解决的问题
  - ▶ 如何准确的描述需求
- 需求管理
  - ▶ 如何对需求进行量化管理和分析，包括方便查询、排序等
  - ▶ 如何实现需求的可追踪性，并有效追踪需求的变更所造成的影响
- 需求变更管理
  - ▶ 如何有效管理用户的变更请求



# 需求管理解决方案

## ■ 需求开发

- ▶ 使用变更管理平台，方便收集客户需求
- ▶ 使用建模工具 (WBI、Rose XDE、RSM / RSA) 进行业务建模，以准确理解客户所要解决的问题
- ▶ 使用模型（用例模型）和文档的有机结合，准确的描述需求

## ■ 需求管理

- ▶ RequisitePro提供统一整个团队的需求管理平台
- ▶ 在RequisitePro中使用属性描述需求，实现需求的量化管理和分析
- ▶ 使用 RUP 成熟的文档体系有效组织需求
- ▶ 使用RequisitePro的追踪矩阵，实现需求的可追踪性，有效追踪需求的变更所造成的影响

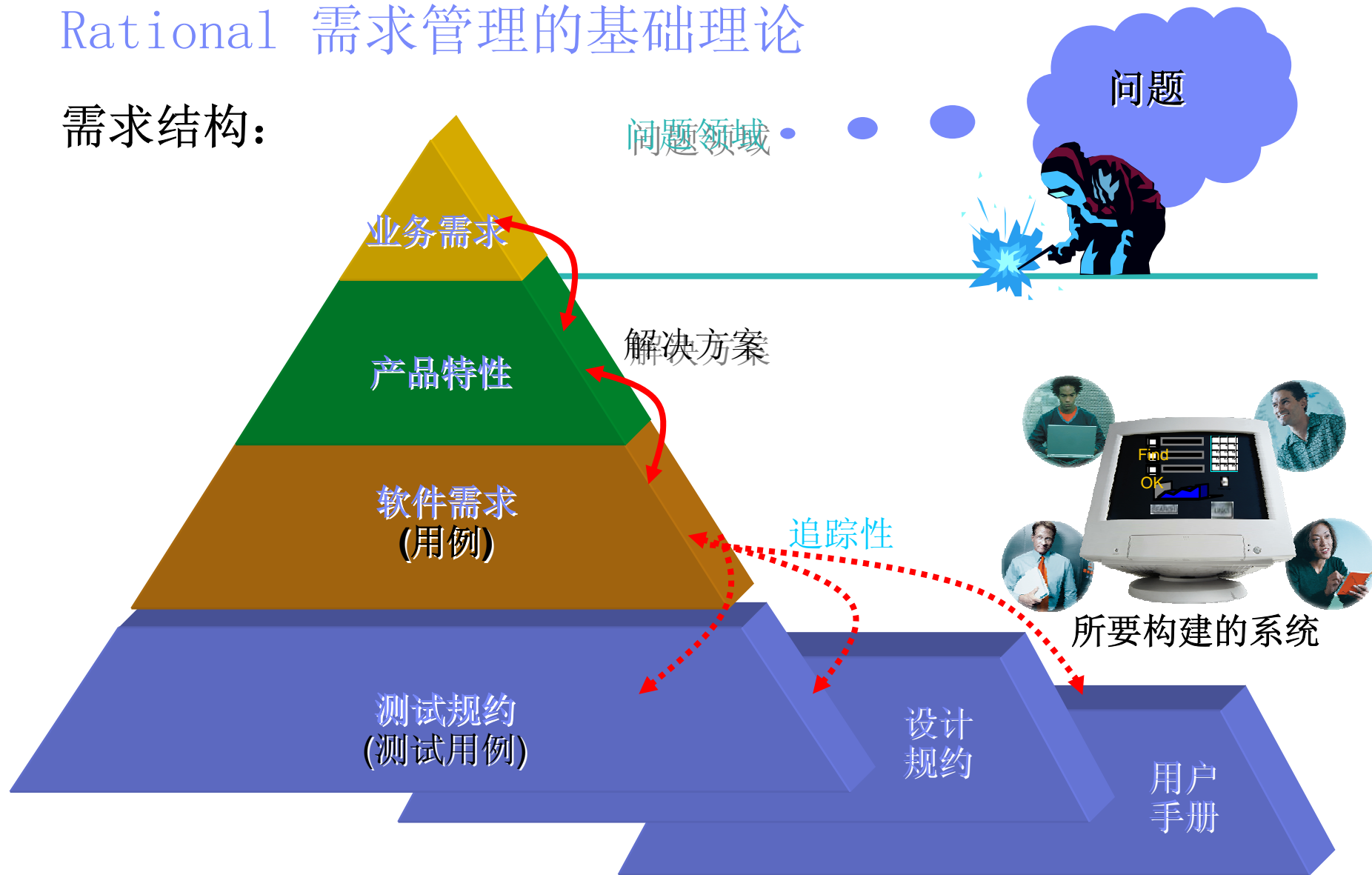
## ■ 需求变更管理

- ▶ 使用ClearQuest变更管理平台，建立标准变更管理流程，有效管理用户的变更请求

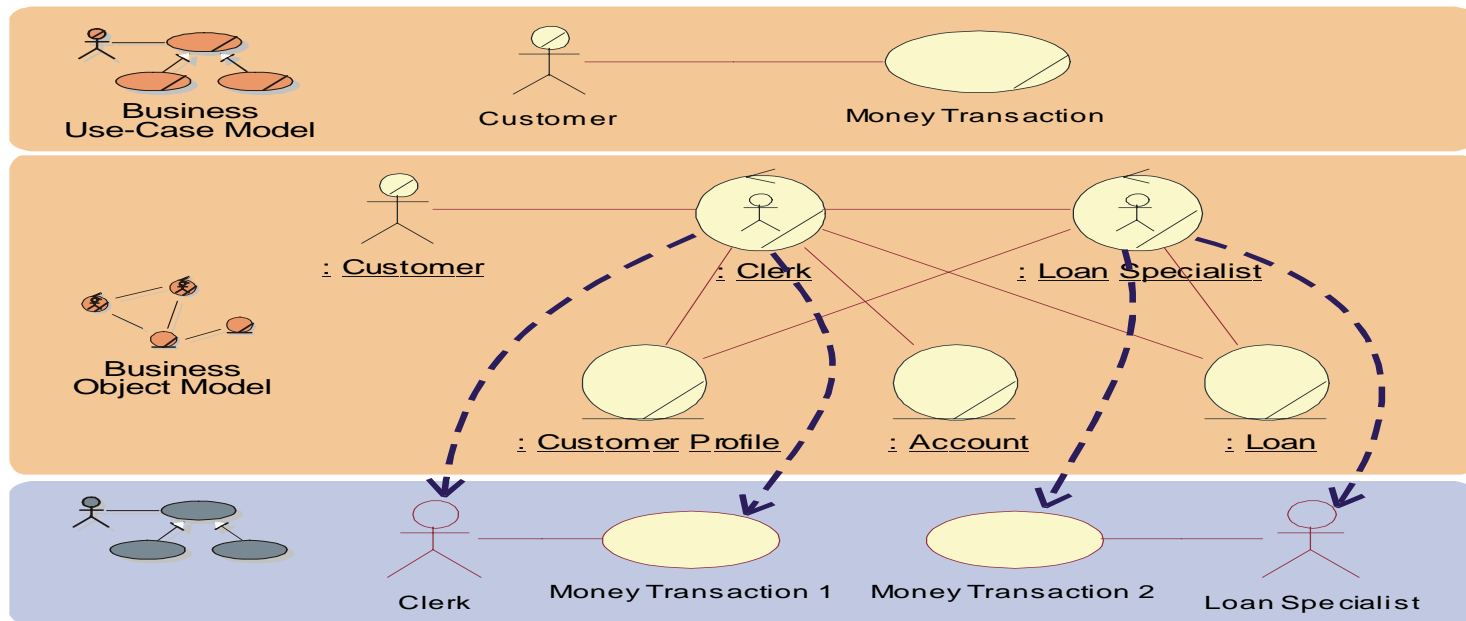
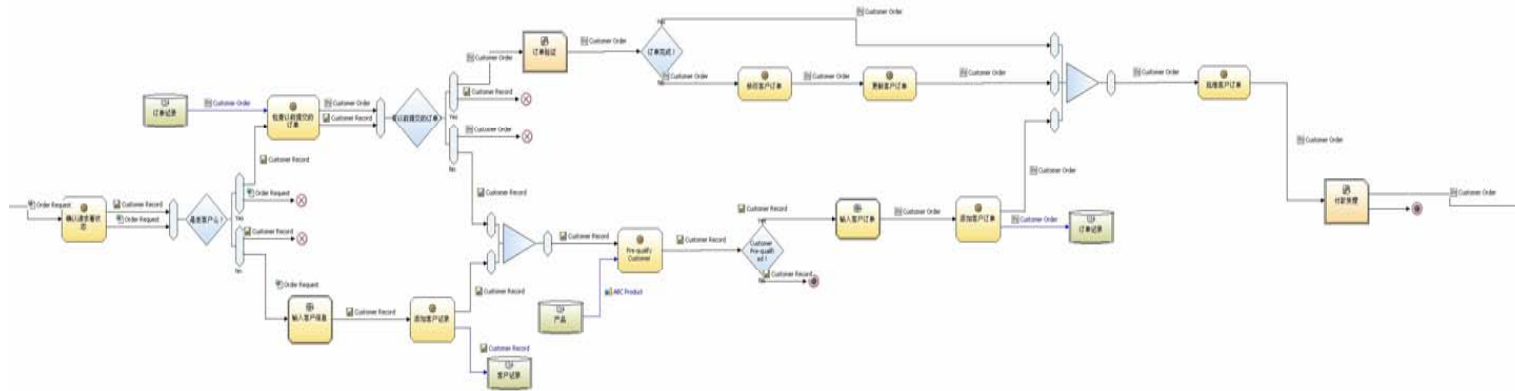


# Rational 需求管理的基础理论

需求结构:



# 业务建模，以准确理解客户所要解决的问题



# 高屋建瓴与细致入微相得益彰

文本 in Word

Microsoft Word - UserVerification.UCS

1. **UCSEC45 User Verification**

1.1. Brief Description

[UCSEC45.1 Sales clerks are required to log into the system prior to processing any sales. Each user has a unique id that must be verified prior to putting the system in the mode to process sales.]

2. **Flow of Events**

2.1. Basic Flow

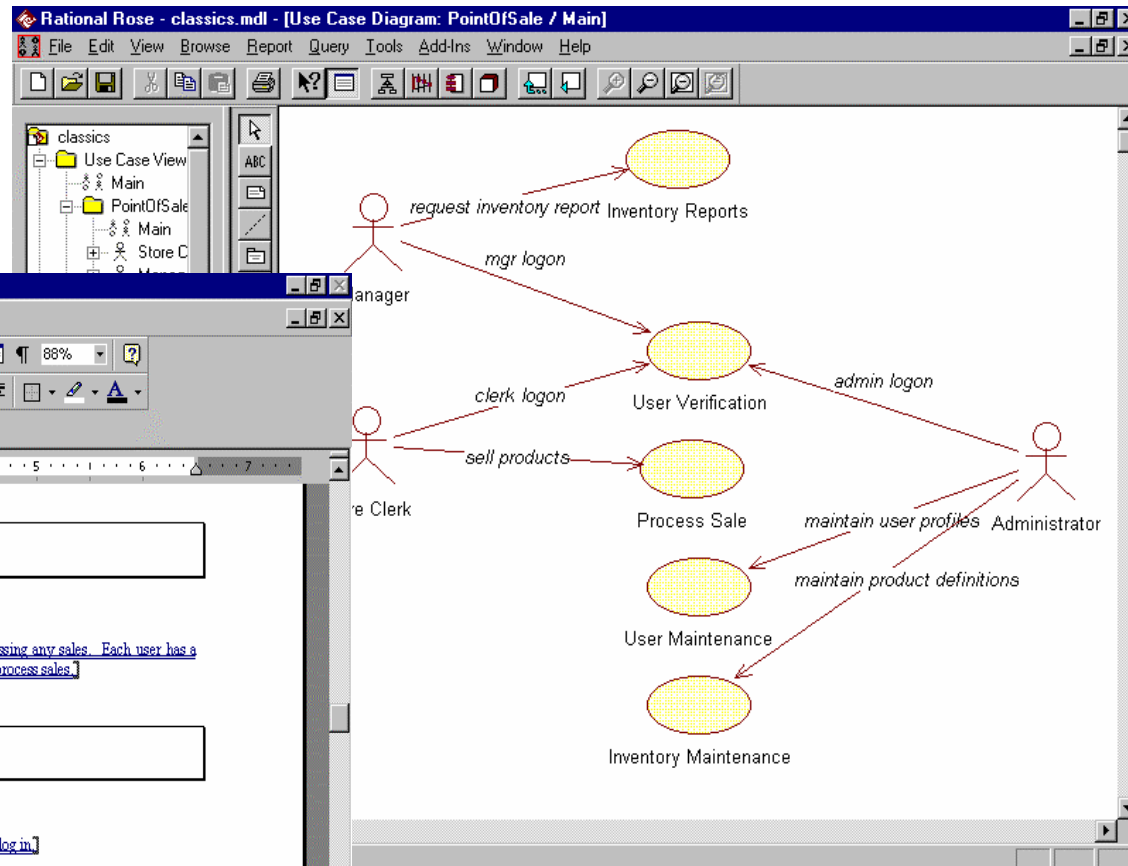
- [UCSEC45.2 The system provides the user with an option to log in.]
- The sales clerk selects the option to log in.
- [UCSEC45.3 The system displays the login screen.]
- The sales clerk enters their name and password.
- [UCSEC45.4 The system verifies that the sales clerk id is valid.]
- [UCSEC45.5 The system informs the sales clerk that the login was successful.]

2.2. Alternative Flows

2.2.1. Invalid Id Entered

- [UCSEC45.6 If the sales clerk enters an invalid ID the system will return an error message]

Page 1 Sec 3 4/7 At 2.5" Ln 5 Col 1 REC TRK EXT OVR WPH U



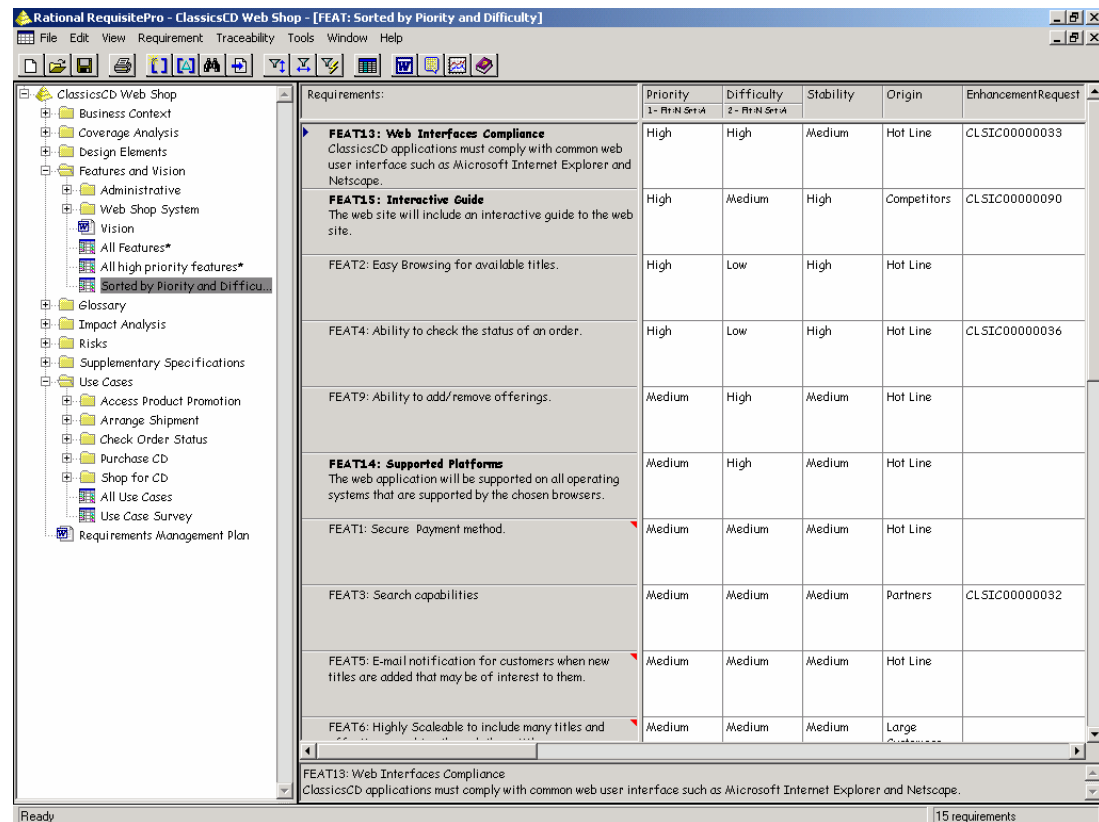
图形 in Rose





## 管理需求的属性

- ✓ 用户自定义的需求类型
- ✓ 用户自定义的需求类型
- ✓ 用户自定义的视图（过滤器）
- ✓ 预定义的各种视图



The screenshot shows the Rational RequisitePro interface for 'ClassicsCD Web Shop'. The left pane displays a project tree with folders like 'Business Context', 'Design Elements', 'Features and Vision', 'Glossary', 'Impact Analysis', 'Risks', 'Supplementary Specifications', and 'Use Cases'. The main pane shows a table of requirements sorted by priority and difficulty.

Requirements:	Priority	Difficulty	Stability	Origin	EnhancementRequest
	1 - Pt-N Set-A	2 - Pt-N Set-A			
<b>FEAT13: Web Interfaces Compliance</b> ClassicsCD applications must comply with common web user interface such as Microsoft Internet Explorer and Netscape.	High	High	Medium	Hot Line	CLSIC00000033
<b>FEAT15: Interactive Guide</b> The web site will include an interactive guide to the web site.	High	Medium	High	Competitors	CLSIC00000090
FEAT2: Easy Browsing for available titles.	High	Low	High	Hot Line	
FEAT4: Ability to check the status of an order.	High	Low	High	Hot Line	CLSIC00000036
FEAT9: Ability to add/remove offerings.	Medium	High	Medium	Hot Line	
<b>FEAT14: Supported Platforms</b> The web application will be supported on all operating systems that are supported by the chosen browsers.	Medium	High	Medium	Hot Line	
FEAT1: Secure Payment method.	Medium	Medium	Medium	Hot Line	
FEAT3: Search capabilities	Medium	Medium	Medium	Partners	CLSIC00000032
FEAT5: E-mail notification for customers when new titles are added that may be of interest to them.	Medium	Medium	Medium	Hot Line	
FEAT6: Highly Scalable to include many titles and	Medium	Medium	Medium	Large Customers	

FEAT13: Web Interfaces Compliance  
ClassicsCD applications must comply with common web user interface such as Microsoft Internet Explorer and Netscape.

Ready | 15 requirements

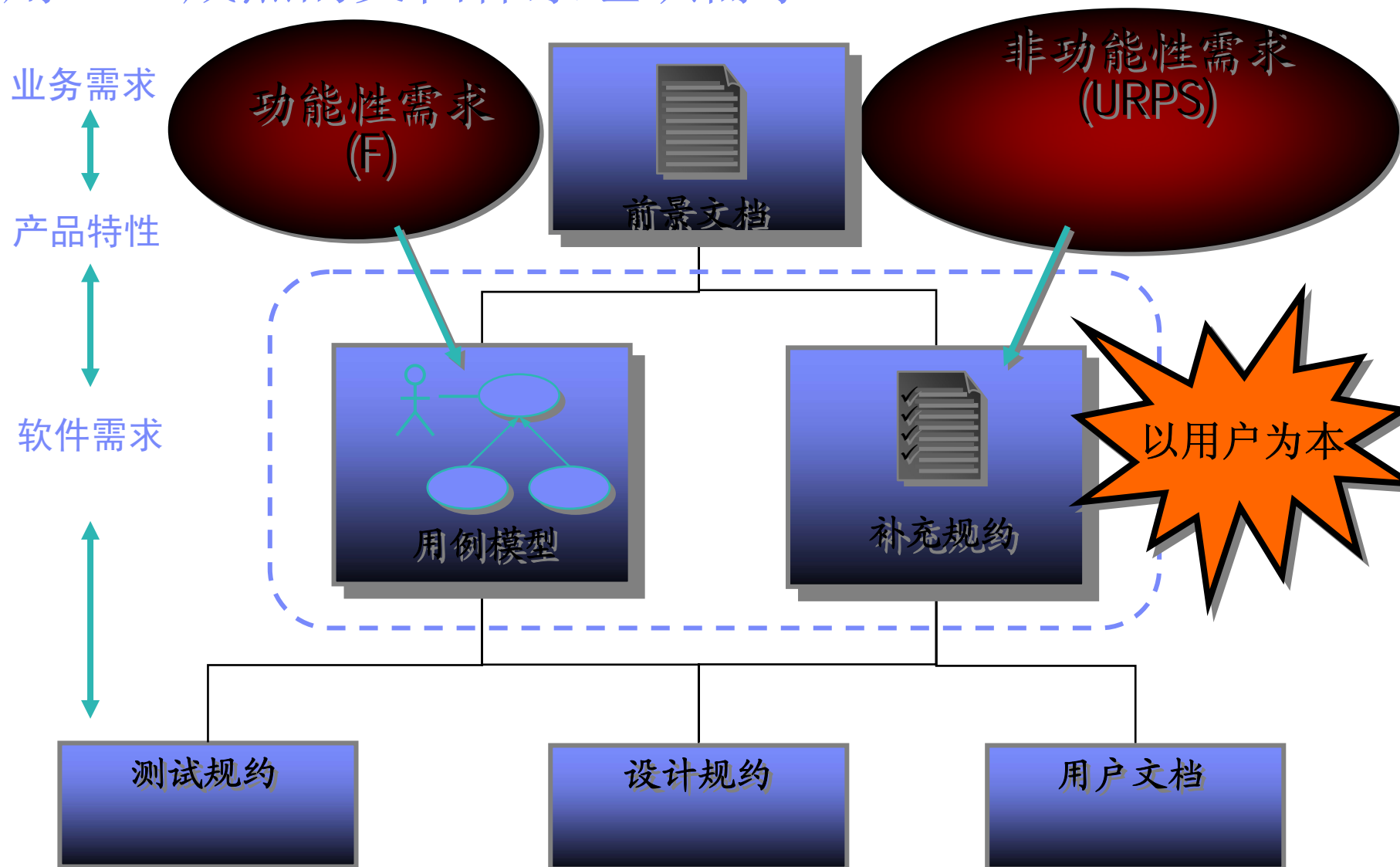
## 利用属性来更全面地描述需求

需求的属性为规划基线提供了丰富的信息，它将需求和项目的其他要素有机地联结在一起...

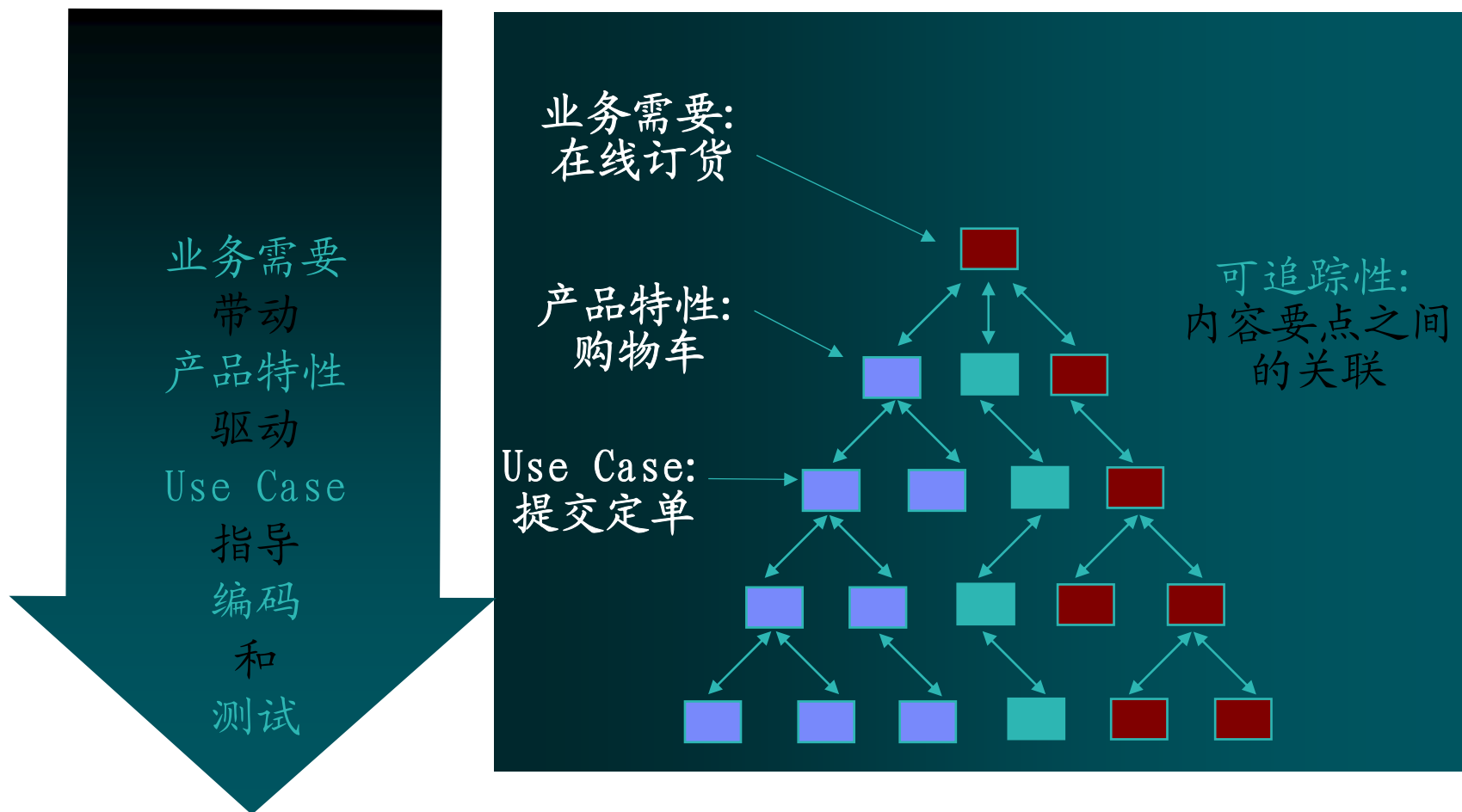
	状态	风险	优先级	工作量	成本	稳定性	来源
Req. 10	批准的	低	高 ✓		\$\$\$	高	客户 ✓
Req. 13	被提议	中等	低	✓	\$\$	低	用户
Req. 40	强制的 ✓	高 ✓	高		\$ ✓	高	用户



# 启用RUP成熟的文档体系组织需求

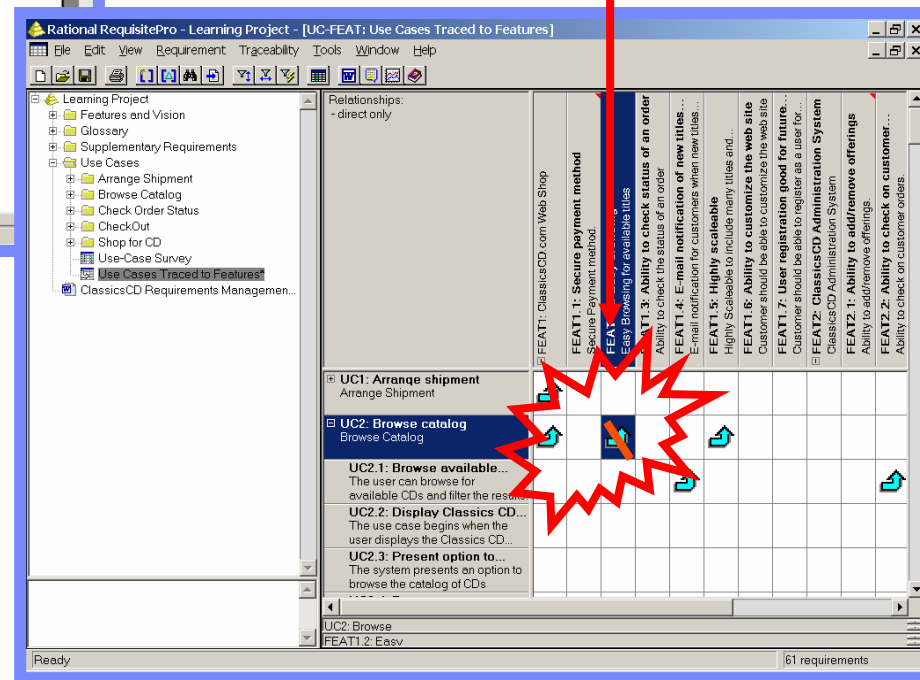
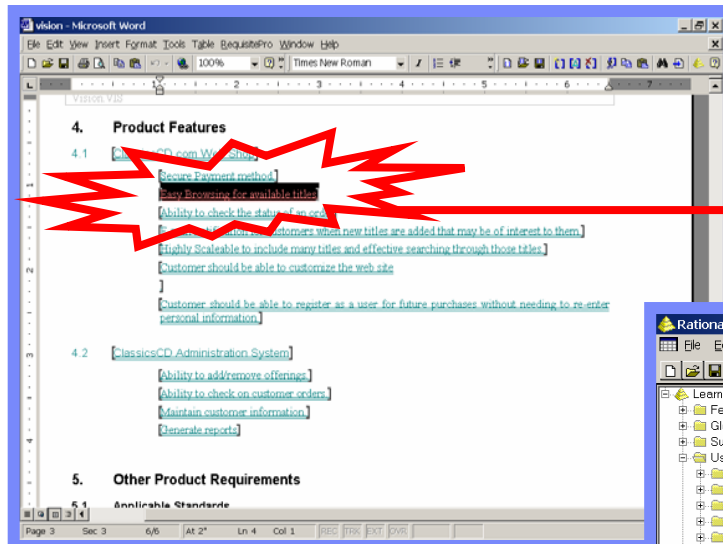


# 需求变化和变化的传播



# 管理需求之间的追踪性

保证需求的覆盖，所有关键的需求都被正确地实现了



评估需求变化对于项目所造成的影响



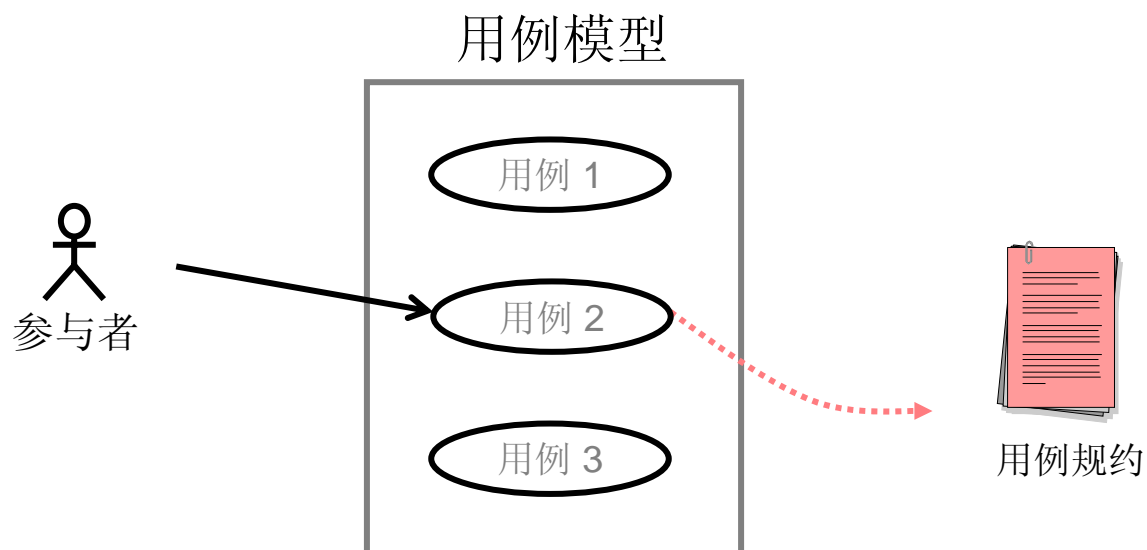
# 内容

- 需求开发体系
- 什么是用例建模
- 用例的开发过程
- 管理用例模型的复杂度

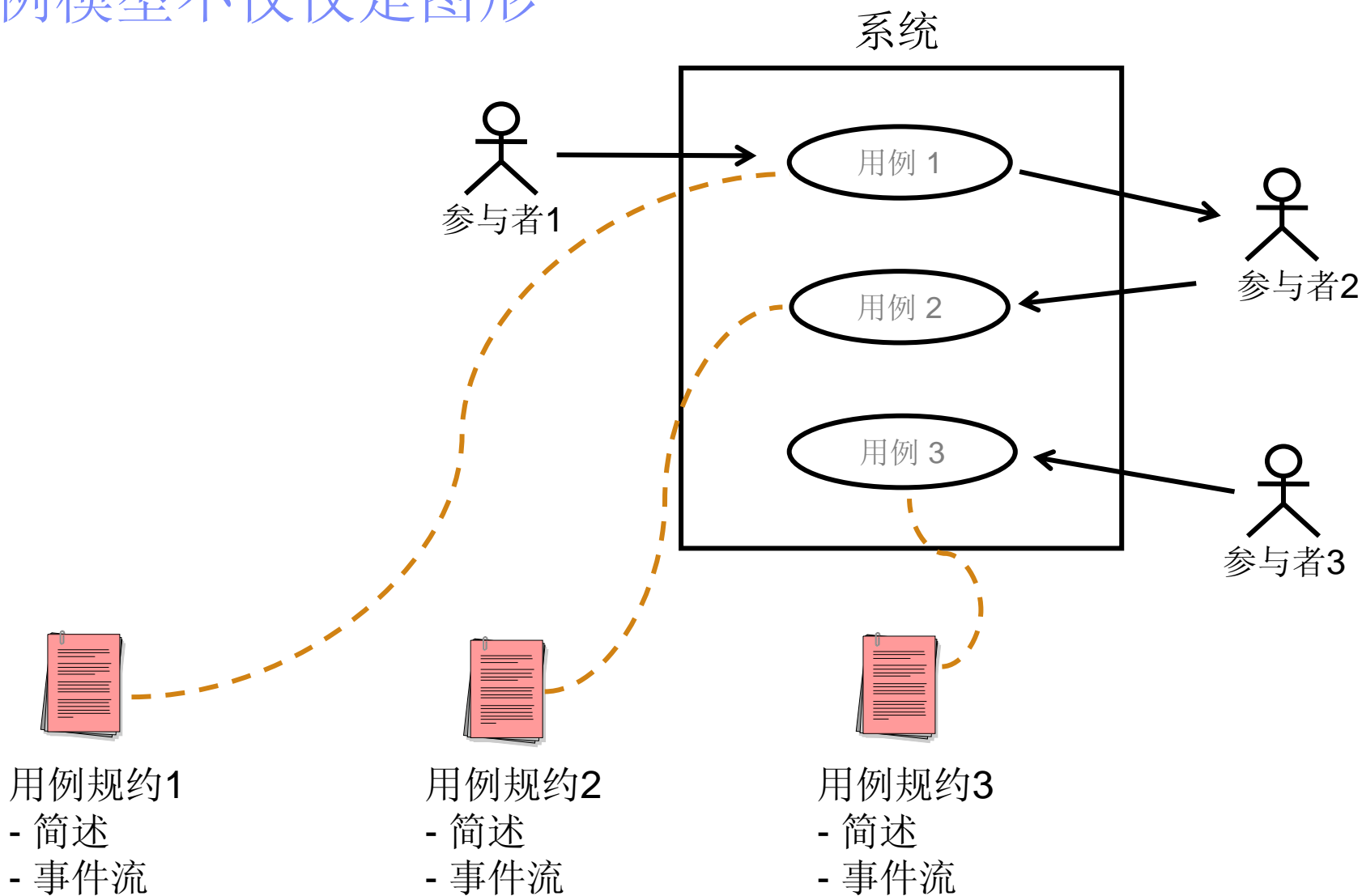


## 什么是用例建模?

- 一种描述功能性需求的方法
- 清晰地定义系统的边界
- 详细描述系统和外部环境的交互

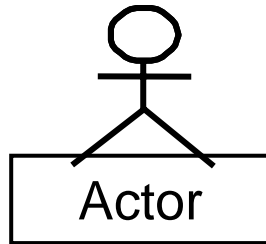


# 用例模型不仅仅是图形





## 用例建模中的主要模型元素



### 参与者 (Actor)

在系统外与系统发生交互的某一个人或系统



### 用例 (Use case)

系统向参与者所提供的某种服务



## 什么是用例？

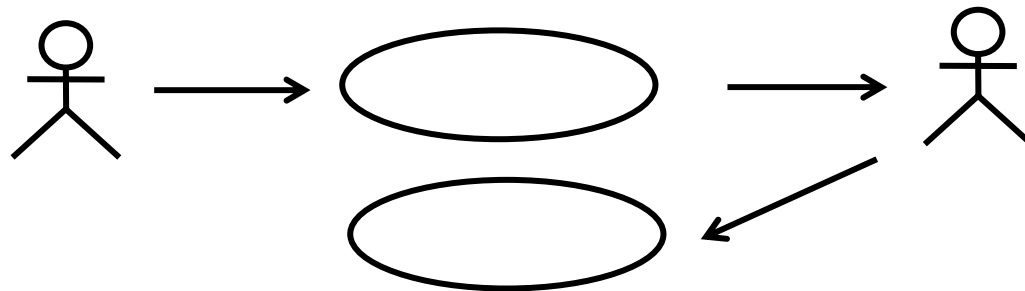
用例名称

用例是由系统所执行的一系统列动作，  
为相关的参与者提供有价值的服务

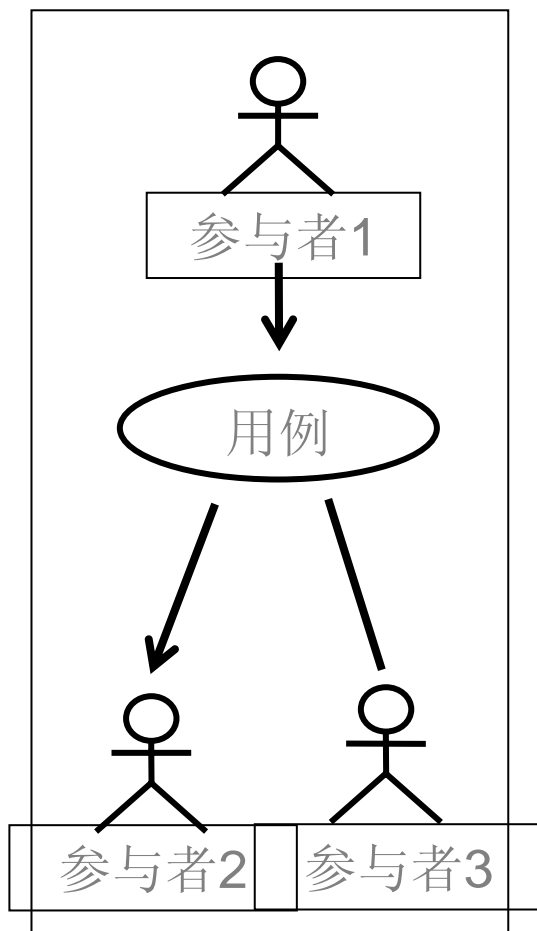


## 用例描述了系统需求

- 每一个用例
  - ▶ 表述了系统向参与者所提供的某种服务
  - ▶ 也表述了参与者是如何使用系统的
  - ▶ 描述了系统和参与者之间的一段对话

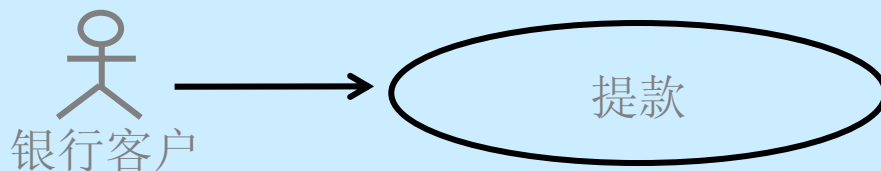


## 通讯关联



- 参与者和用例之间进行对话的一个渠道
- 用一条带或不带箭头的线来表示
  - ▶ 箭头表示是谁发起了这次对话
  - ▶ 没有箭头表示任何一方都可以发起对话
  - ▶ 箭头并不表示数据的流向，数据流向总是双向的

## 场景是用例的一个实例



### 场景 1

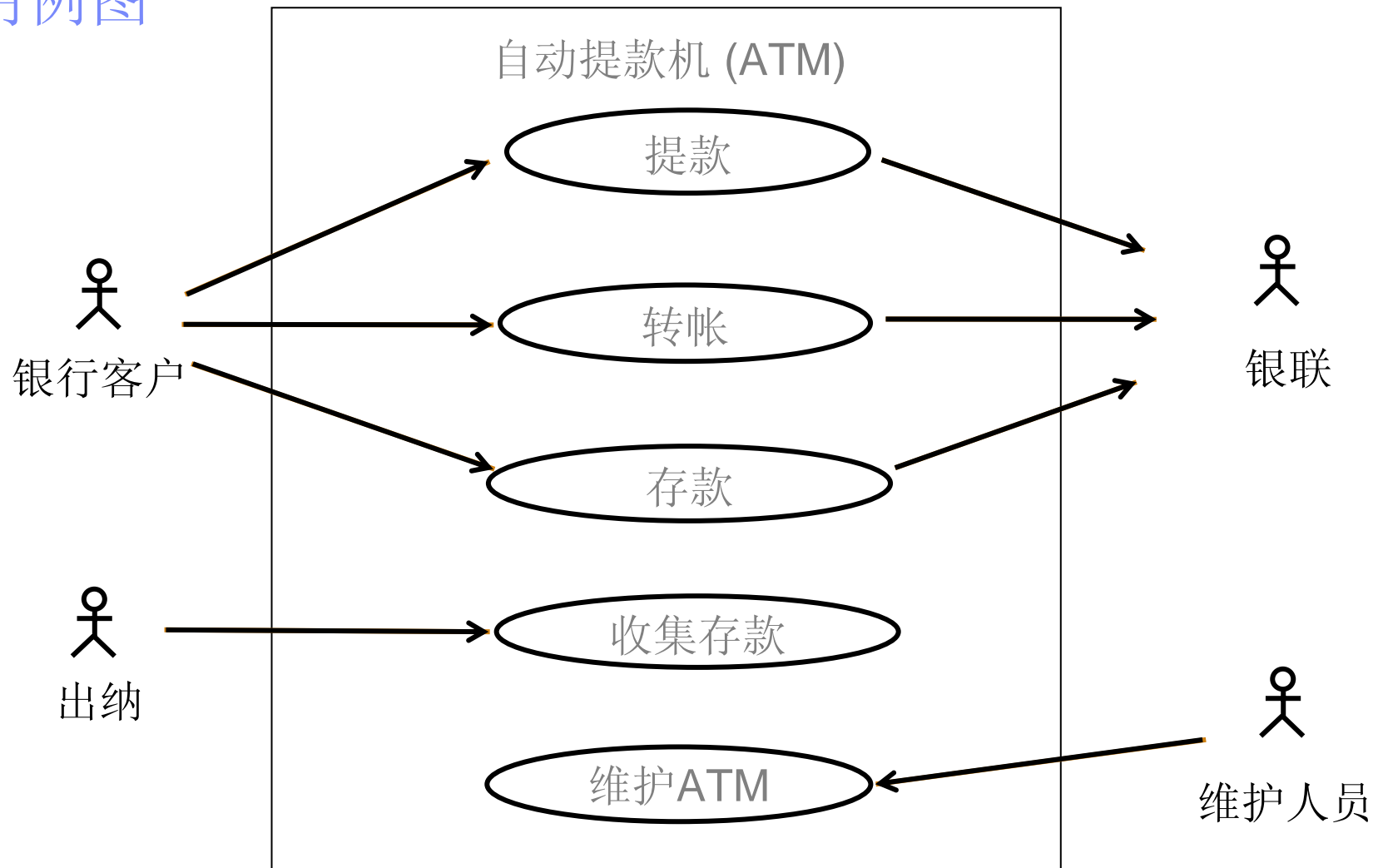
输入提款金额  
提取现金  
推出系统，取回卡片

### 场景 2

输入提款金额，金额大于  
2500元  
系统提示限额信息并提示客  
户重新输入  
输入正确金额  
提取现金  
推出系统，取回卡片



# 用例图



## 用例方法的好处

- 为需求提供了应用环境
  - ▶ 把系统系统定义在一个使用环境中
  - ▶ 帮助解释为什么需要该系统
  - ▶ 有助于保证没有需求的遗漏
- 易于理解
  - ▶ 站在系统使用者的立场上来定义需求
  - ▶ 使用客户和用户都能理解的术语和表述
  - ▶ 易于客户对于需求的审核
- 可以为其他开发环节所重用
  - ▶ 项目计划、设计、测试、用户手册



# 内容

- 需求开发体系
- 什么是用例建模
- 用例建模的步骤
- 管理用例模型的复杂度



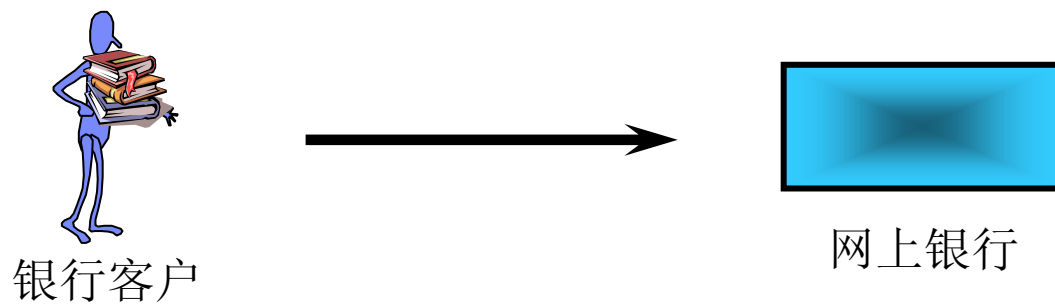
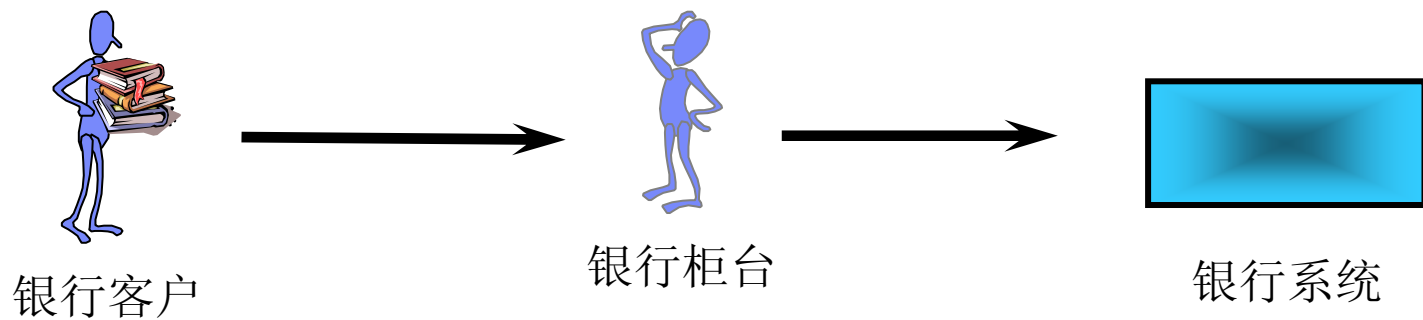


## 内容

- 什么是用例建模
- 用例建模的步骤
  - ▶ 找出参与者
  - ▶ 找出用例
  - ▶ 描述用例框架
  - ▶ 细化用例描述
- 管理用例模型的复杂度



# 找出参与者



## 找出参与者

- 系统有哪些用户？
- 有谁想从系统中获取信息？
- 谁负责向系统提供信息？
- 系统将会在哪些部门使用？
- 是谁负责管理理和维护系统？
- 有没有其他的系统会和该系统发生交互？



## 对于参与者的描述

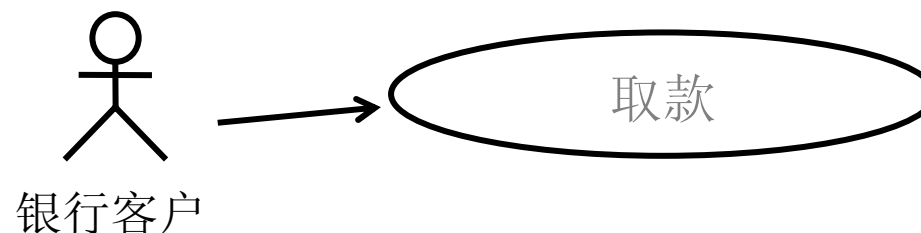
名字

银行客户

简要描述

使用该系统存取款及转帐的人

和用例之间的关系



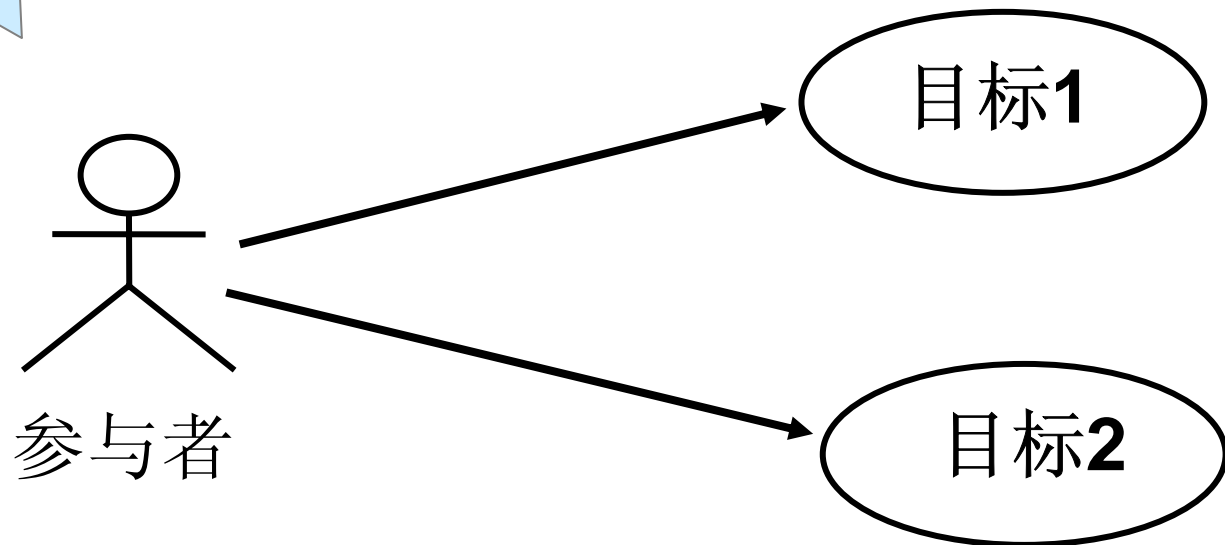
# 内容

- 什么是用例建模
- 用例建模的步骤
  - ▶ 找出参与者
  - ▶ 找出用例
  - ▶ 描述用例框架
  - ▶ 细化用例描述
- 管理用例模型的复杂度



## 找出用例

使用该系统我想达到一个什么目标？



## 找出用例

- 每一个参与者想达到的目标是什么？
  - ▶ 为什么该参与者需要使用该系统？
  - ▶ 该参与者会在系统中创建、修改、删除或访问任何数据吗？
  - ▶ 当外部有事件发生的时候，该参与者需要通知系统吗？
  - ▶ 当系统内部有某些事件发生时，需要通知该参与者吗？
- 已定义的系统功能是否足以满足所有的业务需求？



# 内容

- 什么是用例建模
- 用例建模的步骤
  - ▶ 找出参与者
  - ▶ 找出用例
  - ▶ 描述用例框架
  - ▶ 细化用例描述
- 管理用例模型的复杂度





## 描述每一个用例的框架

为每一个步骤  
标上数字

用例名称

简述

基本流

1. 步骤一
2. 步骤二
3. 步骤三

**A1** 备选流一

**A2** 备选流二

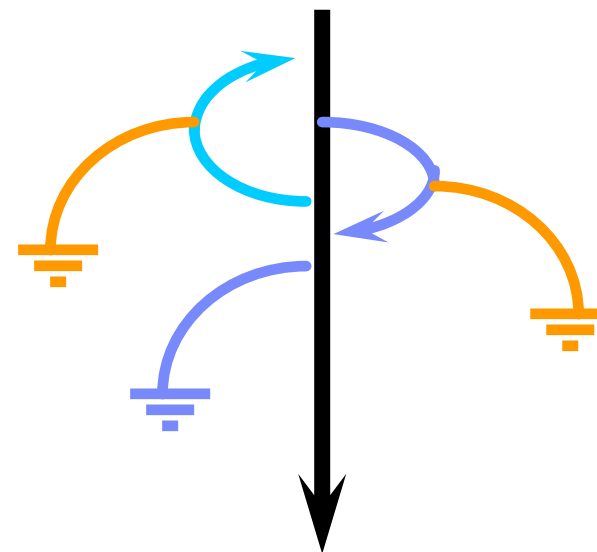
**A3** 备选流三

列出主要的  
备选流



## 事件流

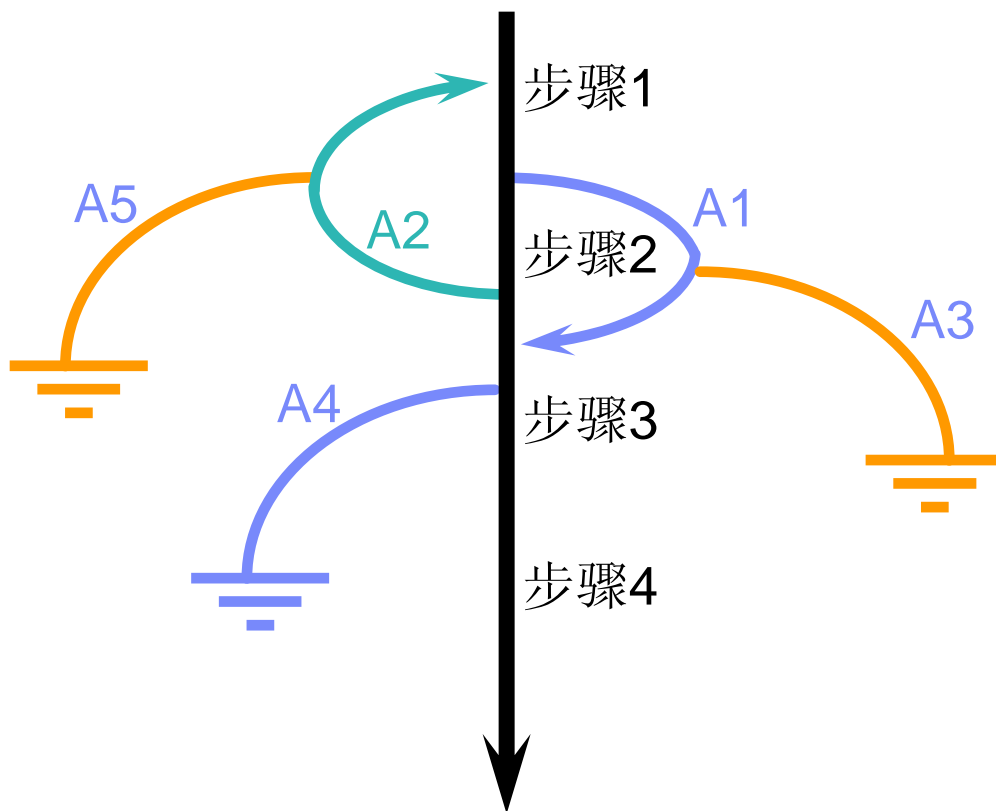
- 一个基本流
  - ▶ 最顺利的场景
  - ▶ 从开始到结束一切都顺利的场景
- 很多个备选流
  - ▶ 除基本流之外的另外一些正常场景
  - ▶ 偶尔发生的场景
  - ▶ 异常或错误处理



流：是指一系列的步骤



## 表示基本流和备选流



### 用例名称

1. 简要说明

2. 事件流

2.1 基本流(Basic Flow)

步骤 1

步骤 2

步骤 3

步骤 4

2.2 备选流(Alternative Flows)

2.2.1 A1 ...

2.2.2 A2 ...

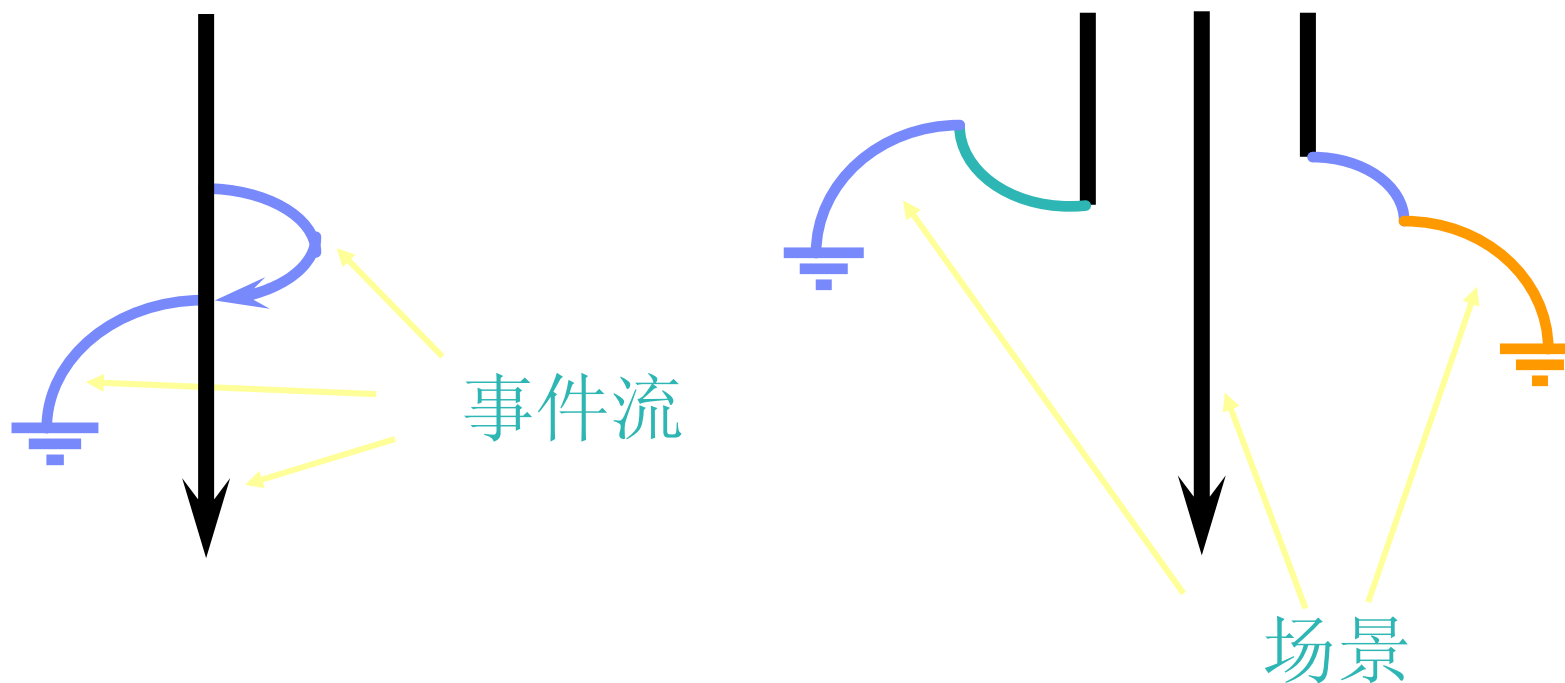
2.2.3 A3 ...

2.2.4 A4 ...

2.2.5 A5 ...



## 什么是场景？



事件流：一系列的步骤

用例：描述所有的流

场景：用例的某一个实例，从用例开始直到它的某一个结束点



## 描述用例框架

- 基本流
  - ▶ 是什么事件启动了用例?
  - ▶ 用例是如何结束的?
  - ▶ 用例中最常用到的行为是什么?
- 备选流
  - ▶ 在用例中有无可选执行的情况?
  - ▶ 那些行为是偶尔才发生的?
  - ▶ 是否有没有包含在基本流中的正常情况?
  - ▶ 发生错误的情况下该如何处理?



# 内容

- 什么是用例建模
- 用例建模的步骤
  - ▶ 找出参与者
  - ▶ 找出用例
  - ▶ 描述用例框架
  - ▶ 细化用例描述
- 管理用例模型的复杂度



## 为什么细化用例描述？

- 描述软件需求
  - ▶ 为后续的开发准备一个系统功能描述
- 在事件流中描述更为详细的信息
  - ▶ 参与者做了什么动作
  - ▶ 系统是如何响应的
  - ▶ 系统和参与者之间交换了哪些信息
- 描述用例场景
  - ▶ 成功场景
  - ▶ 失败场景
- 描述额外的用例信息
  - ▶ 前置条件
  - ▶ 后置条件



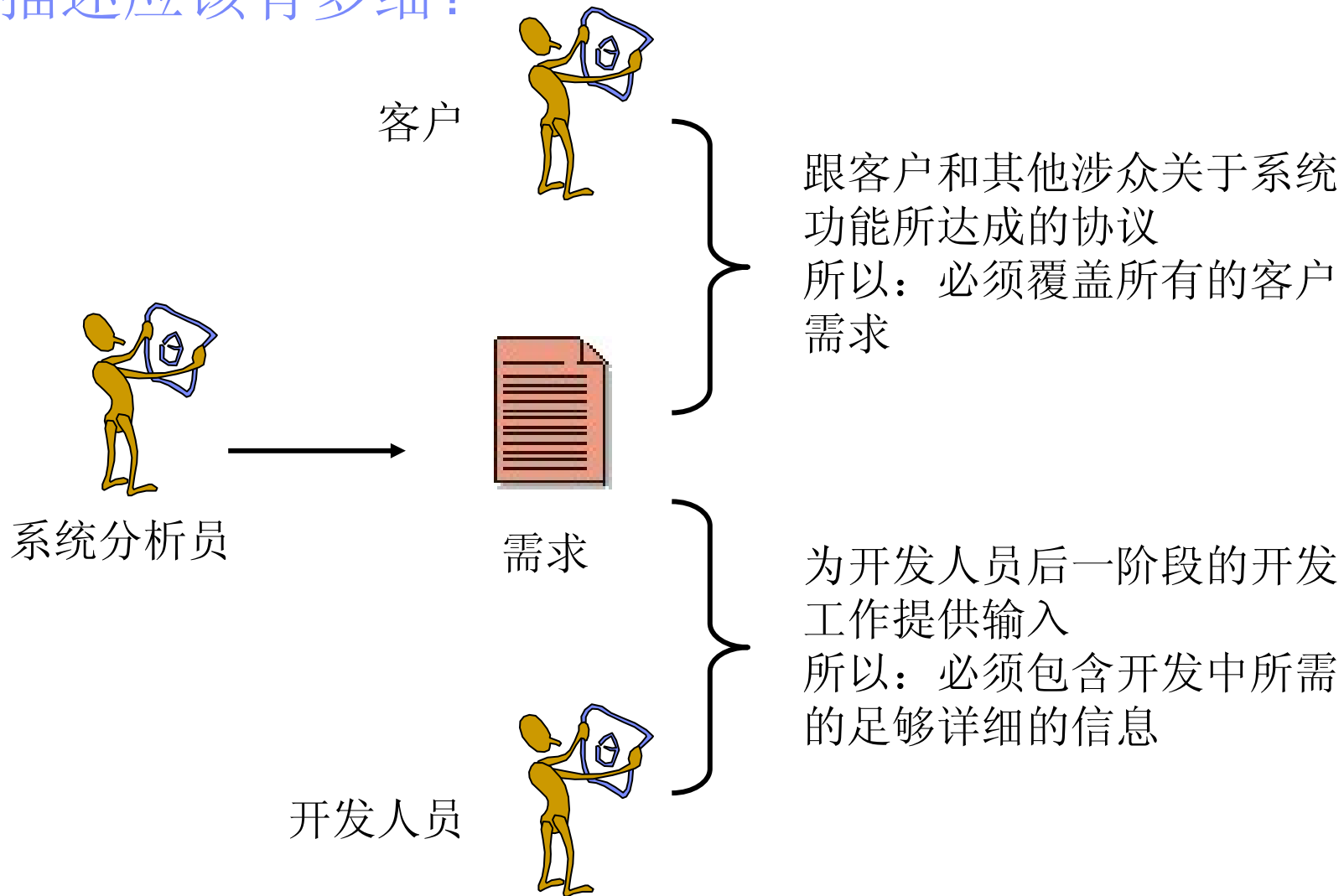
# 内容

- 需求开发体系
- 什么是用例建模
- 用例建模的步骤
- 管理用例模型的复杂度

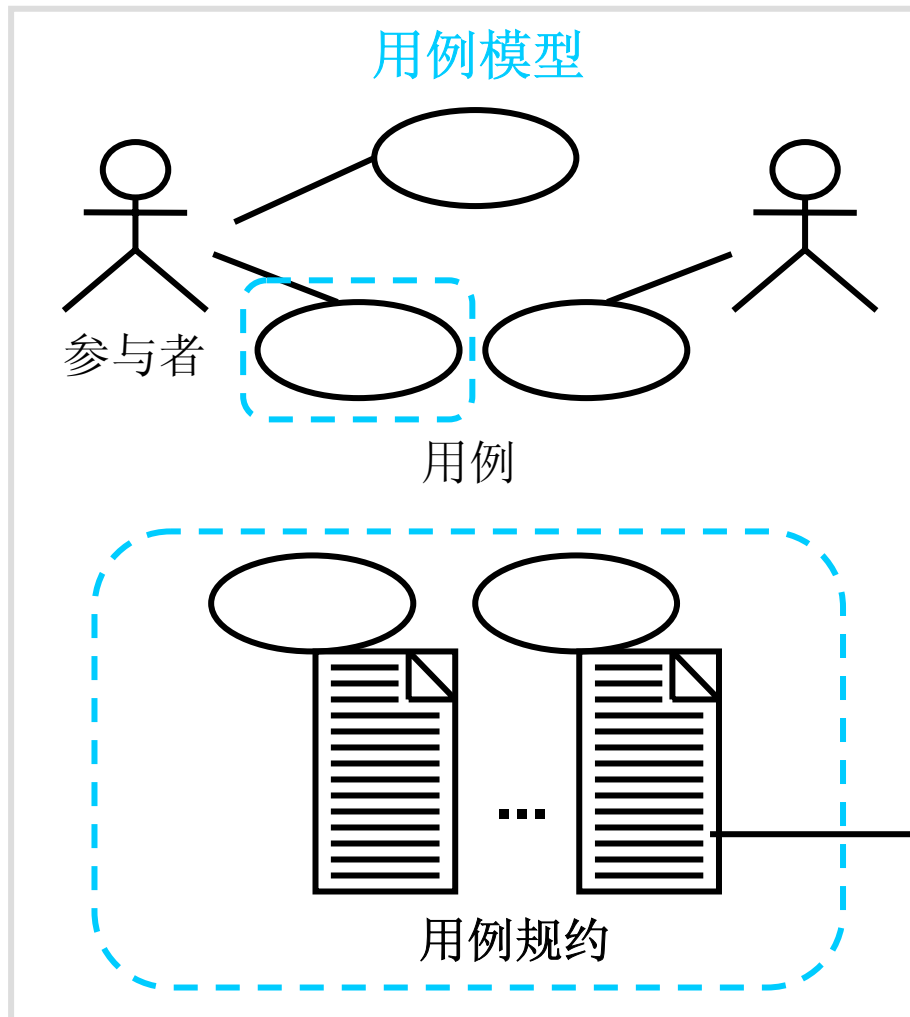




# 用例描述应该有多细?



# 总结：用例模型



## 用例规约

- 名称
- 简要说明
- 事件流
  - ▶ 基本流
  - ▶ 备选流
- 场景
  - ▶ 成功场景
  - ▶ 失败场景
- 特殊需求
- 前置条件
- 后置条件



Thank  
you

