

\* CHANGEDESIGNSTUDIO V1.0

# 第6章：用例图

\* COPYRIGHT(C) 2001 CHANGEDESIGN ALL RIGHT RESERVED  
\* REQUIRES IE4.0+ -- 800\*600+ -- MICROMEDIA FLASH/5 PLUGIN  
\* WEBSITE: [WWW.CHANGEDESIGN.COM](http://WWW.CHANGEDESIGN.COM) \* SITE IMAGES FOR SOPHOTO AND TONYSTONE

## 本章内容

---

- 什么是面向对象的分析
- 用例和用例驱动开发
- 如何阅读用例图
- 如何绘制用例图
- 用例图应用说明



\* CHANGEDESIGNSTUDIO V2.0

\* COPYRIGHT(C) 2001 CHANGEDESIGN ALL RIGHT RESERVED  
\* REQUIRES IE4.0+ -- 800\*600+ -- MICROMEDIA FLASH/5 PLUGIN  
\* VISIT OUR WEBSITE JARGES FOR SOPHOTO AND TONYSTONE



# 本章内容

---

- 什么是面向对象的分析
- 用例和用例驱动开发
- 如何阅读用例图
- 如何绘制用例图
- 用例图应用说明



\* COPYRIGHT(C) 2001 CHANGEDESIGN ALL RIGHT RESERVED  
\* REQUIRES IE4.0+ -- 800\*600+ -- MICROMEDIA FLASH/5 PLUGIN  
\* WEBSITE CHARGES FOR SOPHOTO AND TONYSTONE



# 什么是面向对象的分析？

---

面向对象的分析（**Object Oriented Analysis, OOA**），是软件生命周期的一个阶段，具有一般分析方法共同具有的内容、目标及策略，但强调运用面向对象方法进行分析，用面向对象的概念和表示法表达分析结果。

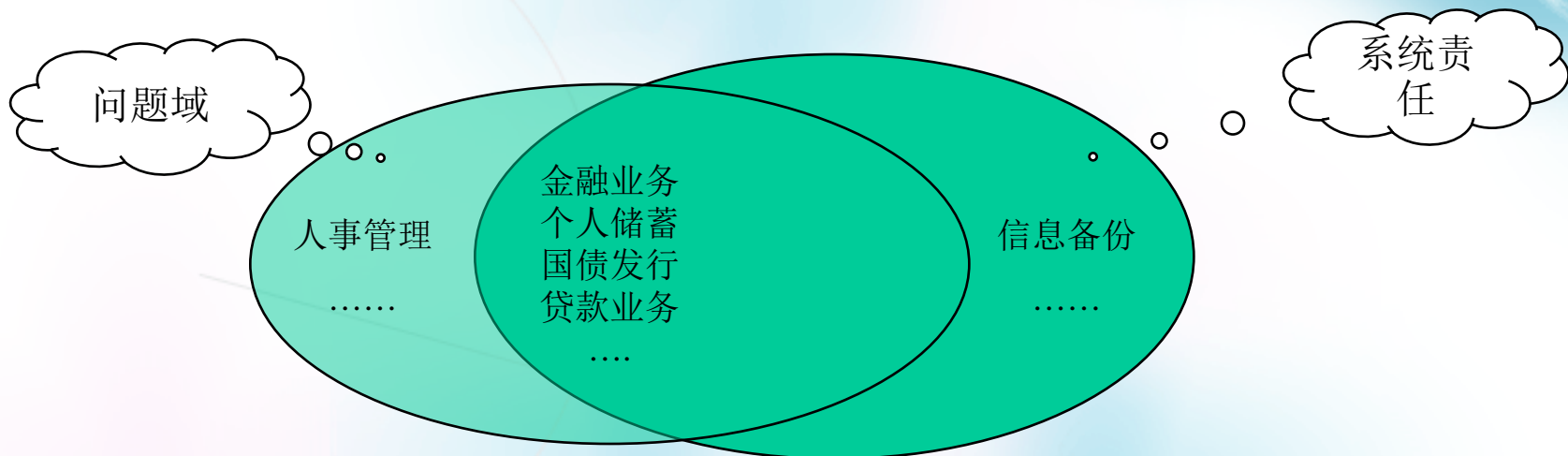
**基本任务：**运用面向对象方法，对问题域和系统责任进行分析 and 理解，找出描述问题域及系统责任所需的**对象**，定义对象的**属性**、**操作**以及它们之间的**关系**。

**目标：**建立一个符合问题域、满足**用户需求**的**OOA**模型。

# 什么是面向对象的分析？

**问题域（problem domain）**：被开发系统的应用领域，即在现实世界中由这个系统进行处理的业务范围。

**系统责任（system responsibilities）**：所开发的系统应该具备的职能。



# 需求分析面临的主要问题

---

## 1、问题域和系统责任复杂性日益增长

### 困难所在：

- 软件开发人员与领域专家的技术背景不同（学习、有更高的见解、适合于软件）
- 现今的系统所面临的问题域比以往更为广阔和复杂，系统比以往更为庞大。

## 2、交流问题

如果分析所产生的文档使分析员以外的其他人员都很难读懂，那就很不利于交流。这会使彼此的思想不易沟通，并容易隐藏许多错误。

——例如：分析人员与客户及领域专家，分析人员之间，分析人员与设计人员，开发人员与管理人员



# 需求分析面临的主要问题

---

## 3、需求的不断变化

**引起需求变化的因素：**用户（主客观因素）、竞争因素、经费、技术因素等

软件开发必须以合作的态度满足用户需求

## 4、软件复用的要求

复用级别提高——分析结果复用

要求分析模型中的可复用部分用于多个系统

要求一个分析模型在多种条件下设计和实现

## OOA方法概貌：概念与表示法

基本概念	建模元素	原则
对象，类	对象、类	分类
属性	属性	
操作	操作	行为分析
继承	泛化	继承
消息	消息	消息通信
关联	关联、链	关联
聚合	聚合	整体-部分关系
多态	多态性	
主动对象	主动对象(类)	
包	包	粒度控制
.....		



# OOA模型

给出所有与问题域和系统责任有关的对象，用对象类表示

捕获与描述用户的要求

一幅交互图表现完成某一项特定功能的一组对象之间的详细交互；一幅状态图描述了一个对象的状态变迁；活动图描述了一个服务的流程或业务流程。

定义每个对象类的属性与服务

需求模型

基本模型(类图)

对象层

特征层

关系层

用例图

辅助模型

交互图  
活动图  
状态图  
包图  
.....

对关系的元素打包，帮助理解系统模型。

通过已定义的关系描述对象类之间的关系

模型规约

对模型中的所有元素进行详细说明。

# OOA过程

定义用例（辅助模型，可选）

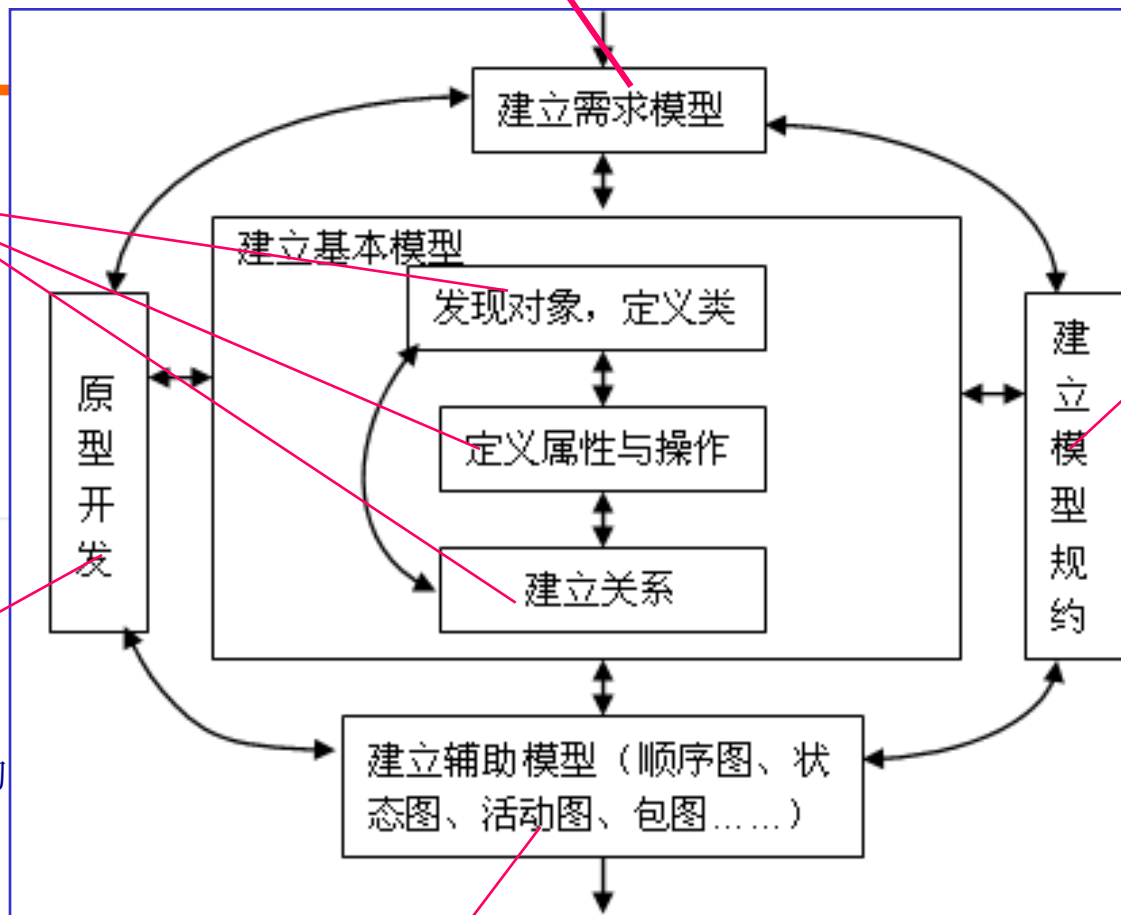
用用例对用户需求进行规范化描述。

建立详细说明

对模型中的成分进行规范的定义和文字说明。可以集中进行，也可分散在各个活动中。

建立类图（基本模型）

- \* 发现对象、定义对象类
- \* 识别对象的内部特征
- \* 识别对象的外部关系



原型开发（可选）  
结合其他活动  
反复进行

建立交互图、状态图和活动图等  
（辅助模型，可选）

# 本章内容

---

- 什么是面向对象的分析
- 用例和用例驱动开发
- 如何阅读用例图
- 如何绘制用例图
- 用例图应用说明



# 现代需求实践

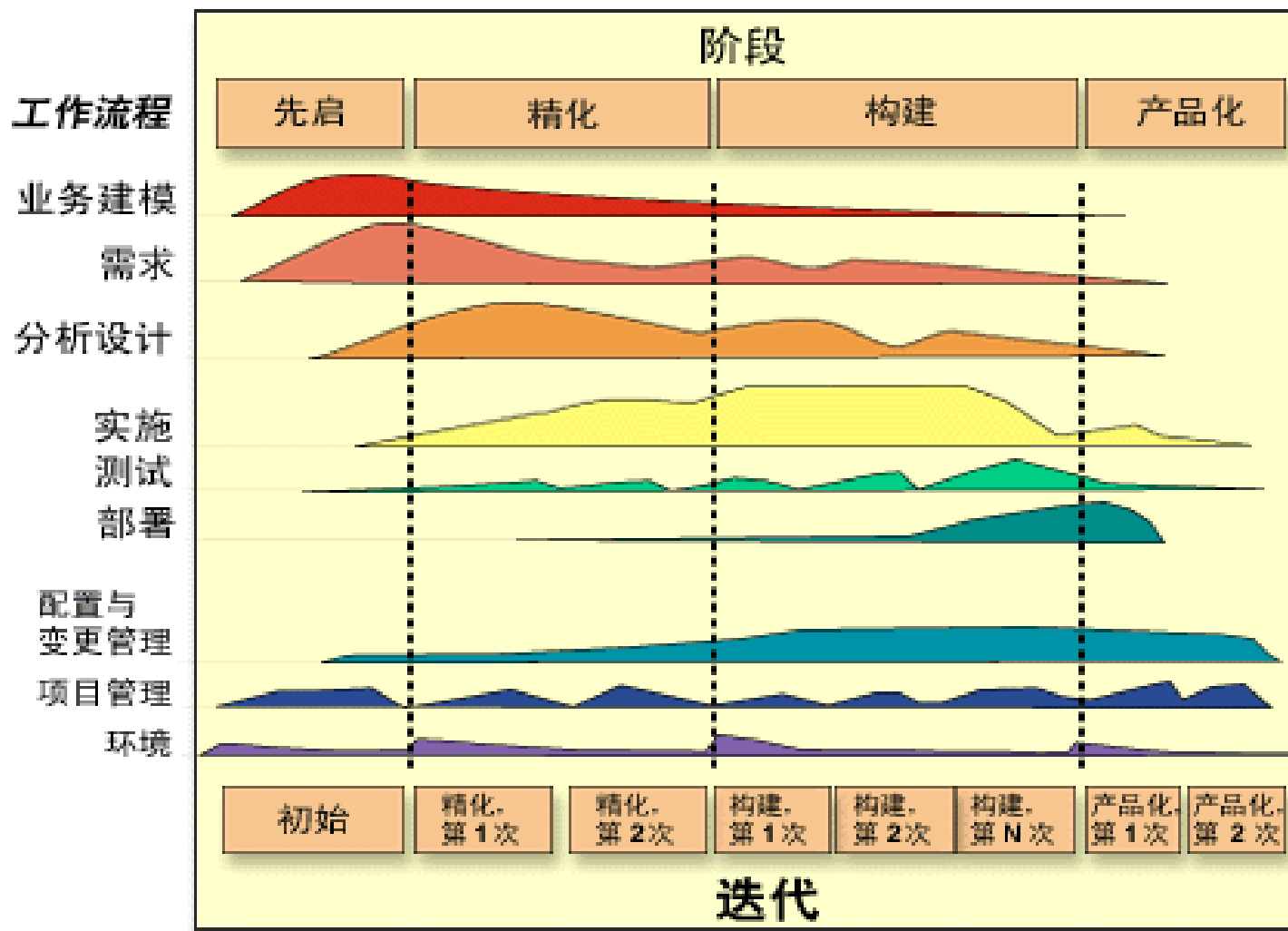
实践名称	描述
用例 (Use case)	描绘一个系统外在可见的需求情况，是代表系统中各个项目相关人员(风险承担人，Stakeholder)之间就系统的行为所达成的契约
用户故事 (user story)	由客户参与编写，说明他们需要系统为他们做什么，一般用客户的术语编写，其长度约为三句话左右 (XP:extreme programming)
特性 (Feature)	就是一个小的，具有客户价值的功能，通常表示为 <action><result><object> (FDD:Feature Driven Development)

- 原则：站在用户的角度看待系统、定义系统；使用用户能够看懂的语言来表述。

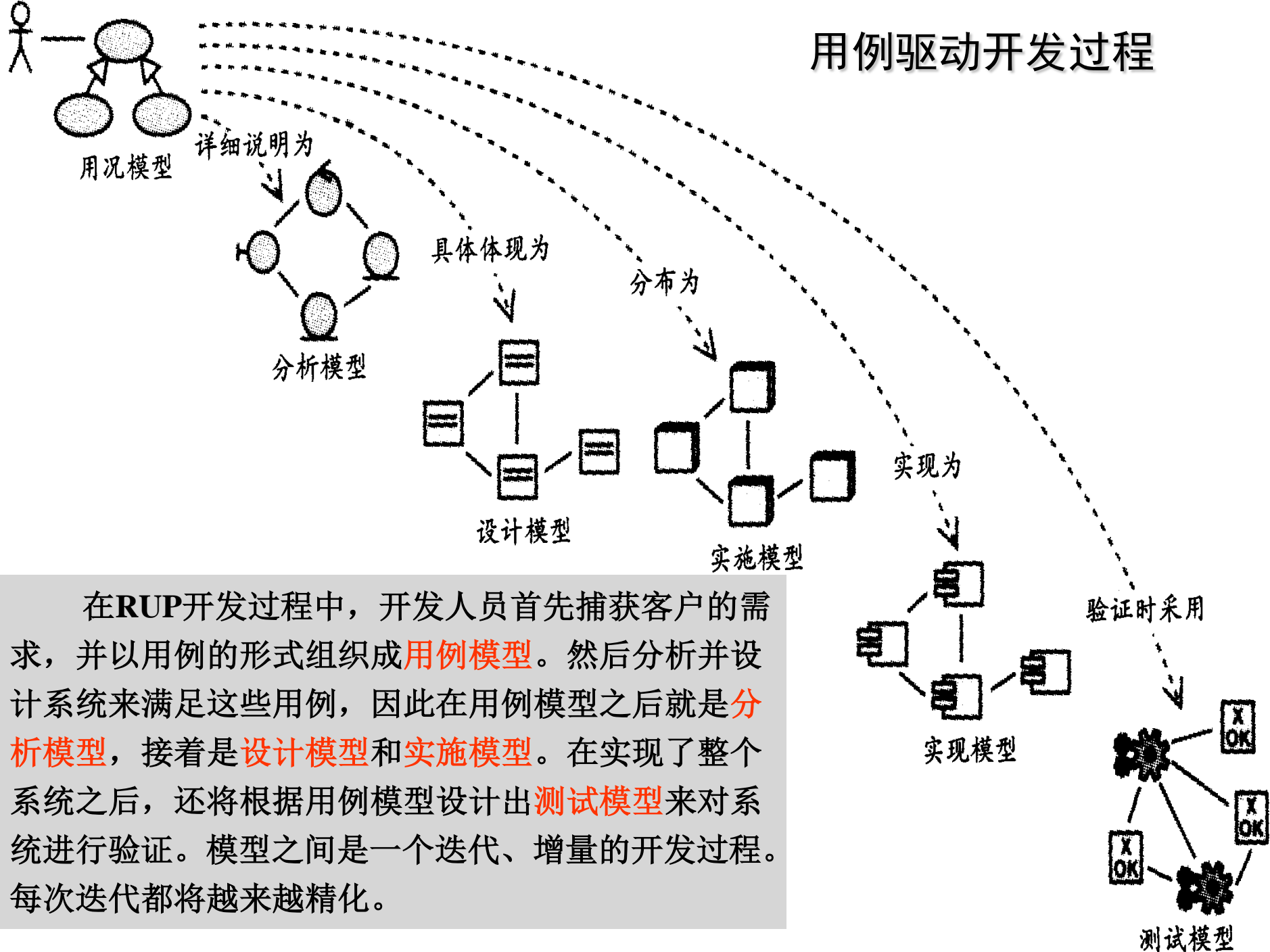


# 统一软件开发过程 RUP (Rational Unified Process)

用例驱动、以架构设计为中心、迭代增量开发



# 用例驱动开发过程



在RUP开发过程中，开发人员首先捕获客户的需求，并以用例的形式组织成**用例模型**。然后分析并设计系统来满足这些用例，因此在使用例模型之后就是**分析模型**，接着是**设计模型**和**实施模型**。在实现了整个系统之后，还将根据用例模型设计出**测试模型**来对系统进行验证。模型之间是一个迭代、增量的开发过程。每次迭代都将越来越精化。



# 业务模型

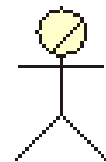
---

- 研究机构结构及其中的角色和他们之间的联系，机构的工作流等。作用是帮助我们理解机构和软件系统。
- 包括
  - **Business Use Case图**
  - **Activity 图（工作流）**
  - **分析级Class图（业务实体）**

## 业务模型—Business Use Case图

- 业务参与者（business actor）

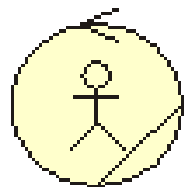
机构外与机构交互的一切



客户

- 业务工人（business worker）

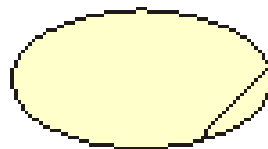
机构中的角色



销售人员

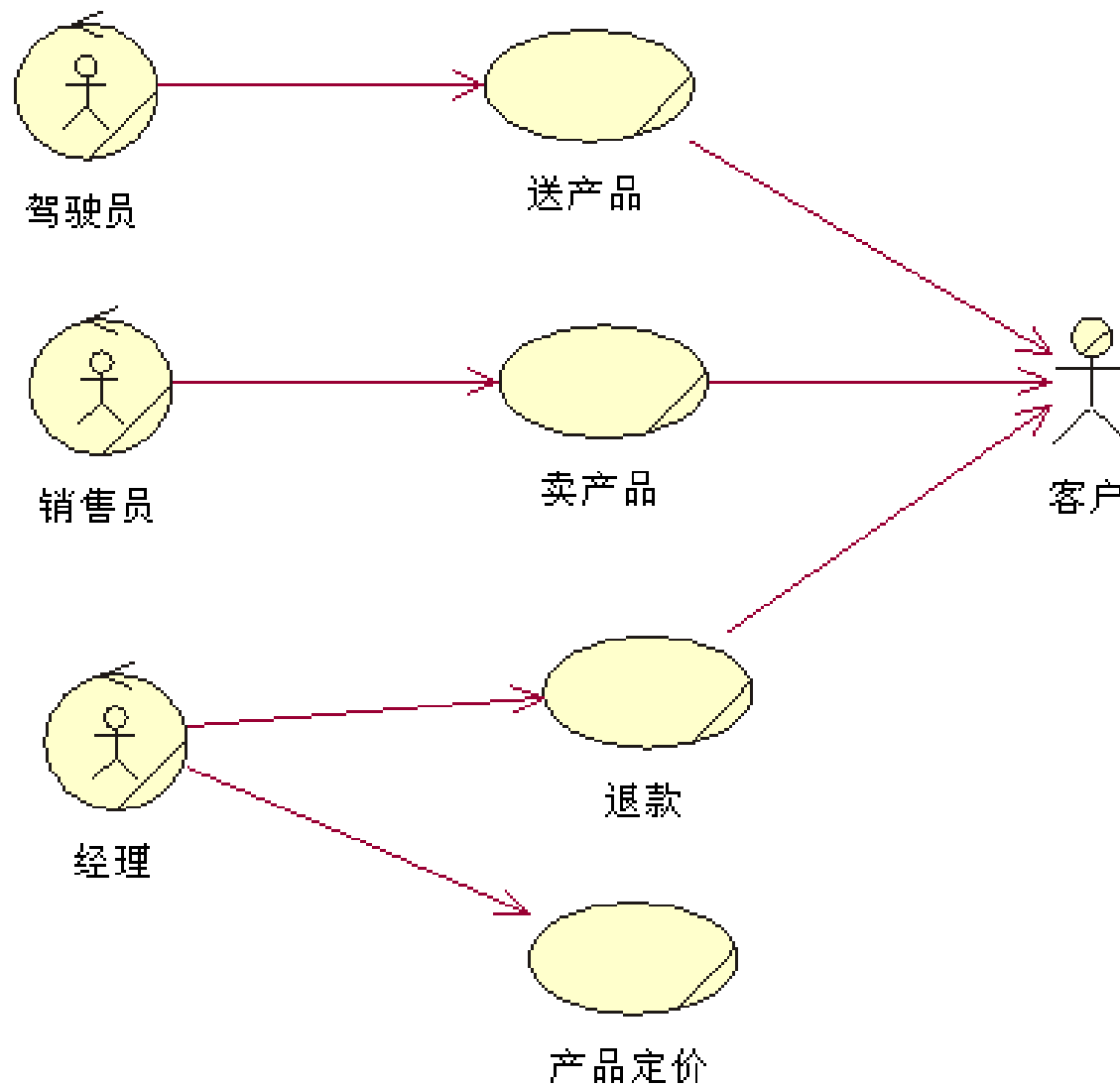
- 业务用例

机构中一组相关的工作流，  
告诉人们机构做什么



整理库存

# 业务模型—Business Use Case图举例



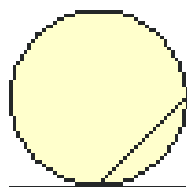
\* CHANGEDESIGNSTUDIO V1.0

CHANGEDESIGN ALL RIGHT RESERVED  
-- MICROMEDIA FLASH/5 PLUGIN  
ES FOR SOPHOTO AND TONYSTONE

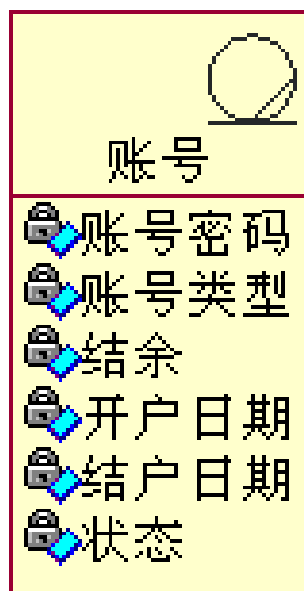
# 业务模型—业务实体

- 业务实体（business entity）

机构经营业务期间使用的对象或业务处理过程中产生的对象。包括业务工人日常处理的项目。

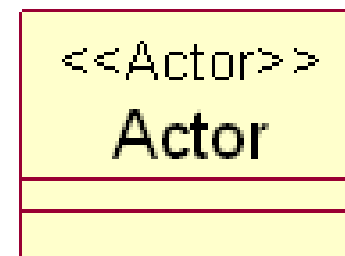
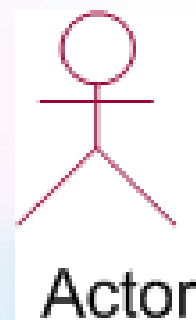


账号



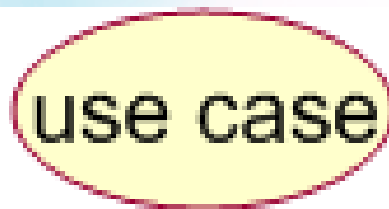
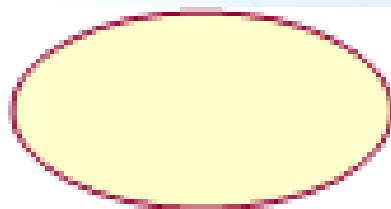
## 参与者 actor

- 参与者是为了完成一个事件而与系统交互的实体
- 可以是人、其它系统、硬件设备、时钟
  - 其它系统：当系统需要与其它系统交互时，如ATM柜员机系统中，银行后台系统就是一个参与者
  - 硬件设备：如果系统需要与硬件设备交互时，如在开发IC卡门禁系统时，IC卡读写器就是一个参与者
  - 时钟：当系统需要定时触发时，时钟就是参与者



## 用例 use case

- 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参与者可见的价值结果。
- 用例是系统提供的功能块，演示了参与者如何使用系统
- 取名时通常使用“**动名词**”形式



use case



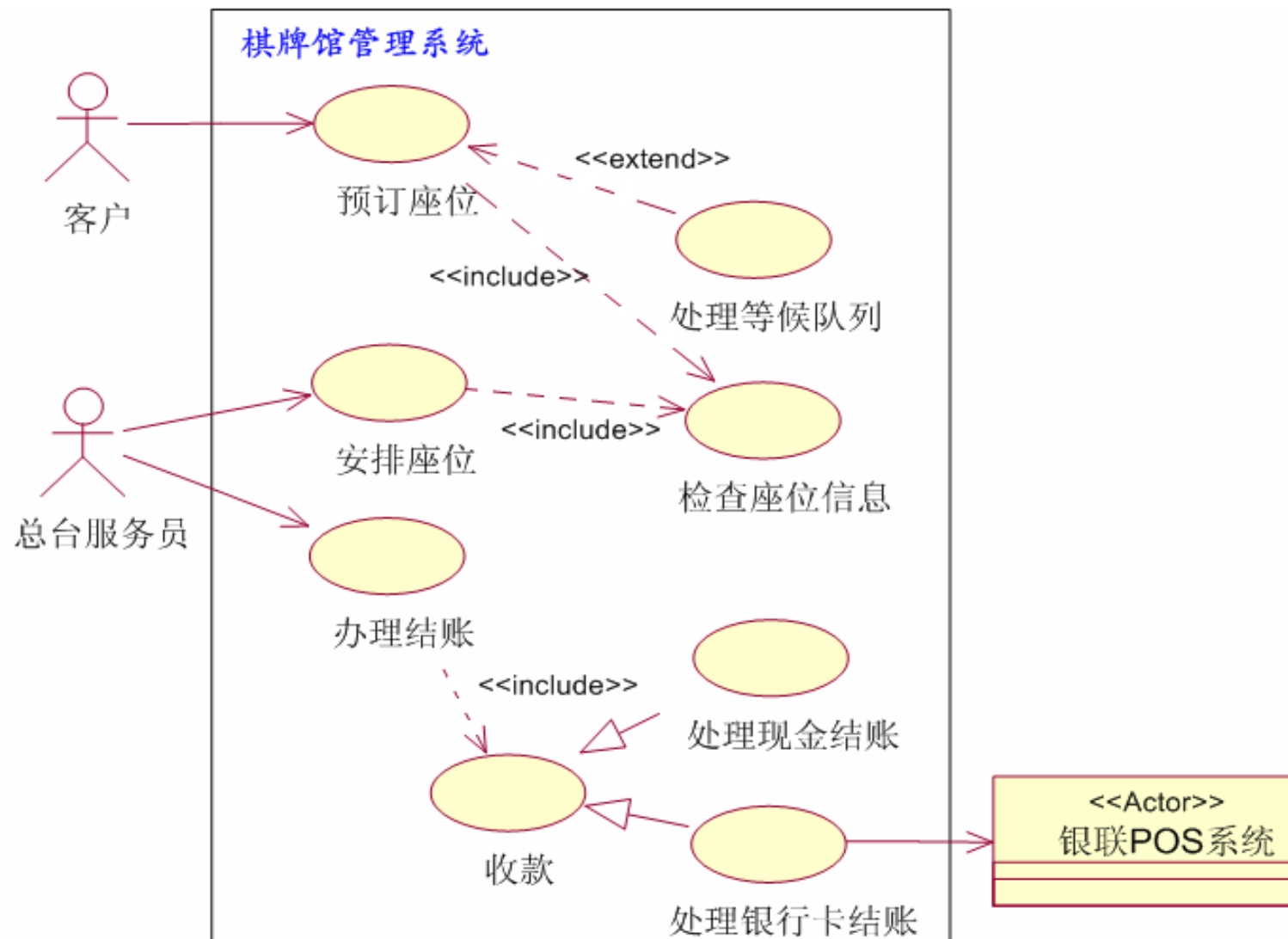
# 本章内容

---

- 什么是面向对象的分析
- 用例和用例驱动开发
- 如何阅读用例图
- 如何绘制用例图
- 用例图应用说明



# 阅读用例图



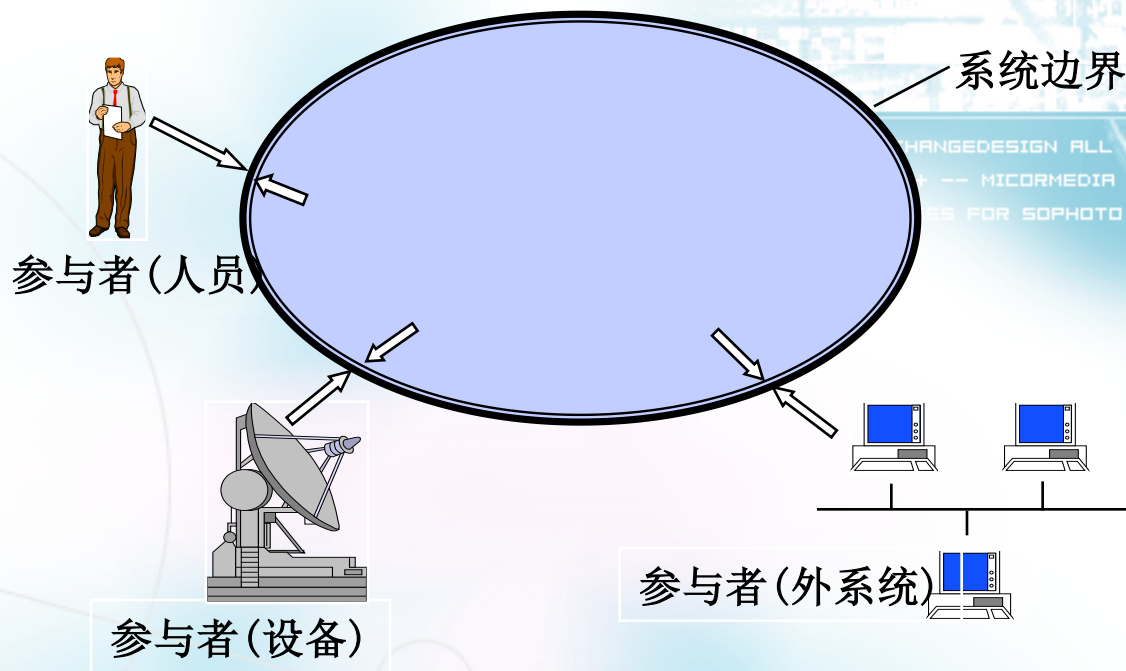
# 用例图的组成元素

## ● 系统边界

一个系统所包含的所有系统成分与系统以外各种事物的分界线。

## ● 参与者：

在系统边界以外，与系统进行交互的事物——人员、设备、外系统



# 用例图的组成元素

- 用例：

对参与者使用系统的一项功能时所进行的交互过程的一个描述。描述手段包括图形和文字。

- 一些关系：

- 参与者与用例的关系：带箭头的线来表示

- 用例之间的关系：

- 1) 包含关系：构造型 **《include》**

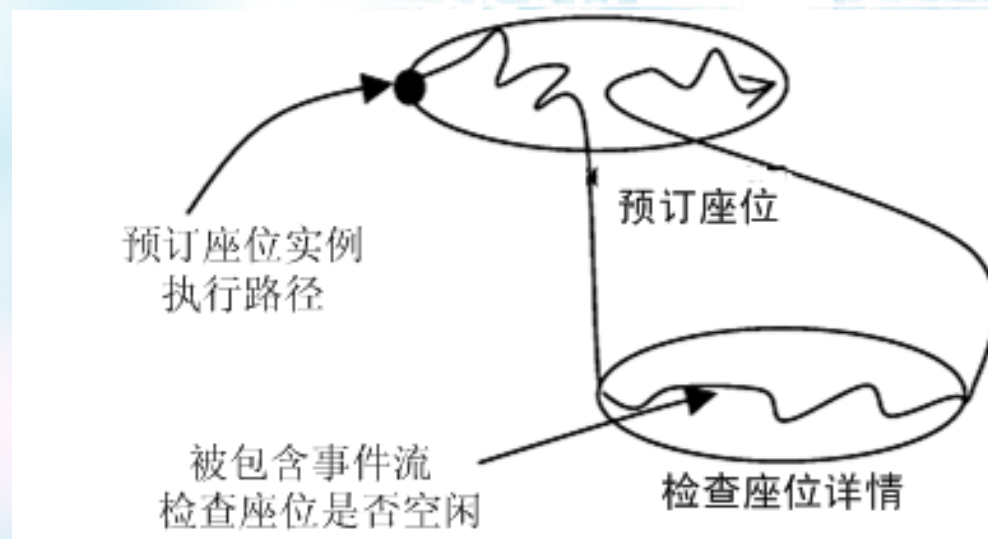
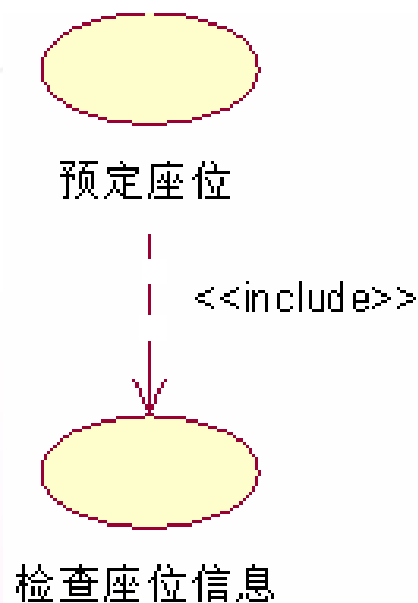
- 2) 扩展关系：构造型 **《extend》**

- 3) 泛化关系



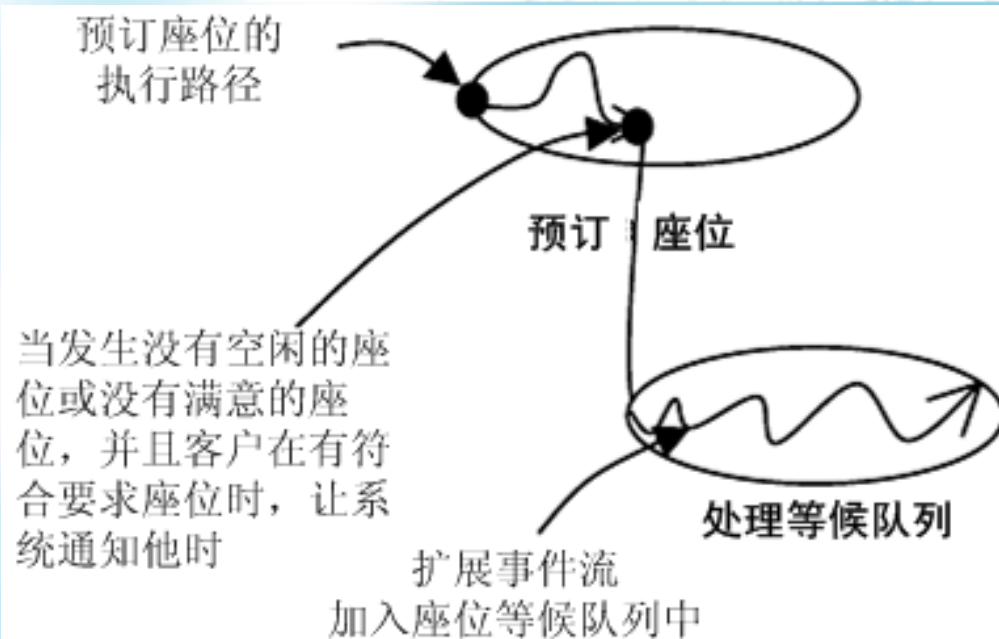
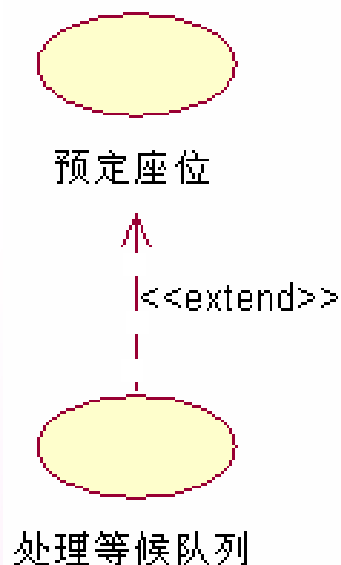
## 包含关系

- 用构造型 **《include》** 表示
- 箭头方向从基用例到被包含用例，指基用例在它内部说明的某一位置上显式地合并了另一个用例的行为



## 扩展关系

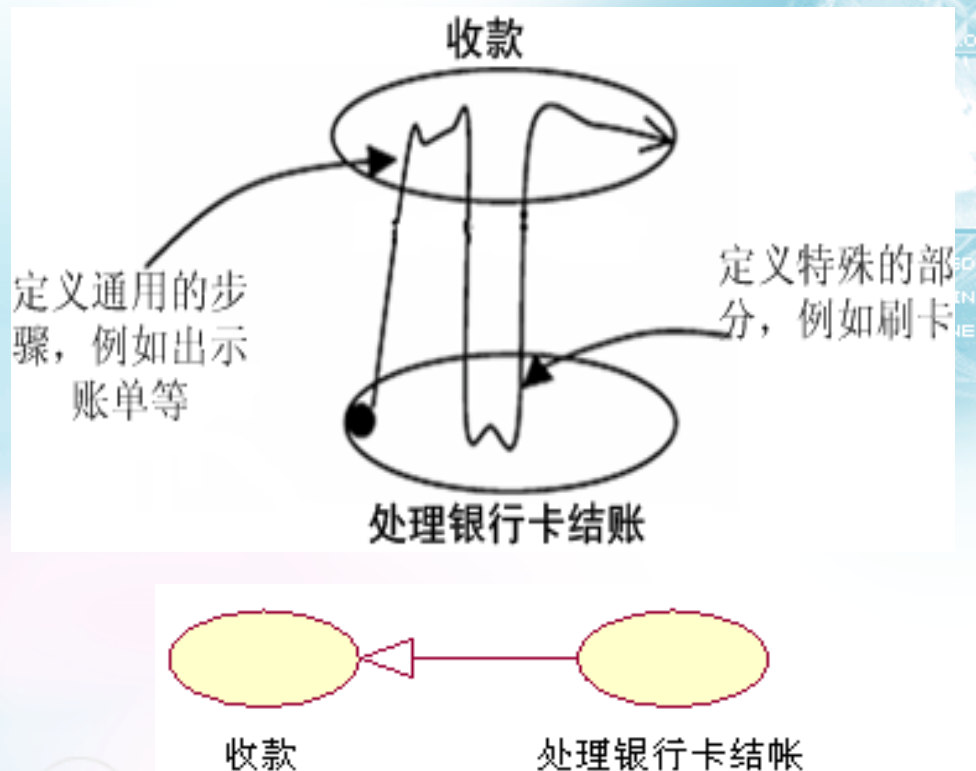
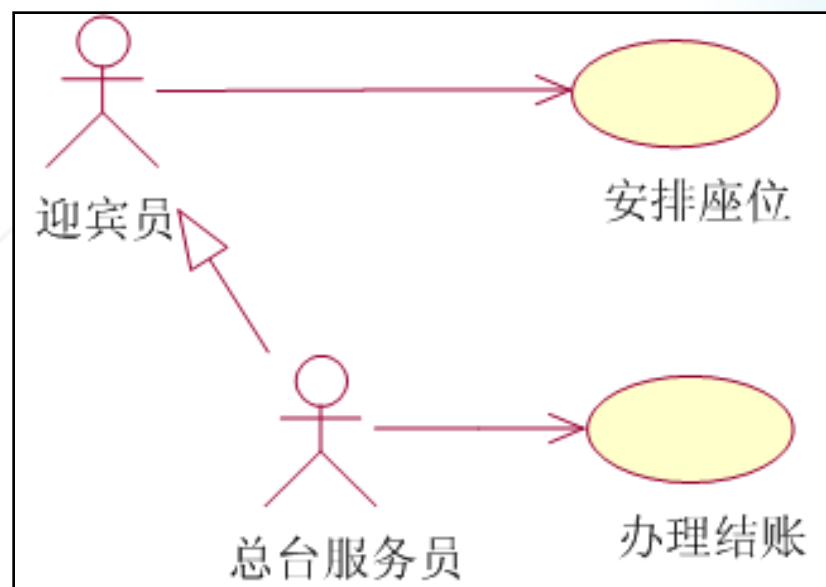
- 用构造型《extend》表示
- 箭头方向从扩展用例到基用例，基用例是可以独立于扩展用例存在的，只是在特定的条件下，它的行为可以被另一个用例的行为所扩展





## 泛化关系

- 可以用来表示参与者与参与者之间，用例与用例之间的特殊/一般化关系



## 读图小结

---

- 这张用例图首先定义了三个基用例：预订座位、安排座位和处理结账
- 客户通过Internet启动“预订座位”用例，在“预订座位”用例的执行过程中，将“检查座位信息”（被包含用例），如果没有空闲的座位或满意的座位，可以选择进入等候队列，这样就将启动扩展用例“处理等候队列”。
- 总台服务员在客户到棋牌馆时，启动“安排座位”用例，在执行过程中，将启动被包含用例“检查座位信息”。
- 当客户要离开棋牌馆时，总台服务员将启动“处理结账”用例，并且定义了两种“收款”用例，一个是“处理现金结账”，另一个是“处理银行卡结账”，而后一个用例将通过与外部系统“银联POS系统”交互来完成。

## 思考题

---

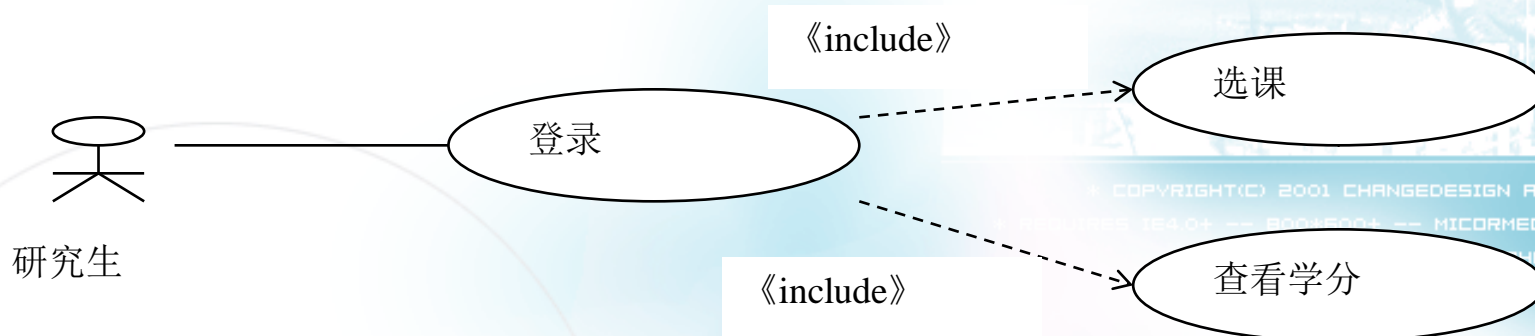
很多软件系统在一开始都需要登录，若用户登录成功，则可进入系统。

如下以一个研究生学籍管理系统为例，描述五种登录方法。

为了简化起见，假设此处仅描述[登录](#)、[选课](#)和[查看学分](#)这3项功能。

## 方案一：

由于选课和查看学分都需要登录，故专门设立一个“登录”用例。若登录成功，则可以进行选课，也可以进行查看学分。

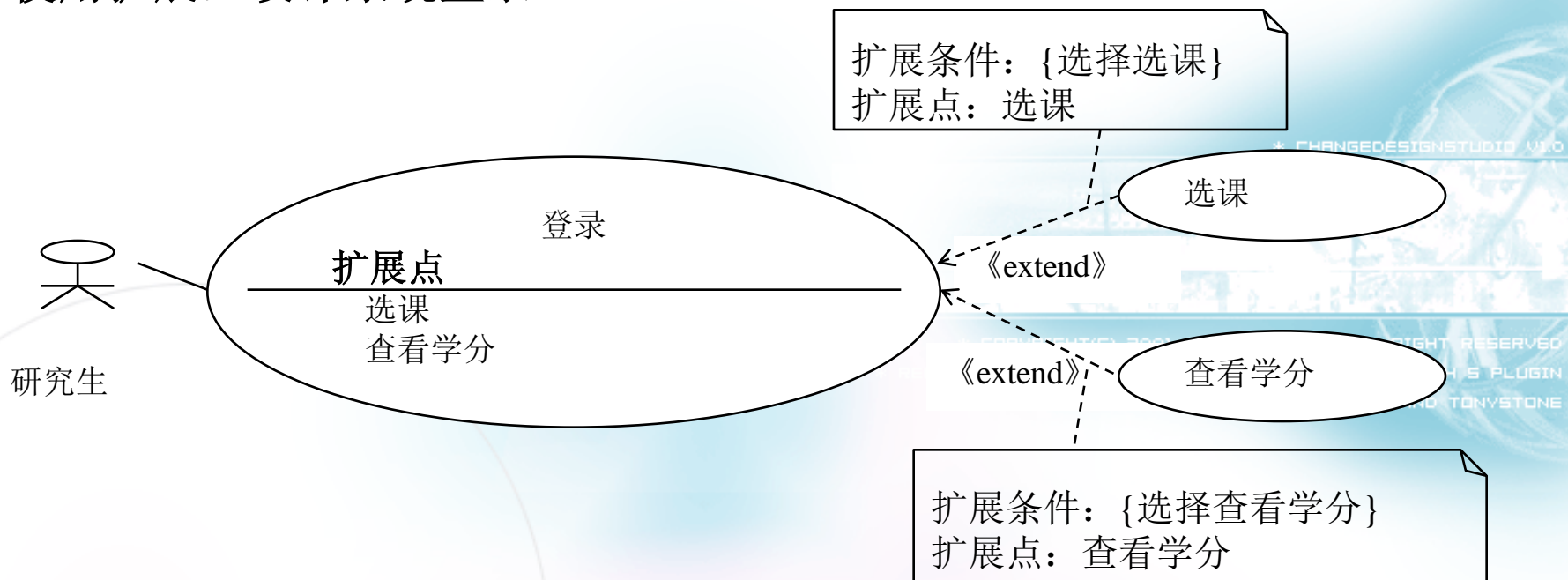


(1) 必须要了解系统的所有其它模块，才能描述清楚“登录”用例。向系统增加新用例时，也要修改登录用例。从维护的角度看，有时会忘记对“登录”用例进行修改。

(2) “登录”用例的功能不单一。

## 方案二：

使用扩展，设计系统登录。

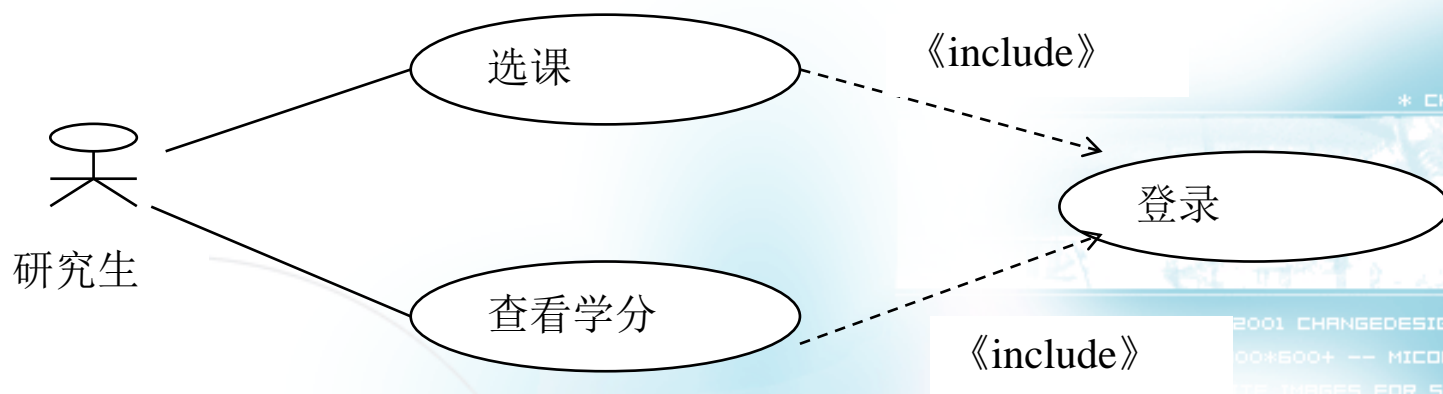


该方法与方法一相比，对“登录”用例的描述要清楚一些。在增加新用例时，仅在登录用例中添加扩展点即可。

缺点：“登陆”用例的功能仍然不单一。

## 方案三：

让所有的相关用例都包含登录用例。



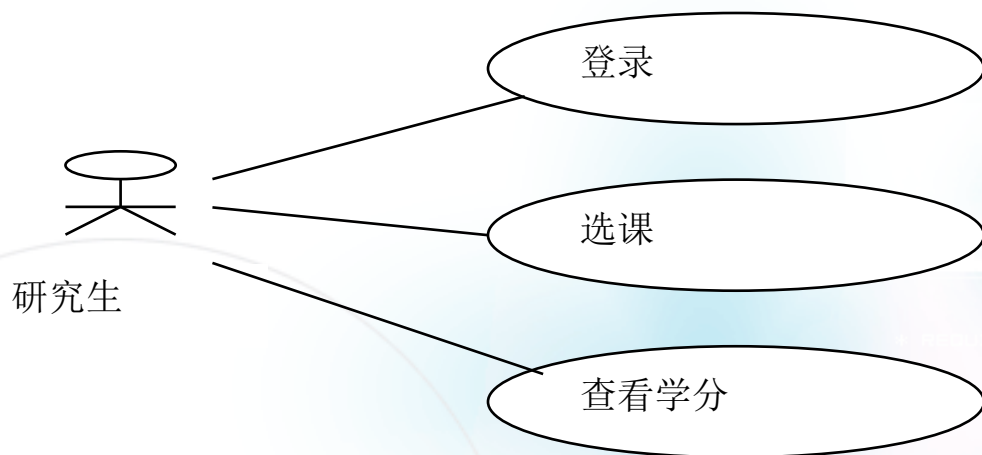
### 缺点：

- (1) 对研究生要进行多次验证。——研究生执行系统的每项功能都要先登录。
- (2) 用例的功能不单一。



## 方案四：

登录用例完全独立于其它用例

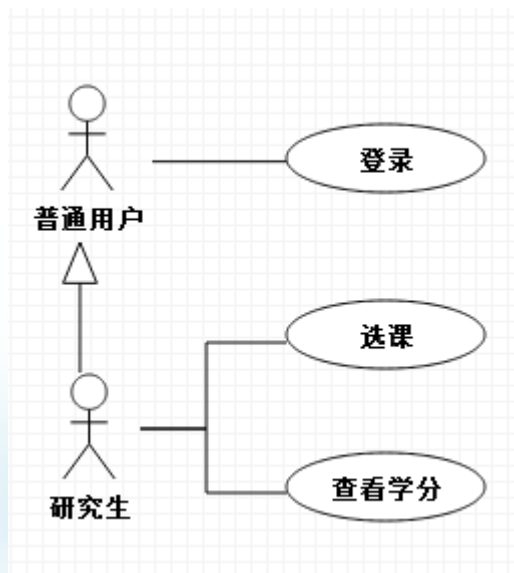


使用该方法，必须要在“选课”用例和“查看学分”用例中指定前置条件：只有在登录成功后才能执行自己。

缺点：在图上不能直接看出用例之间的关系。

## 方案五：

### 使用参与者的泛化表示



登录用例是每个参与者都要用到的。可以抽象出一个“基类”参与者，其他参与者都直接或间接继承于他，该“基类”参与者能执行所有参与者都执行的用例。

在“选课”用例和“查看学分”用例中指定前置条件：只有在登录成功后才能执行自己。

# 用例描述

---

- 用例模型=用例图+用例描述
- 用例描述的是一个系统做什么（what）的信息，并不说明怎么做（how），怎么做是设计模型的事
- 事件流：
  - 前置条件：用例启动时参与者和系统置于什么状态
  - 后置条件：用例结束时系统置于什么状态
  - 基本事件流：对用例中常规、预期的路径的描述
  - 扩展事件流：对一些异常情况、选择分支进行描述

## 用例描述模板

用例编号	[为用例制定一个唯一的编号，通常格式为 <b>UCxx</b> ]	
用例名称	[应为一个动词短语，让读者一目了然地知道用例的目标]	
用例概述	[用例的目标，一个概要性的描述]	
范围	[用例的设计范围]	
主参与者	[该用例的主Actor，在此列出名称，并简要的描述它]	
次要参与者	[该用例的次要Actor，在此列出名称，并简要的描述它]	
项目相关人 利益说明	项目相关人	利益
	[项目相关人员名称]	[从该用例获取的利益]
	.....	.....
前置条件	[即启动该用例所应该满足的条件。]	
后置条件	[即该用例完成之后，将执行什么动作。]	
成功保证	[描述当前目标完成后，环境变化情况。]	
基本事件流	步骤	活动
	1	[在这里写出触发事件到目标完成以及清除的步骤。]
	2	.....(其中可以包含子事件流，以子事件流编号来表示)
扩展事件流	1a	[1a表示是对1的扩展，其中应说明条件和活动]
	1b	.....(其中可以包含子事件流，以子事件流编号来表示)
子事件流	[对多次重复的事件流可以定义为子事件流，这也是抽取被包含用例的地方。]	
规则与约束	[对该用例实现时需要考虑的业务规则、非功能需求、设计约束等]	

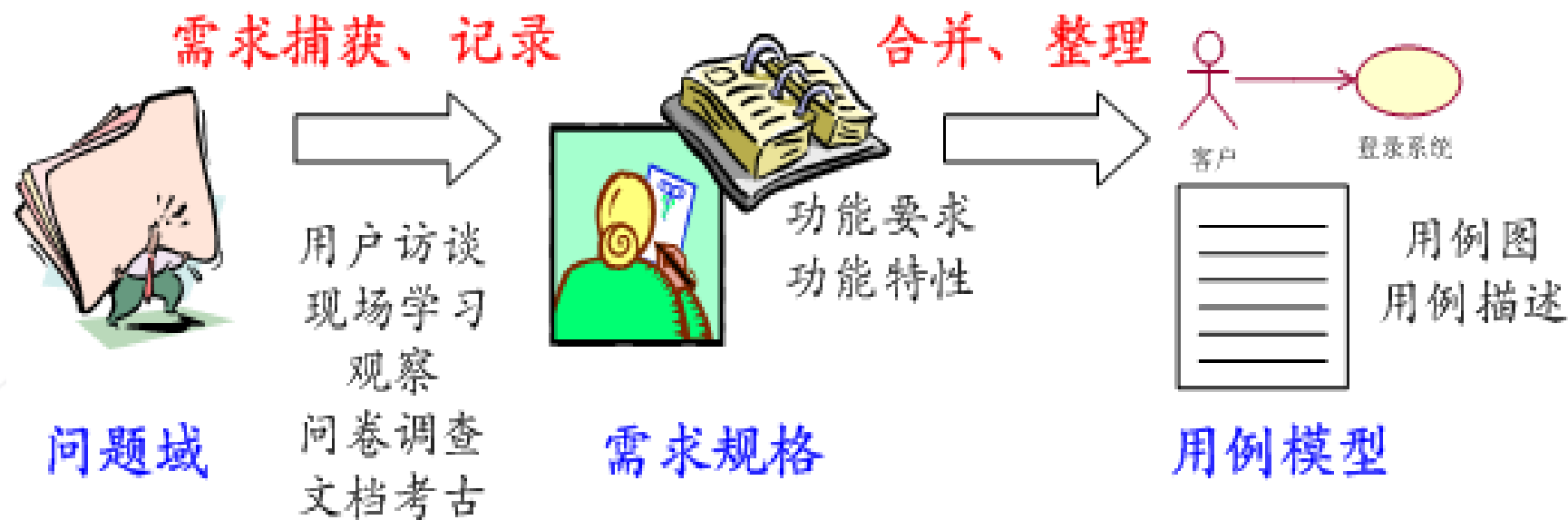
# 本章内容

---

- 什么是面向对象的分析
- 用例和用例驱动开发
- 如何阅读用例图
- 如何绘制用例图
- 用例图应用说明



# 用例图的绘制流程





## 建模用例图之前，请思考以下几个问题？

---

- 什么角色会用这个系统？
- 每个角色通过这个系统能做什么事情？
- 哪些功能是系统必需的功能呢？
- 哪些功能是“锦上添花”的功能？
- 哪些功能应该优先做呢？

\* CHANGEDESIGNSTUDIO V1.0

\* COPYRIGHT(C) 2001 CHANGEDESIGN ALL RIGHT RESERVED

\* RESOURCES: 164.0+ -- 800+600+ -- MICROMEDIA FLASH/5 PLUGIN

\* WEBSITE: JURGAS FOR SOPHOTO AND TONYSTONE

## “个人图书管理系统”需求描述

- 小王是一个爱书之人，家里各类书籍已过千册，而平时又时常有朋友外借，因此需要一个个人图书管理系统。该系统应该能够将书籍的基本信息按计算机类、非计算机类分别建档，实现按书名、作者、类别、出版社等关键字的组合查询功能。在使用该系统录入新书籍时系统会自动按规则生成书号，可以修改信息，但一经创建就不允许删除。该系统还应该能够对书籍的外借情况进行记录，可对外借情况列表打印。另外，还希望能够对书籍的购买金额、册数按特定时间周期进行统计。

## 识别参与者

- 已有的上下文关系图（表示系统范围）及其他相关模型：它们描述了系统与外部系统的边界，从这些图中可以寻找出与系统有交互关系的外部实体。
- 项目相关人员分析：对项目的相关人员进行分析，就能够决定出哪些人将会与系统进行交互。
- 书面的规格说明和其它项目文档（如会谈备忘录等）
- 需求研讨会和联合应用开发会议的记录：这些会议的参与者通常是很重要的，因为他们在组织中所代表的角色就是可能与系统发生交互的参与者。
- 当前过程和系统的培训指南及用户手册：这些东西中经常会有潜在参与者。

# 记录需求—特性表

## ● 举例：“个人图书管理系统”

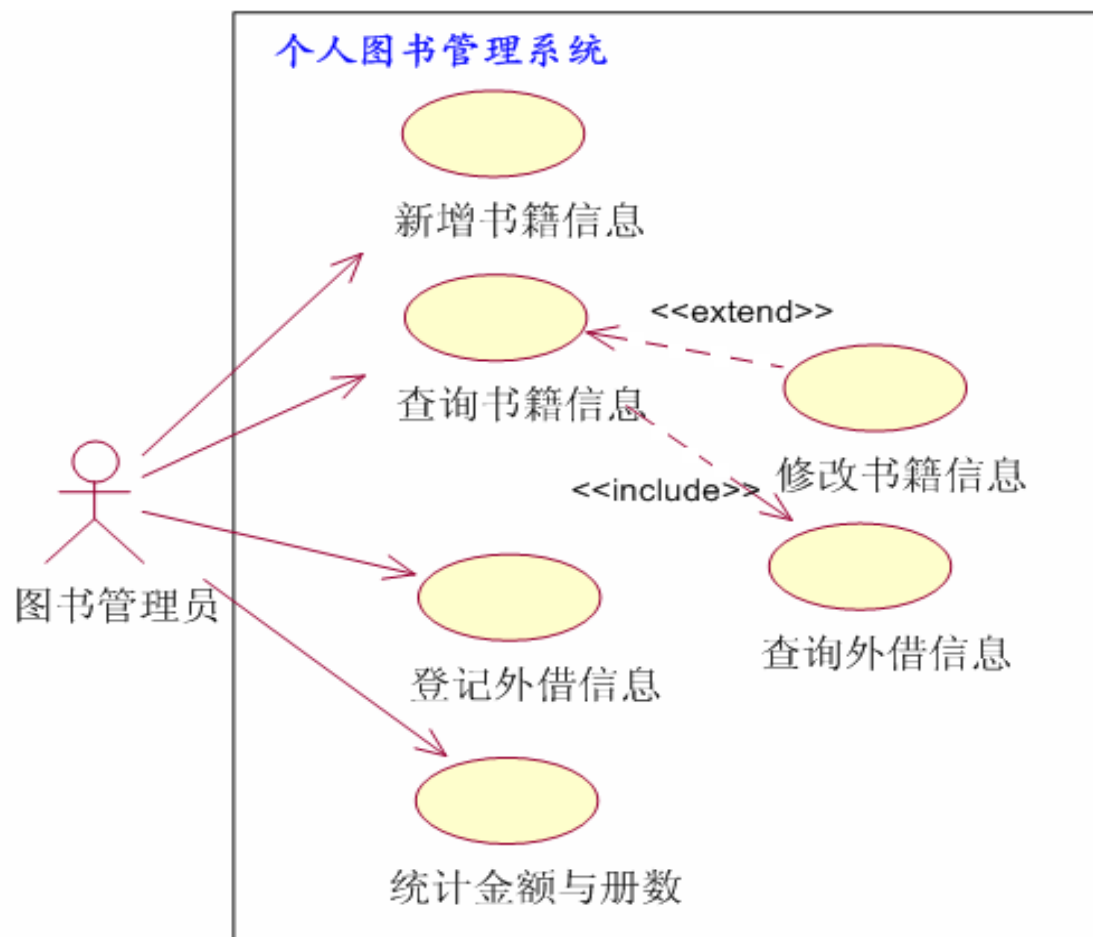
编号	说明
FEAT01	新增书籍信息
FEAT02	修改已有的书籍信息
FEAT03	书籍信息按计算机类、非计算机类分别建档
FEAT04	录入新书时能够自动按规则生成书号
FEAT05	计算机类与非计算机类书籍采用不同的书号规则
FEAT06	录入新书时如果重名将自动提示
FEAT07	按书名、作者、类别、出版社等关键字组合查询书籍
FEAT08	列出所有书籍信息
FEAT09	记录外借情况
FEAT10	外借状态能够自动反应在书籍信息中
FEAT11	按人、按书查询外借情况
FEAT12	列出所有的外借情况
FEAT13	按特定时间段统计购买金额、册数
FEAT14	所有查询、列表、统计功能应可以单独对计算机类或非计算机类进行

# 合并需求获得用例

特性	用例
FEAT01.新增书籍信息 FEAT03.书籍信息按计算机类、非计算机类分别建档 FEAT04.录入新书时能够自动按规则生成书号 FEAT05.计算机类与非计算机类书籍采用不同的书号规则 FEAT06.录入新书时如果重名将自动提示	UC01.新增书籍信息
FEAT02.修改已有的书籍信息	UC02.修改书籍信息
FEAT07.按书名、作者、类别、出版社等关键字组合查询书籍 FEAT08.列出所有书籍信息 FEAT14.所有查询、列表、统计功能应可以单独对计算机类或非计算机类进行	UC03.查询书籍信息
FEAT09.记录外借情况 FEAT10.外借状态能够自动反应在书籍信息中	UC04.登记外借信息
FEAT11.按人、按书查询外借情况 FEAT12.列出所有的外借情况 FEAT14.所有查询、列表、统计功能应可以单独对计算机类或非计算机类进行	UC05.查询外借信息
FEAT13.按特定时间段统计购买金额、册数 FEAT14.所有查询、列表、统计功能应可以单独对计算机类或非计算机类进行	UC06.统计金额和册数



## 绘制用例图





## 细化用例描述—搭框架

- 1.用例名称：新增书籍信息（UC01）
- 2.简要说明：录入新购书籍信息，并自动存储建档。
- 3.事件流：
  - 3.1 基本事件流
  - 3.2 扩展事件流
- 4.非功能需求
- 5.前置条件：用户进入图书管理系统。
- 6.后置条件：完成新书信息的存储建档。
- 7.扩展点：无
- 8.优先级：√高 中 低

见下页ppt

## 细化用例描述—填血肉

---

### 3.事件流:

#### 3.1 基本事件流

- 1) 图书管理员向系统发出“新增书籍信息”请求;
- 2) 系统要求图书管理员选择要新增的书籍是计算机类还是非计算机类;
- 3) 图书管理员做出选择后, 显示相应界面, 让图书管理员输入信息, 并自动根据书号规则生成书号;
- 4) 图书管理员输入书籍的相关信息, 包括: 书名、作者、出版社、ISBN号、开本、页数、定价、是否有CDROM;
- 5) 系统确认输入的信息中书名未有重名;
- 6) 系统将所输入的信息存储建档。

#### 3.2 扩展事件流

- 5a) 如果输入的书名有重名现象, 则显示出重名的书籍, 并要求图书管理员选择修改书名或取消输入;
  - 5a1) 图书管理员选择取消输入, 则结束用例, 不做存储建档工作;
  - 5a2) 图书管理员选择修改书名后, 转到5)

### 4.非功能需求: 无特殊要求

# 本章内容

---

- 什么是面向对象的分析
- 用例和用例驱动开发
- 如何阅读用例图
- 如何绘制用例图
- 用例图应用说明



\* COPYRIGHT(C) 2001 CHANGEDESIGN ALL RIGHT RESERVED  
\* REQUIRES IE4.0+ -- 800\*600+ -- MICROMEDIA FLASH/5 PLUGIN  
\* WEBSITE CHARGES FOR SOPHOTO AND TONYSTONE



# 用例模型的运用

---

- 为开发者提供一种认识和理解系统的方法
- 为领域专家、最终用户和开发者提供一种相互交流的手段。
- 易于对需求规范化

用户给出的需求材料常常是不规范或不够准确的。故要较为规范地定义用例，以全面和比较准确地用用例中表达用户的功能需求。

- 对系统测试来说，产生测试用例。
- 有助于设计人机界面
- .....

# 建模要点

---

- 构建结构良好的用例：
  - 1) 为系统和部分系统中单个的、可标识和合理的原子行为命名；
  - 2) 将公共的行为抽取出来，放到一个被包含用例中，再将它《include》进来；
  - 3) 对于变化部分，将其抽取出来，放到一个扩展用例（用《extend》连接）中；
  - 4) 清晰地描述事件流，使得读者能够轻而易举地理解
- 构建结构良好的用例图：摆放元素时，应该避免交叉线的出现；对于语义上接近的行为和角色，最好使它们在物理上也更加接近；
- 根据系统实际情况控制**粒度**：如 **CRUD**的处理

## 对实际项目中用例图组织的一些建议

- 画一个表示系统宏观需求的总用例图，使用系统边界，每个用例用高度概括的语言命名。
- 将总用例图分解为多个具体的用例图，当用例较多、层次较复杂时，可分层次展开用例图，多次分解。
- 通过包对用例进行适当分类
- 用户角色较多时，先单独画出角色（参与者）之间关系，并列一个表格说明每个角色在本系统期望解决的问题、关注点等。



## 本章内容回顾

---

- 什么是面向对象的分析
- 用例驱动开发的概念
- 参与者和用例的概念
- 阅读用例图的方法，包括系统边界、包含关系、扩展关系以及泛化关系
- 绘制用例图的方法
- 用例图的应用