用例和用例图

内容

- 用例模型概述
- 用例图
- 用例
- 用例建模过程

内容

- 用例模型概述
- 用例图
- 用例
- 用例建模过程

用例模型概述

用例模型：

- 描述用户怎样使用系统。
- 描述参与者如何与系统进行交互。
- 强调从最终用户的角度来理解软件系统的需求。
- 浅显易懂，方便客户和系统设计者之间的交流

用例模型概述

用例图：

- 描述参与者和用例及其之间的关系。
- 描述谁使用系统的什么功能。

用例模型概述

用例：

- 是对一个参与者使用系统的一项功能时所进行的交互过程的一个文字描述序列。
- 是外部参与者和系统交互的动作序列的说明，包括可选的动作序列和会出现异常的动作序列。

用例模型概述

用例从使用者观点描述软件。从使用者观点来描述软件是非常适合的，因为软件最终是否符合需求是由使用者决定的。

使用者观点告诉需求收集人员，他希望这个系统是什么样，他将怎样使用这个系统，希望获得什么结果。那么软件只需要按照使用的要求提供一个实现，就不会偏离使用者的预期。

用例模型概述

- 用例是代表系统中各项目相关人员之间就系统行为所达成的契约。
- 用例驱动软件开发过程：用例把软件开发过程的各个阶段捆绑在一起，用例分析结果为预测系统开发时间和预算提供依据。

内容

- 用例模型概述
- 用例图
- 用例
- 用例建模过程

用例图

- 用例图(use case diagram)是显示一组用例、参与者以及它们之间的关系的图
- 用例图组成的三要素：
  - 用例（use case）
  - 参与者（actor）
  - 参与者以及用例之间的关系（relationship）

参与者

- 参与者是系统以外的与系统交互的人或事物，包括人、设备、外部系统等。
- 其它译名有活动者、角色、执行者、行动者等。

参与者的相关说明

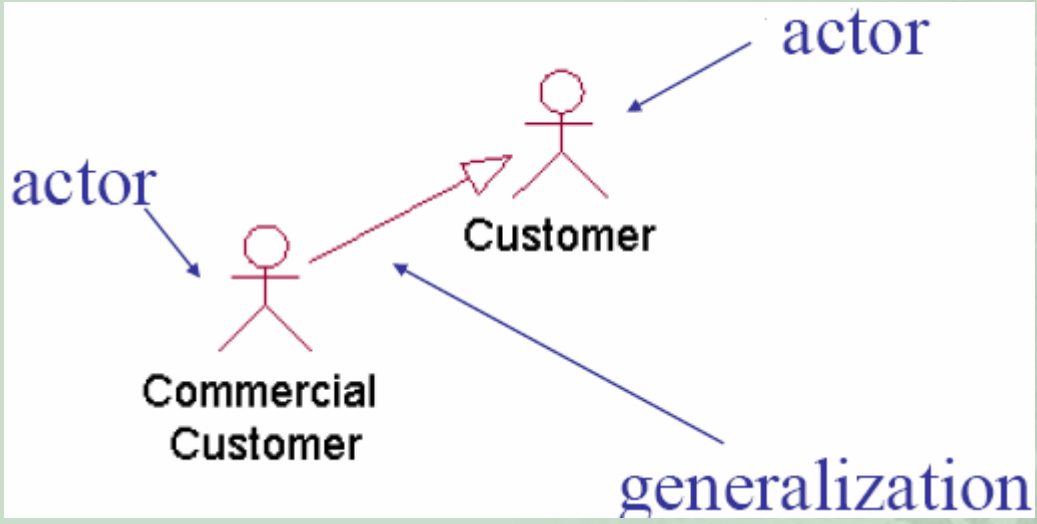
- UML建模是以人为本的。
- 参与者位于系统边界之外。尽管在用例建模时使用参与者，但参与者实际上并不是系统的一部分。
- 一个参与者可以使用多个用例。
- 一个用例也可以被多个参与者所使用。

参与者的表示

- UML中的参与者实际上是一个版型化的类，可以有三种表示形式：

参与者的泛化关系

- 参与者之间可以存在泛化关系，表示一个一般性的参与者与另一个更为特殊的参与者之间的联系。



参与者的获取

可以从以下几点考虑寻找系统的参与者：

- 谁使用系统的主要功能
- 谁需要系统支持完成日常工作
- 谁来维护、管理，使系统正常工作
- 对系统产生的结果感兴趣的人或物
- 系统需要操纵哪些硬件
- 需要与哪些其他系统交互

ATM自动取款机系统的参与者

- 客户
- 银行职员
- 信用系统

用例

- 用例定义了一组相关的由系统执行的动作序列，将有价值的可见结果提供给某个参与者。
- UML的用例图中用例表示为椭圆。用例一般用动宾结构命名，使用业务术语，而不是技术术语。

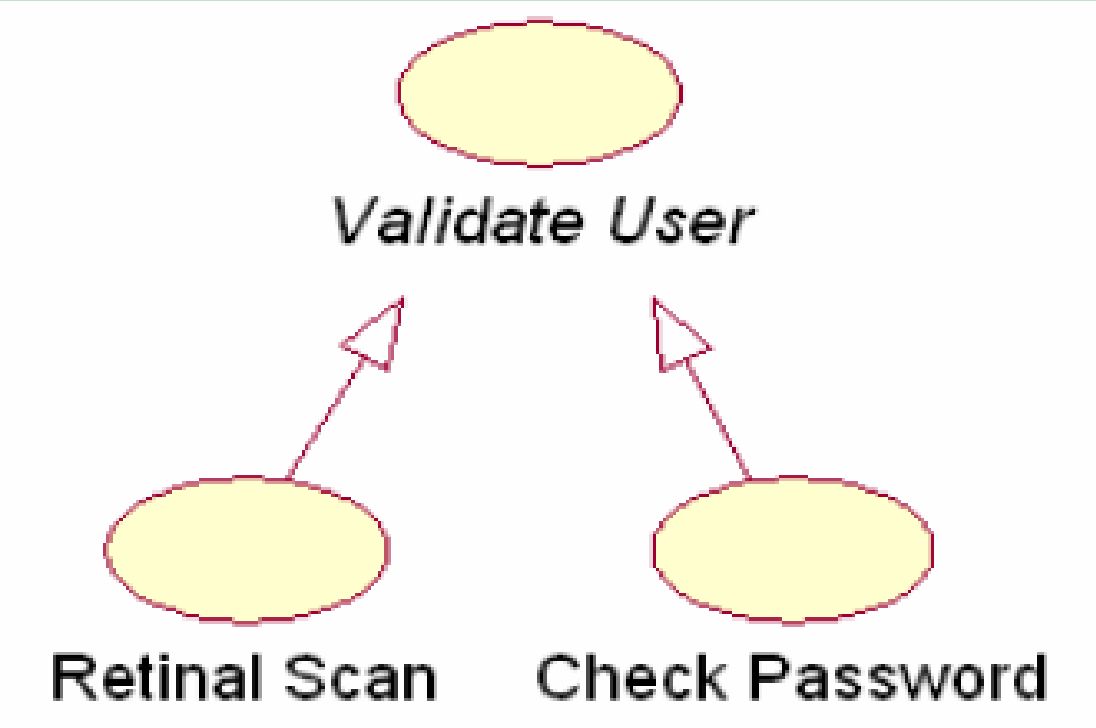
用例之间的关系

- 泛化（generalization）：抽取一些用例共同的属性，定义一个基础用例
- 包含（include）：一个用例的执行要用到另外的用例
- 扩展（extend）：把新的用例（功能）插入到已有用例（基础用例）

泛化关系

泛化(generalization)代表一般与特殊的关系，用例之间的泛化关系与类之间的泛化关系（继承关系）类似。

泛化关系例子



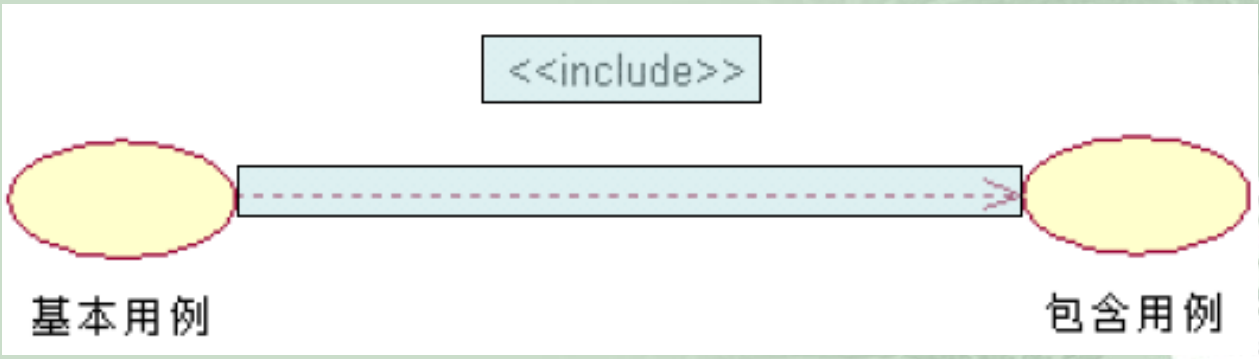
泛化关系

泛化关系的说明：

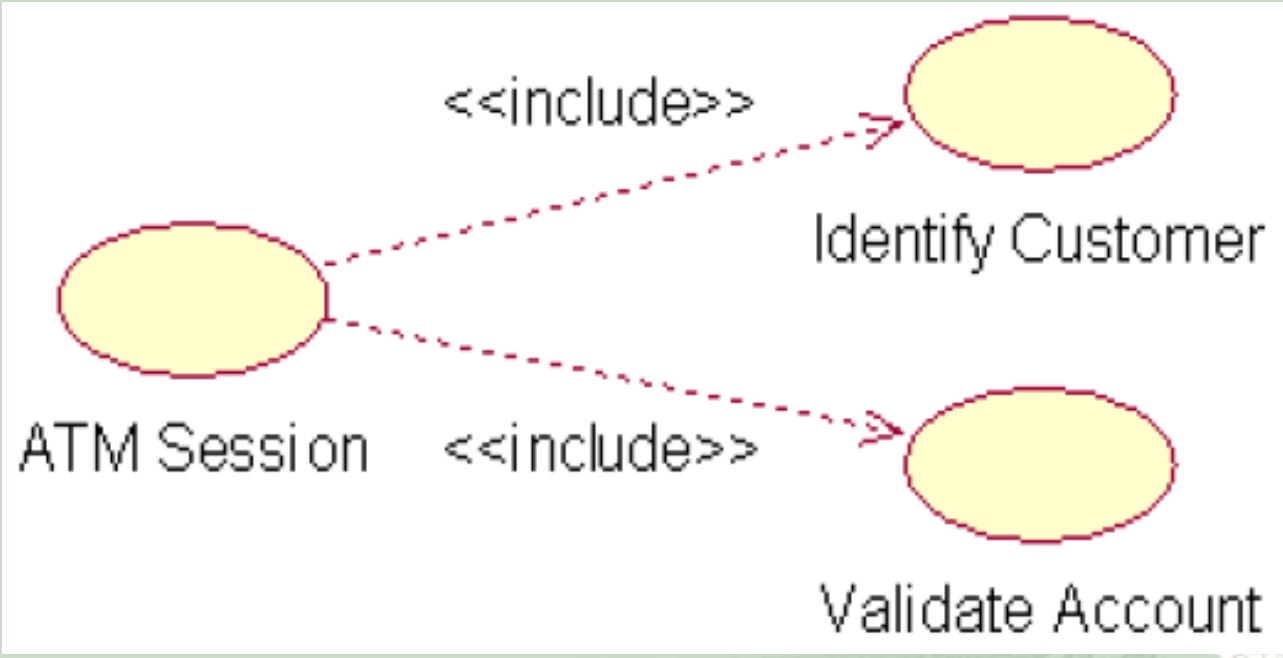
- 用例之间的泛化表示子用例继承了父用例的行为和含义，子用例还可以增加或者覆盖父用例的行为

包含关系

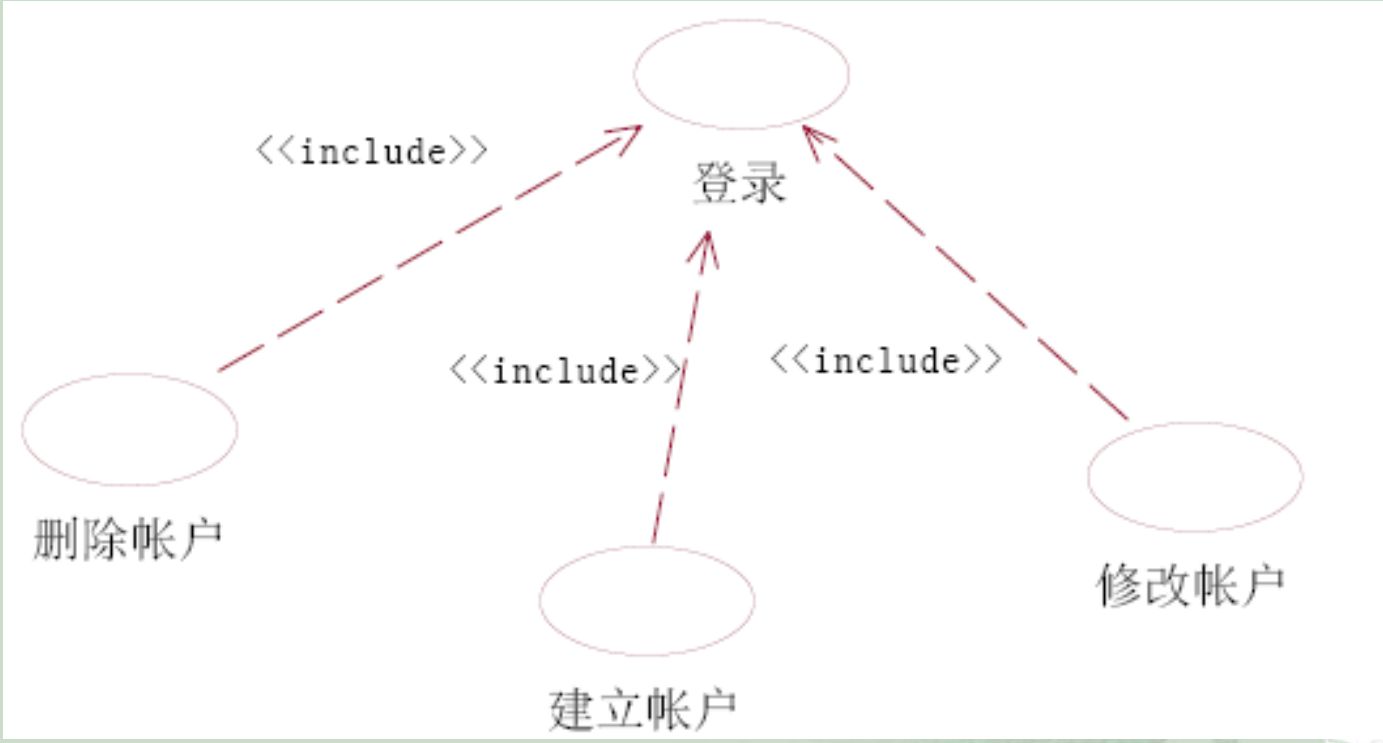
- 包含关系是指一个用例(基本用例，base use case)的行为包含了另一个用例(包含用例,inclusion use case)的行为。
- 包含关系是依赖关系的版型, 但其含义更多。



包含关系例子

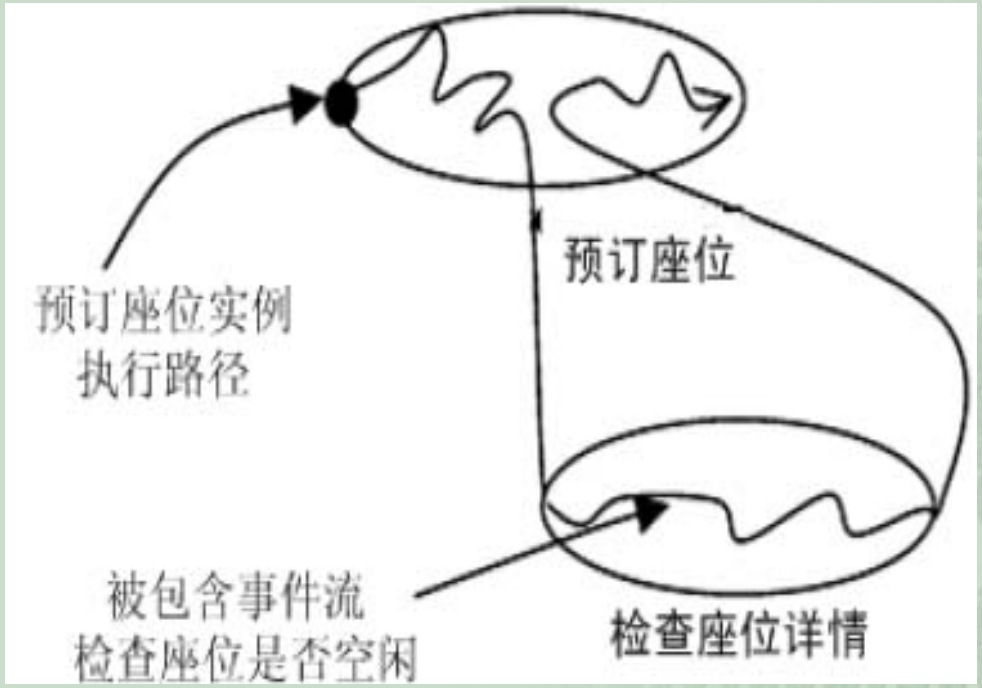

<<include>>
Identify Customer
ATM Session
<<include>>
Validate Account

包含关系例子



包含关系的说明

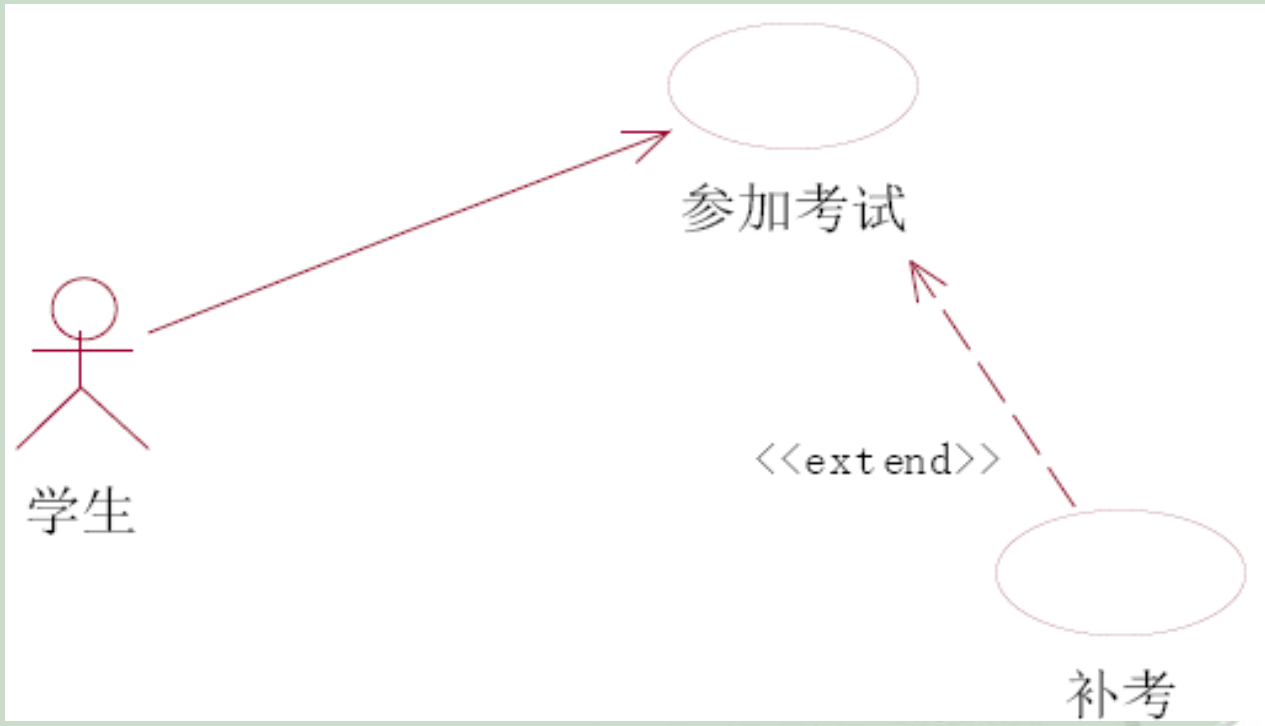
- 在包含关系中，箭头的方向是从基本用例到包含用例，也就是说基本用例依赖于包含用例。



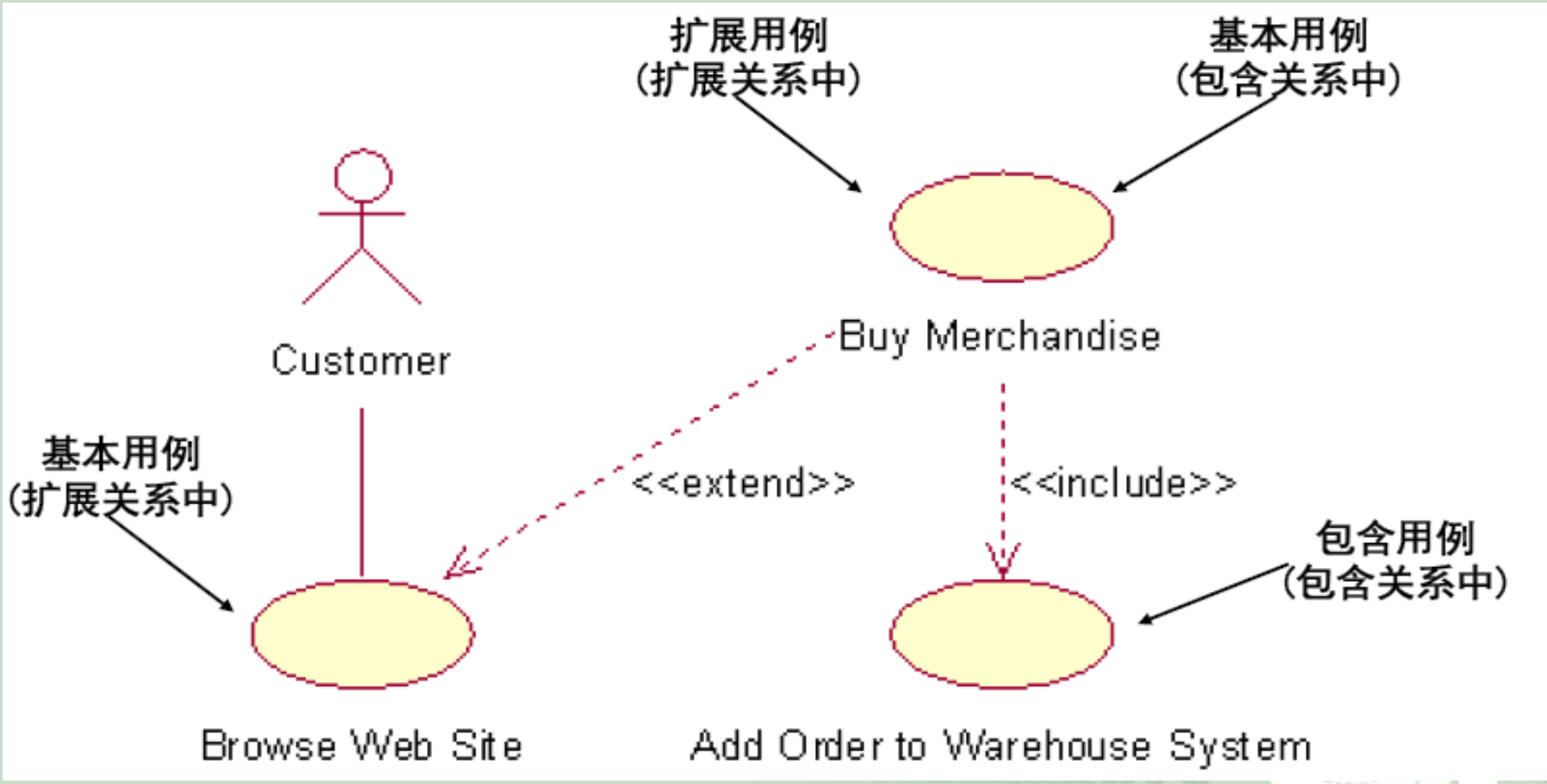
扩展关系

- 表示基本用例在某个条件成立时，合并执行扩展用例。基本用例独立于扩展用例而存在，只是在特定的条件下，它的行为可以被另一个用例(扩展)所扩展。
- 扩展关系是依赖关系版型。

扩展关系例子



扩展关系和包含关系的例子



用例之间几种关系的比较

- 在扩展关系中，基本用例可以独立存在。一个基本用例执行时，可以执行、也可以不执行扩展部分。
- 在包含关系中，在执行基本用例时，一定会执行包含用例部分。

用例的获取

Jacobson建议通过回答下列问题来帮助发现用例：

- 参与者的主要任务是什么?
- 参与者需要了解系统的什么信息?
- 参与者需要修改系统的什么信息?
- 参与者是否需要把系统外部的变化通知系统?
- 参与者是否希望系统把异常情况通知自己?

ATM自动取款机系统的部分用例

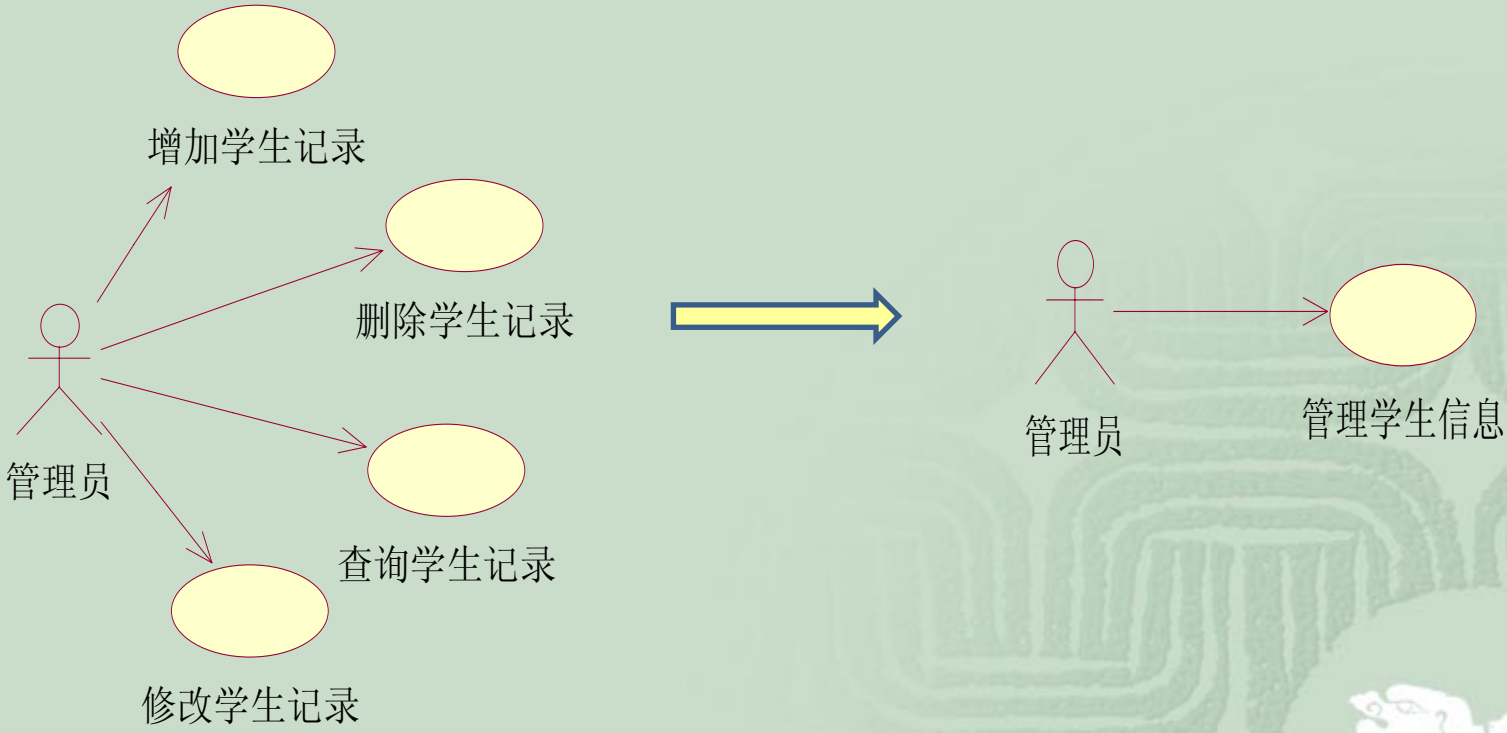
- 取款
- 存款
- 转账
- 查询余额
- 修改密码
- 还款
- ……

用例粒度

从具体的用户目标出发，用例的粒度大小一般应符合“可见的价值结果”的原则。用例粒度的划分要注意以下几点：

- 粒度大小要合适
- 粒度没有一个统一的标准
- 在一张图中，用例之间应具有大体一致的粒度

用例粒度



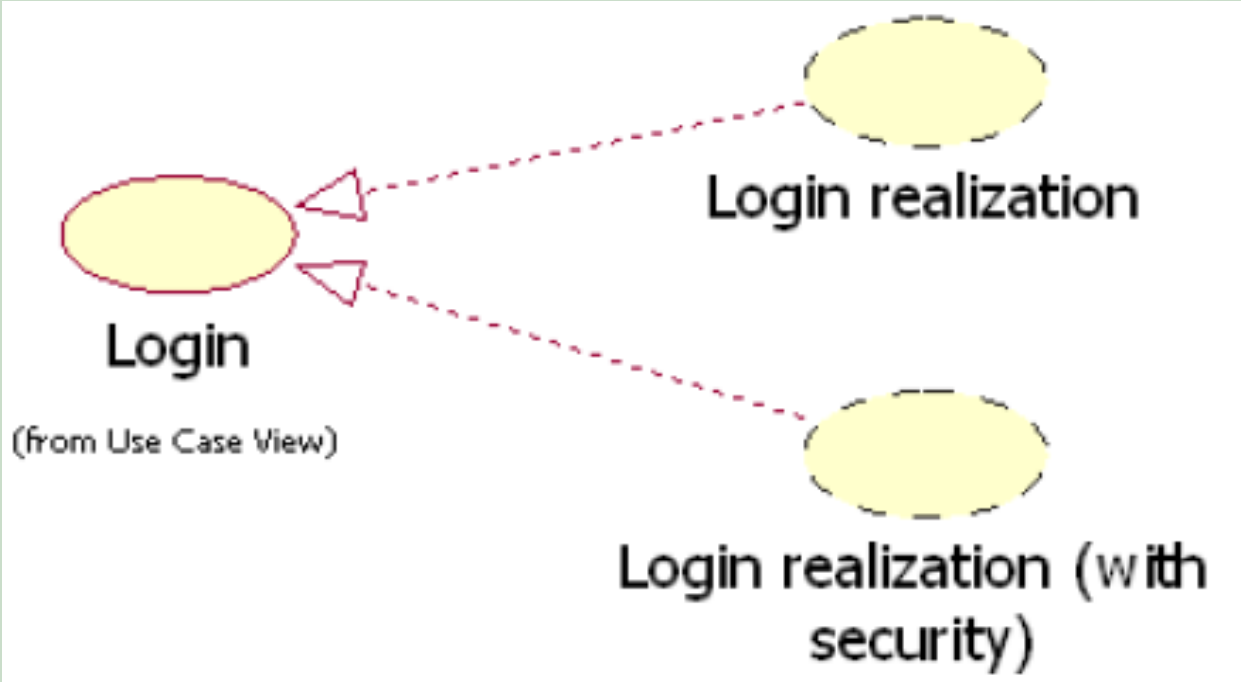
用例的实现

- 用例是与实现无关的关于系统功能的描述。
- 在UML中，可以用协作（collaboration）

来说明对用例的实现。

用例实现的表示

- 在UML中，协作用虚线椭圆表示。

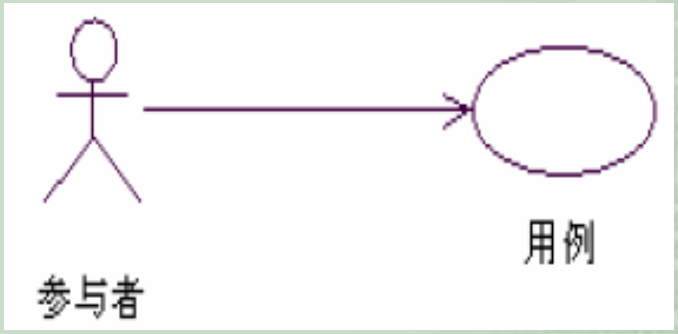


参与者与用例之间关系

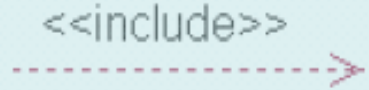
- 用例图中，用例和参与者之间的关联描述执行者所代表的系统外部实体与该用例所描述的系统需求有关，即“谁使用哪个用例”。

参与者与用例之间关系

- 参与者和用例之间是关联关系，表示参与者与用例间的通信
- 表示方法：可以用一条实线箭头表示，由参与者指向用例



参与者、用例之间关系的表示

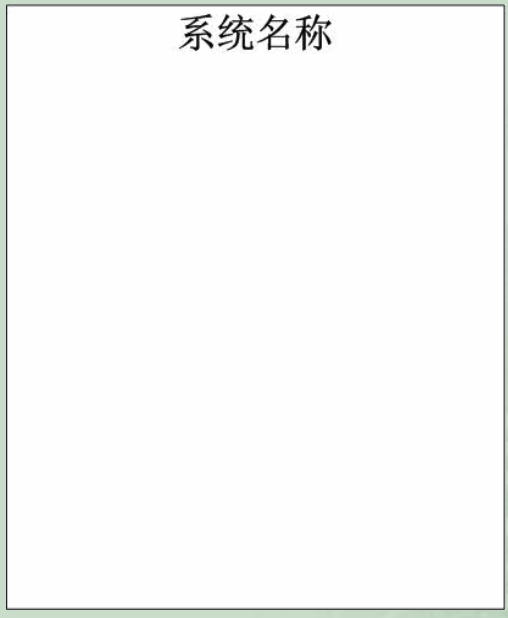
关系类型	说明	表示符号
关联(association)	参与者和用例之间的关系	
泛化(generalization)	参与者之间或用例之间的关系	
包含(include)	用例之间的关系	
扩展(extend)	用例之间的关系	

对环境建模

- 对于一个系统，会有一些事物存在于其内部，而一些事物存在于其外部。
- 存在于系统内部的事物的任务是完成系统外部事物所期望的系统行为。
- 存在于系统外部并与其进行交互的事物构成了系统存在的环境。

系统边界

- 系统边界决定了系统内外的界线。
- 系统边界在UML中表示为一个矩形框。



系统边界

- 边界本质上是面向对象方法的一个很重要的概念。面向对象里任何对象都有一个边界。外界只能通过这个边界来认识对象与对象打交道，而对象内部则是一个禁区。
- 在收集需求时，我们先假定一个范围边界，在这个边界内寻找需求，而找到的需求集合又决定了最终边界的范围。

用例图

- 用例图描绘拟建系统和外部环境的关系。
- 用例从使用系统的角度描述系统中的信息。
- 用例描述用户提出的一些可见需求，对应一个具体的用户目标。

UML2.2中的图

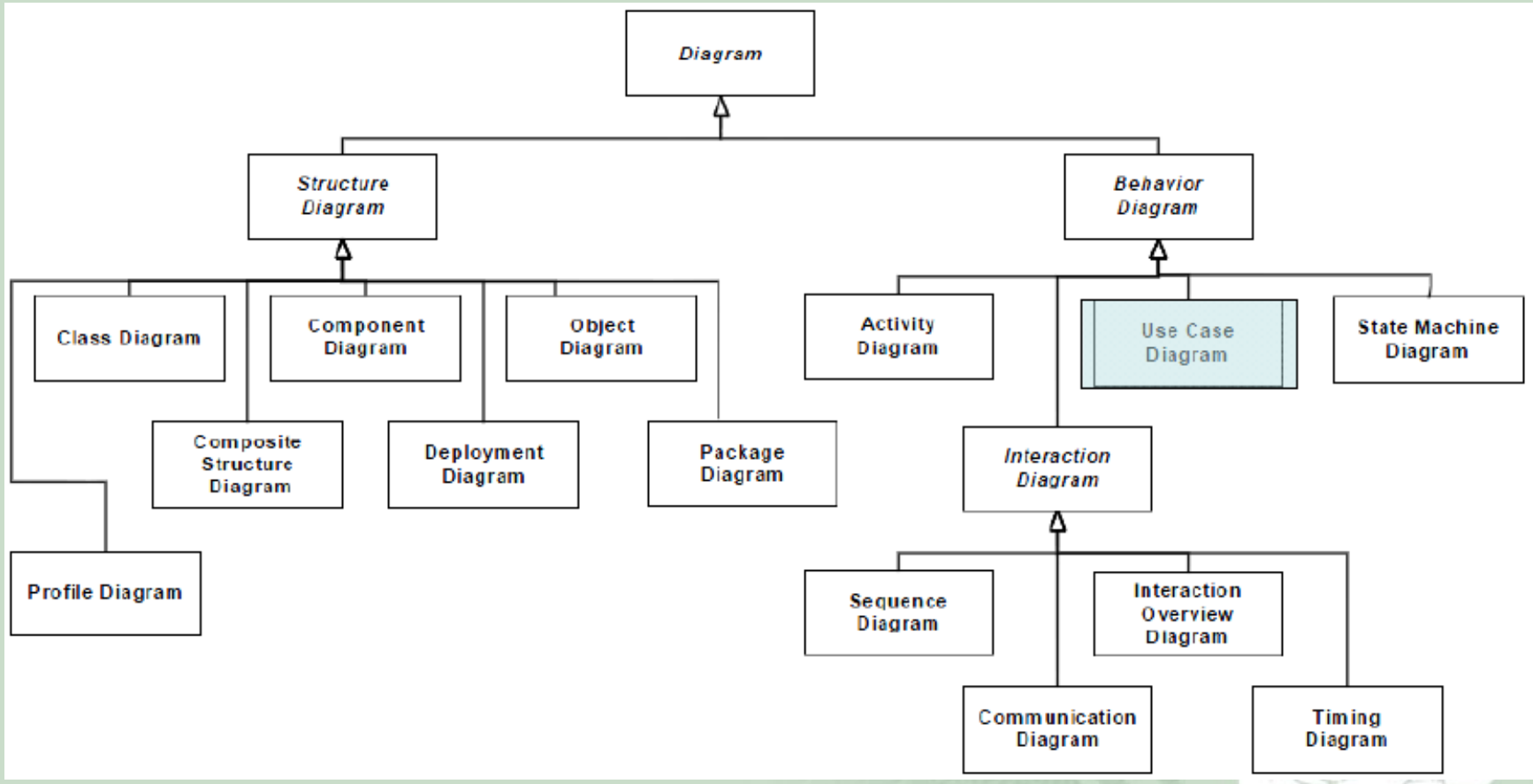
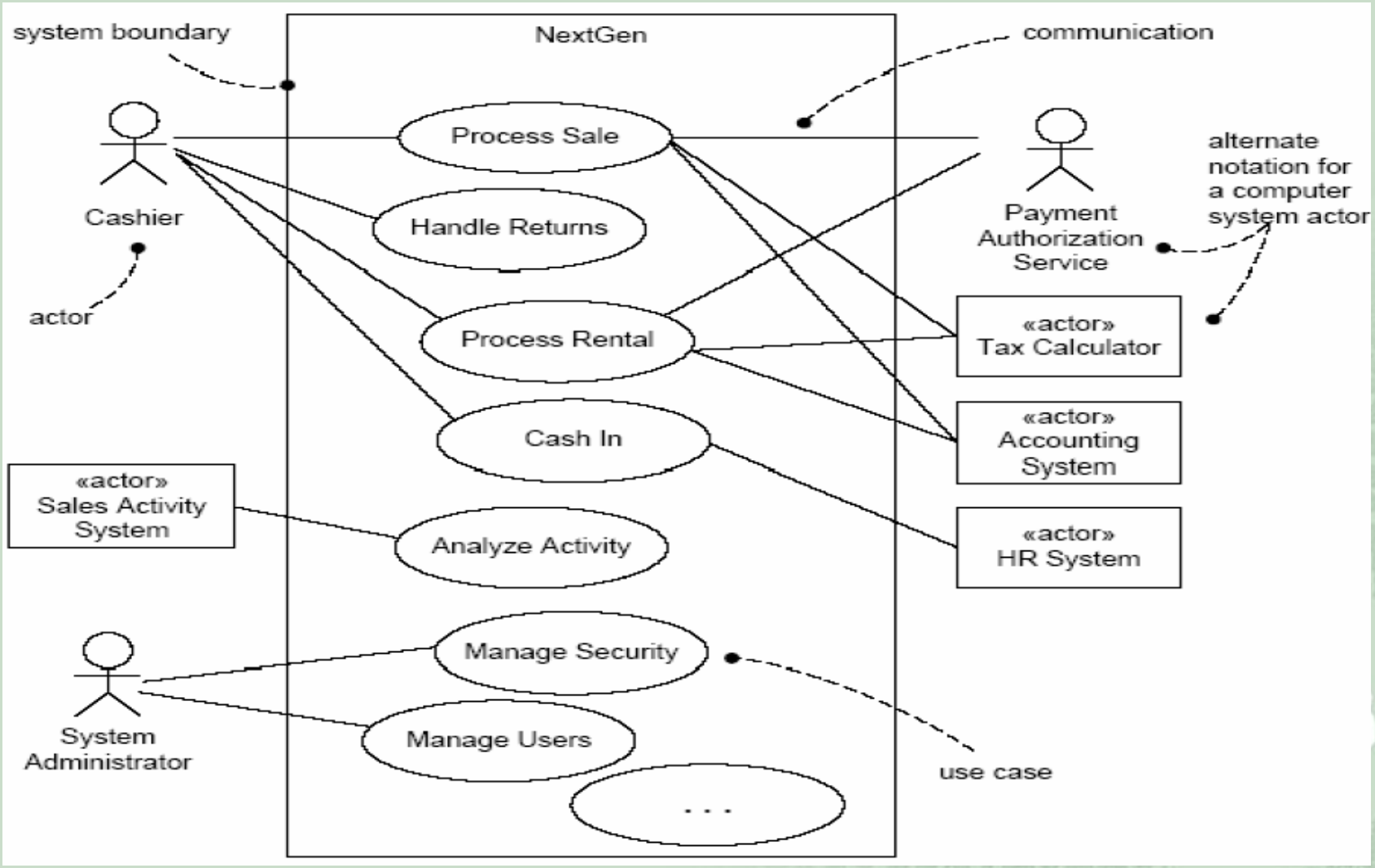


Diagram
Structure Diagram
Behavior Diagram
Class Diagram
Component Diagram
Object Diagram
Activity Diagram
Use Case Diagram
State Machine Diagram
Composite Structure Diagram
Deployment Diagram
Package Diagram
Interaction Diagram
Profile Diagram
Sequence Diagram
Interaction Overview Diagram
Communication Diagram
Timing Diagram

用例图中的主要图表说明

用例图


system boundary
NextGen
communication
Process Sale
Handle Returns
Process Rental
Cash In
Analyze Activity
Manage Security
Manage Users
...
Cashier
actor
Payment Authorization Service
alternate notation for a computer system actor
«actor» Tax Calculator
«actor» Accounting System
«actor» HR System
«actor» Sales Activity System
System Administrator
use case

内容

- 用例模型概述
- 用例图
- 用例
- 用例建模过程

用例

- 用例是对一个参与者使用系统的一项功能时所进行的交互过程的一个文字描述序列。
- 用例是描述参与者使用系统达成目标的时候一组相关的成功场景和失败场景的集合。

用例的描述

- 用例描述是用例模型的主要部分，是后续的交互图分析和类图分析必不可少的部分。
- 用例采用自然语言描述参与者与系统的交互行为，易于理解。
- 用例描述有三种不同方式：简要的、非正式的、正式的。

用例的描述格式

- 用例名称：表明用户的意图或用例的用途，如“划拨资金”
- 标识符[可选]：唯一标识符，如“UC1701”，在文档的别处用标识符来引用这个用例
- 用例描述：概述用例的几句话
- 参与者：与此用例相关的参与者列表

用例的描述格式

- 优先级： 一个有序的排列，1代表优先级最高。
- 状态[可选]：用例的状态，通常为以下几种之一：进行中、等待审查、通过审查或未通过审查
- 前置条件： 一个条件列表，如果其中包含条件，则这些条件必须在访问用例之前得到满足
- 后置条件 ： 一个条件列表，如果其中包含条件，则这些条件将在用例成功完成以后得到满足

用例的描述格式

- 基本操作流程：描述用例中各项工作都正常进行时用例的工作方式
- 可选操作流程：描述变更工作方式、出现异常或发生错误的情况下所遵循的路径
- 被泛化的用例[可选]：此用例所泛化的用例列表
- 被包含的用例[可选]：此用例所包含的用例列表
- 被扩展的用例[可选]：此用例所扩展的用例列表

用例的描述格式

- 修改历史记录[可选]：关于用例的修改时间、修改原因和修改人的详细信息
- 问题[可选]：与此用例的开发相关的问题的列表
- 决策[可选]：关键决策的列表，将这些决策记录下来以便维护时使用
- 频率[可选]：参与者访问此use case的频率。如用户每日访问一次或每月一次

用例模板应用举例

- 用例名称：处理订单
- 标识符：UC1701
- 用例描述：当一个订单初始化或者被查询的时候这个用例开始。它处理有关订单的初始化定义和授权等问题。当订单业务员完成了同一个顾客的对话的时候，它就结束了
- 参与者：订单业务员
- 优先级：1
- 状态：通过审查
- 前置条件：订单业务员登录进系统
- 后置条件：下订单；库存数目减少

用例模板应用举例

- 基本操作流程：
  - 1. 顾客来订购一个吉他，并且提供信用卡作为支付手段……
  - 2. …..
- 可选操作流程：
  - 顾客来订购一个吉他，并且使用汇票的方式……
  - 顾客来订购一个风琴，并且提供信用卡作为支付手段……
  - 顾客使用信用卡下订单，但那张信用卡是无效的……
  - 顾客来下订单，但他想要的商品没有存货……

用例模板应用举例

- 被泛化的用例：无
- 被包含的用例：登录 (UC1706)
- 被扩展的用例：无
- 修改历史记录：
  - 李四, 定义基本操作流程, 2009/9/3
  - 张三, 定义可选操作流程, 2008/10/6

描述用例时易出现的错误

- 只描述系统的行为，没有描述参与者的行为
- 只描述参与者的行为，没有描述系统的行为
- 在用例描述中就设定了对用户界面的设计的要求
- 描述过于冗长

用例描述的常见错误分析

Use case: Withdraw cash

Actor: customer

主事件流：

- 储户插入ATM卡,并输入密码
- 储户按“取款”按钮,并输入取款数目
- 储户取走现金/ATM卡/收据
- 储户离开

只描述了actor的行为

Use case: Withdraw cash

Actor: customer

主事件流：

- ATM系统获得ATM卡和密码
- 设置交易类型为“取款”
- ATM系统获得取款金额
- 输出现金、收据和ATM卡
- 系统复位

只描述了系统的行为

ATM系统“取款”用例的描述

Use case: Withdraw cash

Actor: customer

主事件流：

- 储户通过读卡机插入ATM卡
- ATM系统从卡上读取银行ID、账号、加密密码，并通过主银行系统验证银行ID和账号
- 储户输入密码，ATM系统根据加密密码对输入密码进行验证
- 储户按“取款”按钮，并输入取款数目，该数目应该为$5的倍数
- ATM系统通知主银行系统，传递账号和金额，并接收返回的确认信息和账户余额
- ATM系统输出现金、ATM卡和收据
- ATM系统记录交易到日志文件

脚本（scenario）

- 在UML中，脚本（scenario）指贯穿用例的一条单一路径，用来显示用例中的某种特殊情况
- 脚本描述了系统一次具体执行的行为路径，即一次完整的事件流
- 其它译法
  - 情景
  - 场景
  - 情节
  - 剧本

脚本的一些说明

- 一个脚本是一个用例的实例(instance)
- 脚本对于用例相当于对象对于类
- 每个用例都有一系列的脚本
  - 主要的脚本：正常情况
  - 次要的脚本：对于主要的脚本来说是例外或可以选择的

用例中的脚本举例

- “订货”用例的脚本包括：
  - 一个是订货进行顺利的脚本
  - 一个是相关货源不足的脚本
  - 一个是涉及购货者的信用卡被拒的脚本
  - ……
- 这些脚本的组合构成了“订货”用例

脚本的表示

- 一个脚本可以用具体的文字描述来表示

脚本名称	取款3000元
参与者	客户小刘

事件流

（1） 小刘将银行卡插入柜员机。
（2）柜员机要求客户输入卡密码。
（3）小刘输入卡密码，并确认密码。
（4）柜员机屏幕提示，请客户选择服务类型。
（5）小刘选择取款服务。
（6）柜员机提示：请客户输入取款数目。
（7）客户输入3000，并确认。
（8）柜员机出钱口输出30张一佰元的人民币。
（9）小刘取回30张100元面额的人民币。
（10）柜员机提示服务类型：确认、或继续，或退卡。
（11）小刘选择服务类型退卡，结束服务。

内容

- 用例模型概述
- 用例图
- 用例
- 用例建模过程

用例建模过程

1. 找出系统外部的参与者和外部系统, 确定系统边界和范围
2. 确定每一个参与者所期望的系统行为
3. 把这些系统行为命名为用例
4. 使用泛化、包含、扩展等关系处理系统行为的公共或变更部分
5. 编制每一个用例的脚本
6. 绘制用例图
7. 区分主要事件流和异常事件流, 如果需要, 可以把异常事件流处理为单独的用例
8. 细化用例图, 解决用例间重复与冲突的问题

寻找参与者和用例的方法

- 和用户交互
- 阅读业务文档
- 把自己当作参与者，与设想中的系统进行交互
- 确定用例和确定参与者不能截然分开

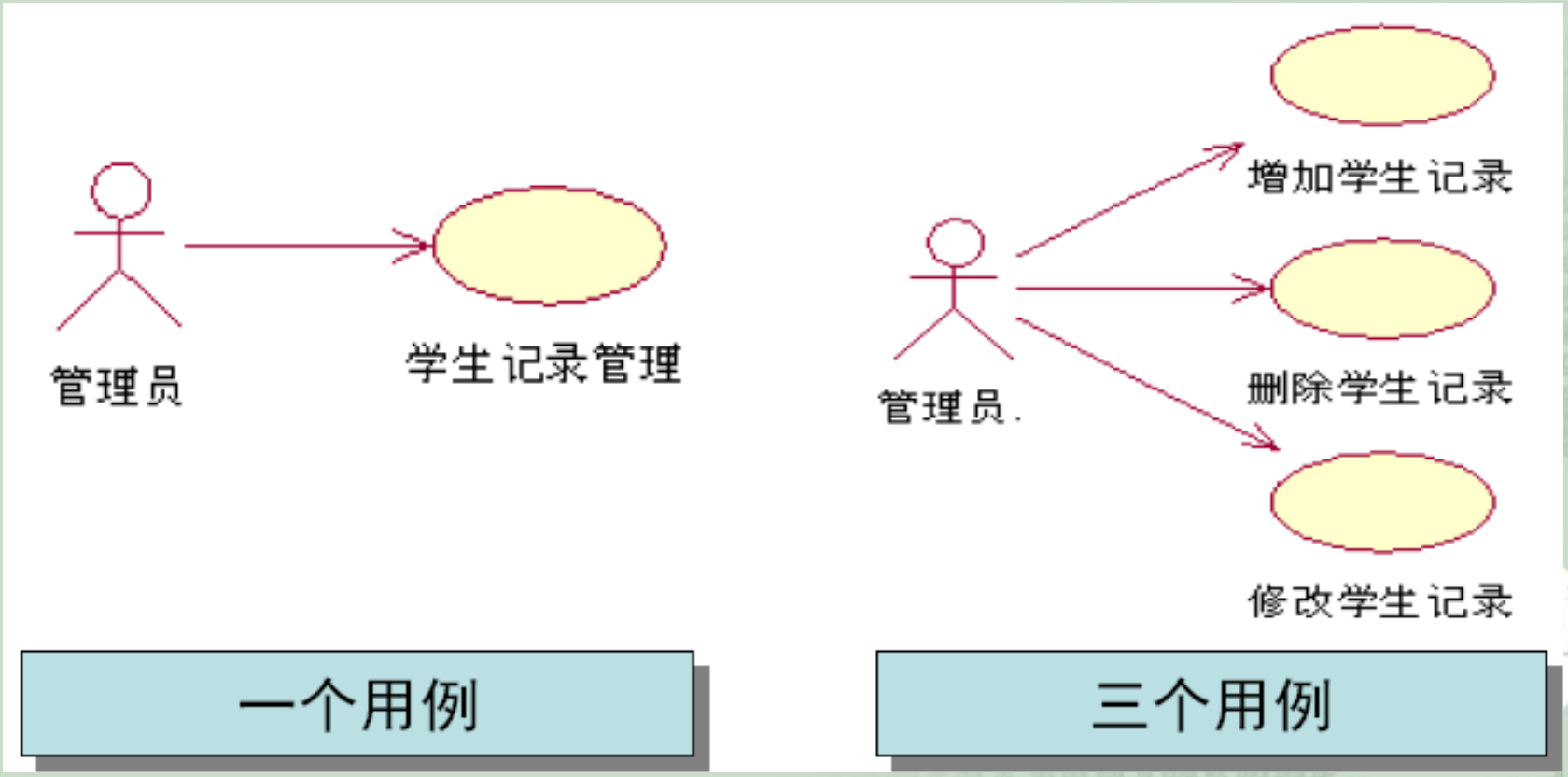
用例建模的常见问题分析

用例的粒度问题（对于一个目标系统进行用例分析后得到的用例数目有多少比较合适?）

- Ivar Jacobson的观点：对于一个10人年的项目需要20个用例
- Martin Fowler的观点：对于同样规模的项目需要100多个用例

用例建模的常见问题分析

- 用例的分解/合并（系统中相似的功能，是合并为一个用例还是分解为几个用例？）



用例建模

- 从具体的用户目标出发，用例的粒度大小一般应符合“可见的价值结果”的原则。
- 适当时候可以违背上述原则。
- 同一层次粒度大小应大致一致。

用例建模

- 用例建模的目标应该是产生简单的、易于理解的用例模型，以尽可能清晰和精确的方式捕获必须的信息。
- 用例描述是用例模型的主要组成部分。用例描述尽可能写得“充分”, 而不是形式化、完整或漂亮。
- 用例图不要过度使用高级特征而掩盖模型的真实目的。“如果怀疑，就舍弃不用”。

用例建模举例—学生选课系统

系统需求描述：

1. 为每个使用系统的人员设置权限，只有通过权限验证的人才能使用系统；
2. 学生可以使用该系统查看课程信息、选择课程、去掉所选课程；
3. 学生选课时，系统要通过学校财务管理系统核对学生是否交费，只有交费的学生才能够选课；
4. 系统录入人员负责录入选修课程和教师信息。

获取参与者

系统的参与者：

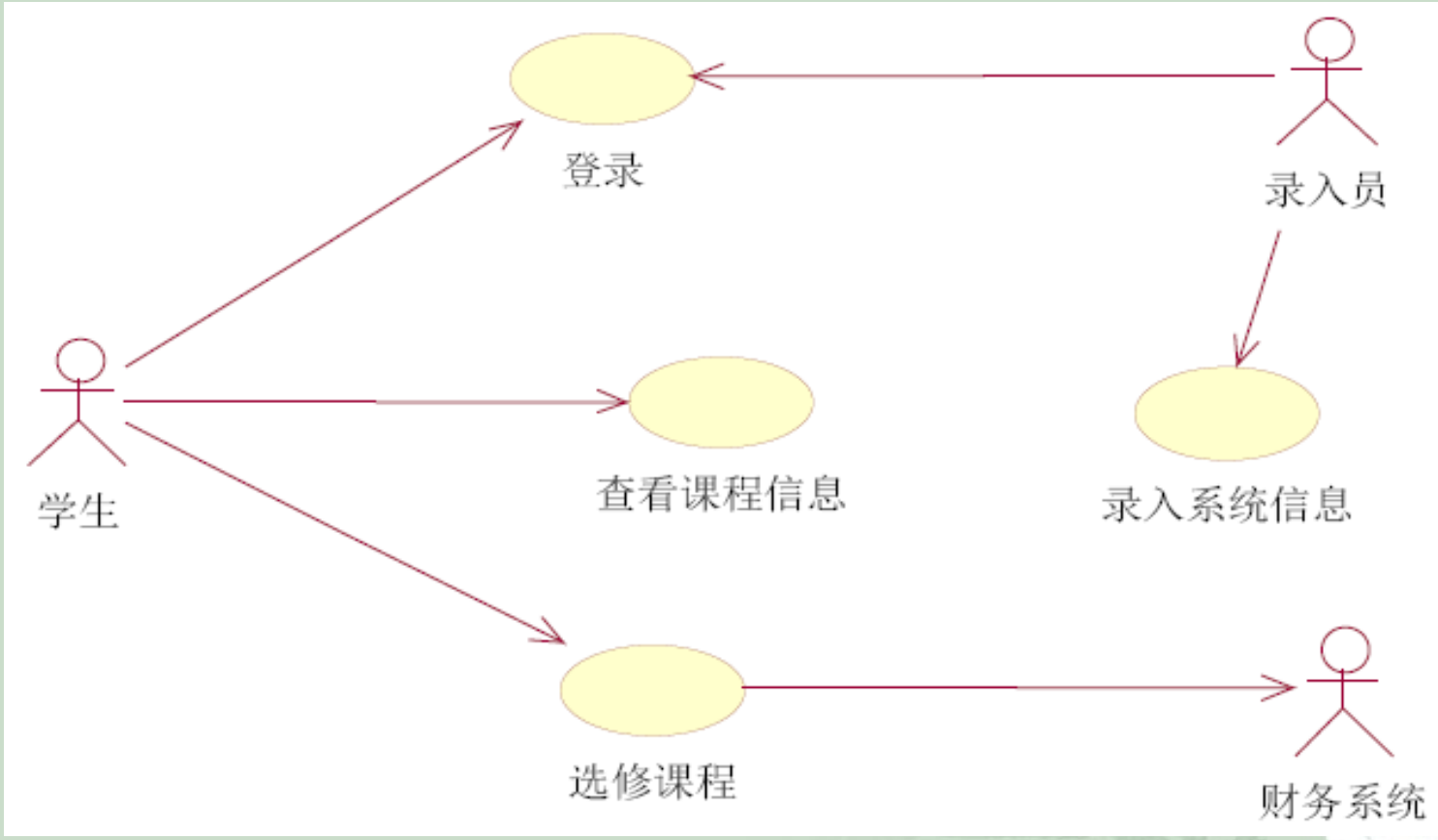
- 学生；
- 录入员；
- 财务系统；

获取用例

系统的用例：

- 查看课程信息；
- 选修课程；
- 录入系统信息；
- 登录

用例图



用例细化—选修课程

基本流：

1. 学生选择要选修的课程；
2. 系统通过财务系统检查学生是否缴费；
3. 系统更新该学生所选的课程；
4. 系统显示学生所选的课程；
5. 学生确认所选课程；
6. 系统保存学生所选课程。

用例细化—选修课程

备选流：

2. a 如果学生没有缴费，给出提示，结束

5. a 如果学生没有确认，给出提示，结束

用例建模举例—POS系统

- POS系统通常用在零售业，主要功能是记录销售情况并处理交易款支付的工作。
- 完整的POS系统由硬件（如计算机、条形码扫描器等）和软件（应用系统）共同构成。
- 在软件应用系统方面，POS系统和其它软件之间会存在交互接口，如库存控制系统、财务系统、税费计算系统等。

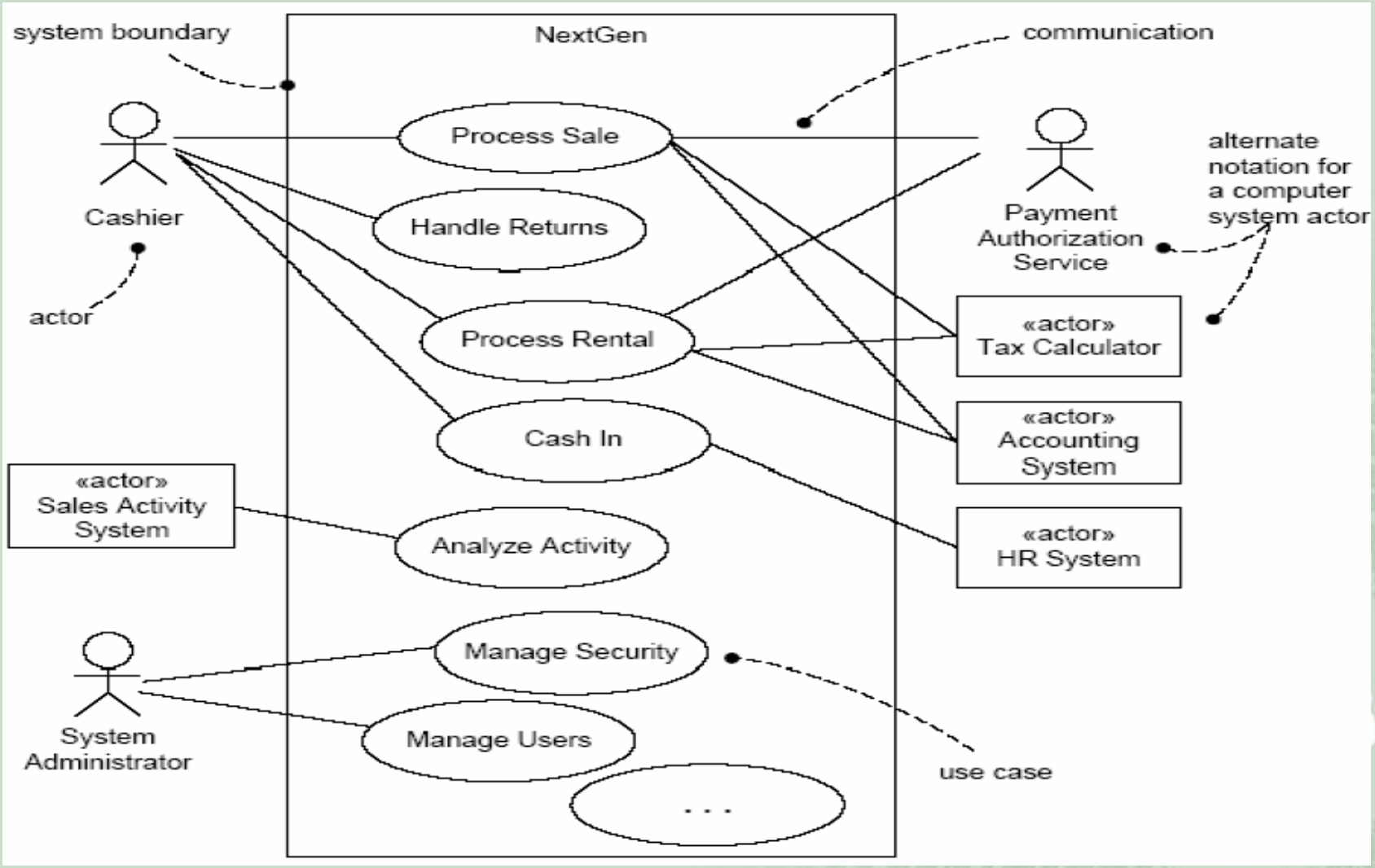
获取参与者

- 收银员
- 系统管理员
- 财务系统
- 库存系统
- 税费计算系统
- 支付授权系统
- ……

获取用例

- 处理销售
- 处理退货
- 入款
- 管理安全
- 管理用户
- 分析活动
- ……

用例图


system boundary
NextGen
communication
Process Sale
Handle Returns
Process Rental
Cash In
Analyze Activity
Manage Security
Manage Users
...
Cashier
actor
Payment Authorization Service
alternate notation for a computer system actor
«actor» Tax Calculator
«actor» Accounting System
«actor» HR System
«actor» Sales Activity System
System Administrator
use case

用例详述—处理销售

Use Case UC1: Process Sale

Primary Actor: Cashier

Stakeholders and Interests:

- Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.
- Salesperson: Wants sales commissions updated.
- Customer: Wants purchase and fast service with minimal effort. Wants proof of purchase to support returns.
- Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server components (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.
- Government Tax Agencies: Want to collect tax from every sale. May be multiple agencies, such as national, state, and county.
- Payment Authorization Service: Wants to receive digital authorization requests in the correct format and protocol. Wants to accurately account for their payables to the store.

Preconditions: Cashier is identified and authenticated.

Success Guarantee (**Postconditions**): Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.

Main Success Scenario (or **Basic Flow**):

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.

Cashier repeats steps 3-4 until indicates done.

5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods (if any).

Extensions (or Alternative Flows):

*a. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Cashier restarts System, logs in, and requests recovery of prior state.
2. System reconstructs prior state.

2a. System detects anomalies preventing recovery:

1. System signals error to the Cashier, records the error, and enters a clean state.
2. Cashier starts a new sale.

3a. Invalid identifier:

1. System signals error and rejects entry. 3b. There are multiple of same item category and tracking unique item identity not important (e.g., 5 packages of veggie-burgers):

1. Cashier can enter item category identifier and the quantity.

3-6a: Customer asks Cashier to remove an item from the purchase:

1. Cashier enters item identifier for removal from sale.
2. System displays updated running total.

3-6b. Customer tells Cashier to cancel sale:

1. Cashier cancels sale on System

3-8c. Cashier suspends the sale:

1. System records sale so that it is available for retrieval on any POS terminal. 4a. The system generated item price is not wanted (e.g., Customer complained about something and is offered a lower price):

1. Cashier enters override price.
2. System presents new price.

5a. System detects failure to communicate with external tax calculation system service:

1. System restarts the service on the POS node, and continues. 1a. System detects that the service does not restart.

1. System signals error.
2. Cashier may manually calculate and enter the tax, or cancel the sale.

5b. Customer says they are eligible for a discount (e.g., employee, preferred customer):

1. Cashier signals discount request.
2. Cashier enters Customer identification.
3. System presents discount total, based on discount rules.

5c. Customer says they have credit in their account, to apply to the sale:

1. Cashier signals credit request.
2. Cashier enters Customer identification.
3. Systems applies credit up to price=0, and reduces remaining credit.

6a. Customer says they intended to pay by cash but don't have enough cash:

1a. Customer uses an alternate payment method.
1b. Customer tells Cashier to cancel sale. Cashier cancels sale on System.

7a. Paying by cash:

1. Cashier enters the cash amount tendered.
2. System presents the balance due, and releases the cash drawer.
3. Cashier deposits cash tendered and returns balance in cash to Customer.
4. System records the cash payment.

7b. Paying by credit:

1. Customer enters their credit account information.
2. System sends payment authorization request to an external Payment Authorization Service System, and requests payment approval.
   2a. System detects failure to collaborate with external system:
      1. System signals error to Cashier.
      2. Cashier asks Customer for alternate payment.
3. System receives payment approval and signals approval to Cashier.
   3a. System receives payment denial:
      1. System signals denial to Cashier.
      2. Cashier asks Customer for alternate payment.
4. System records the credit payment, which includes the payment approval.
5. System presents credit payment signature input mechanism.
6. Cashier asks Customer for a credit payment signature. Customer enters signature.

7c. Paying by check...

7d. Paying by debit...

7e. Customer presents coupons:

1. Before handling payment, Cashier records each coupon and System reduces price as appropriate. System records the used coupons for accounting reasons.
   1a. Coupon entered is not for any purchased item:
      1. System signals error to Cashier. 9a.

There are product rebates:

1. System presents the rebate forms and rebate receipts for each item with a rebate

9b. Customer requests gift receipt (no prices visible): 1.
Cashier requests gift receipt and System presents it.

Special Requirements:

- Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.
- Credit authorization response within 30 seconds 90% of the time.
- Somehow, we want robust recovery when access to remote services such the inventory system is failing.
- Language internationalization on the text displayed.
- Pluggable business rules to be insertable at steps 3 and 7.

Technology and Data Variations List:

3a. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3b. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
7a. Credit account information entered by card reader or keyboard.
7b. Credit payment signature captured on paper receipt. But within two years, we predict many customers will want digital signature capture.

Frequency of Occurrence: Could be nearly continuous.

Open Issues:

- What are the tax law variations?
- Explore the remote service recovery issue.
- What customization is needed for different businesses?
- Must a cashier take their cash drawer when they log out?
- Can the customer directly use the card reader, or does the cashier have to do it?