UML建模基础
—用例图

东软IT人才实训中心

Neusoft

第二章： 用例、用例图

目标:
本章旨在向学员简要介绍用例模型及用例的关系,通过本课的学习,学员应该掌握如下知识:
1）能区分角色和用例
2）了解用例的关系

学时：1学时

教学方法：讲授ppt＋上机练习＋案例分析

2.1 用例模型

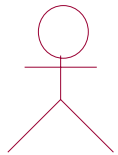
- 为什么要创建用例模型
  - 用例模型允许顾客和系统开发者之间用一种用户可以理解的语言交流系统要做什么
  - 你可以认为用例模型是顾客和开发者之间的可视化契约
- 什么是用例模型
  - 在Use-case View中创建
  - 用例模型代表了从最终用户角度看的系统的功能和环境
  - 是顾客和开发者之间的契约
  - 对于分析、设计和测试活动都是至关重要的
  - 包括用例图、用例规约和补充规约，也可以包括活动图

2.2 用例图（Use case diagram)

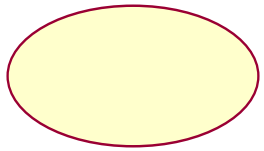
- 用例图表示了用例和角色以及用例和用例之间的关系
  - 可视化的表示出了客户希望系统做什么
  - 代表一些大的完整的功能
  - 表示系统完成的有明确结果的对角色有价值的一系列动作
- 可以模型化
  - 所有的角色和用例（global view）
  - 某个选定角色的所有用例
  - 一个用例以及它所有的关系
  - 一个迭代的所有的用例

2.3 用例图的元素

- 角色
- 用例
- 关系



Student



Registration

2.4 角色（Actor）

- 定义：系统外的与系统进行交互的人、硬件设备、另一个系统。
- 种类
  - 人
  - 外部系统
  - 外部设备或Timer
- 识别Actor要依据Actor的定义，可以这样查找：
  - 有哪些用户使用系统？
  - 系统会用到哪些外部的系统或设备？
  - 有什么外部系统或设备会用到要开发的系统？
  - 有没有定时触发的行为？

2.4 角色（续）

- 如何判断一个事物是不是actor
  - 首先它必须与系统有交互，如果与系统没有交互则不是角色
  - 其次它必须是系统外的，如果它是我们将要开发的系统或是系统的一部分则不是角色
- 其它
  - 用户如果通过标准的输入和输出设备与系统交互则用户是Actor
  - 用户如果通过特殊的设备与系统交互，则设备是Actor

2.5 用例（Use Case）

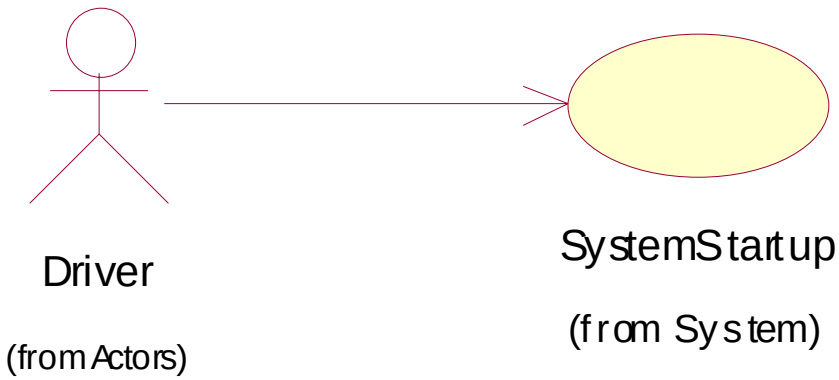
- 定义：是actor与系统的一系列交互
- 特点：
  - 完成actor的某个目的（不是功能），一般会给actor一个有价值的结果
  - 起始于actor的输入
  - 其中，系统是一个黑盒
  - 用于描述系统行为，但不描述如何实现
- 识别用例的依据就是用例的定义和特点
- 识别用例时需要注意
  - 用例的粒度不要太大也不要太小
  - 用例描述的是系统做什么，初始识别用例的时候不要过多考虑系统的实现，即把系统作为黑盒
  - 外部系统或设备的行为不是要开发的系统行为，不要识别出来用例

2.5 用例（续）

- 有些用例不代表系统的主要功能，因而通常会被大家忽视，这些用例可能属于以下类型：
  - 系统启动和停止
  - 系统的维护。例如，添加新用户和建立用户简档
  - 维护在系统中存储的数据。例如，所构建的系统和遗留系统平行工作，所以数据需要在两个系统之间达到同步
  - 修改系统行为所需的功能。例如创建新报告的功能，它不仅可以创建硬代码，还可以对系统中存储的数据创建一组特定报告

2.6 Actor和Use case的关系

- Actor与use case之间的关系是association关系，含义是“触发”，千万不要理解成数据流



2.7 Rose操作—加入模型元素

- 从browser窗口中加入
  - 选择要加入模型元素所在的package，单击鼠标右键，从弹出菜单中选择new-->模型元素种类（如class，package，use case diagram等），此时相应的包下面就会加入一个新的元素，你可以为它命名
- 从Diagram的toolbar中直接加入
  - 从toolbar中选择要加入的元素类型，单击diagram窗口的某个位置，新的元素就会显示到diagram窗口中，此时你可以为新元素命名。同时browser中也出现新的元素（新的模型元素会加入到相应的diagram所在的包中）。

2.8 Rose操作—更改模型元素

- 双击browser或diagram中的元素（或者单击鼠标右键，从弹出菜单中选择open specification）就会打开新建元素的specification，在specification对话框中，你可以更改名字以及做其他的设置
- 注意：diagram没有specification
- 注意：双击diagram中的package，不会打开它的specification，而是进入package下的某个类图

2.9 Rose操作—删除模型元素

- Delete from model：模型元素从模型中删除（也从它参与的所有图中删除）
  - 从browser中选择要删除模型元素，单击鼠标右键，从弹出菜单中选择delete，或者
  - 从diagram中选择要删除模型元素，从菜单中选择edit-->delete from model（快捷键ctrl+D）
- Delete from diagram
  - 从diagram中中选择要删除模型元素，从菜单中选择edit-->delete （快捷键Del）

2.10 详细用例模型

- Actor之间的关系——泛化（generalization）

- Use Case之间的关系
  - 泛化（generalization）：不常用
  - 扩展（extend）
  - 包含（include）

2.11 用例之间的扩展关系

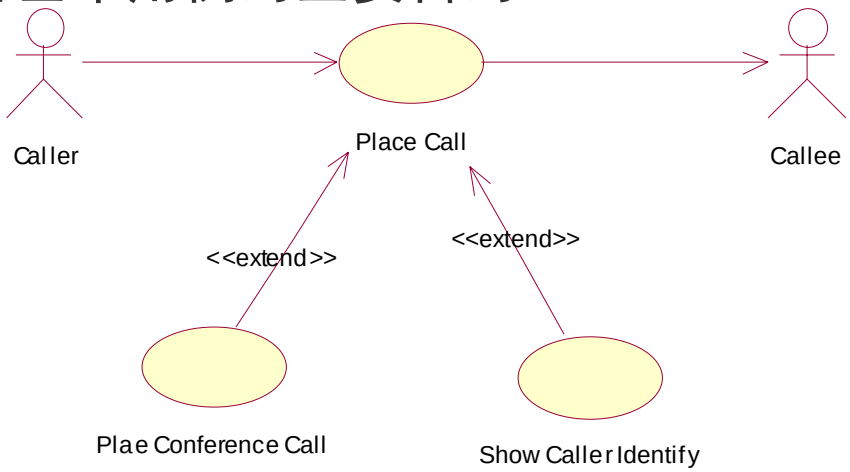
- 扩展关系的双方分别叫做基本用例（Base use case）和扩展用例（extension use case）
- 扩展用例用来模型化基本用例中
  - 有条件的部分，只在某些环境下执行
  - 复杂的或可选的路径。
- 扩展关系用stereotype是“extend” 的association关系来表示

2.11 用例之间的扩展关系 （续）

- 扩展用例和基本用例的关系
  - 基本用例自身应是完整的，即基本用例在不必引用任何扩展用例的情况下，应该是可理解且有意义的
  - 基本用例可以不依赖于扩展用例而单独的运行
  - 扩展用例只有在基本用例中的某种条件满足时才能执行，如果没有基本用例，扩展用例不能运行
  - 扩展用例可以扩展多个基本用例
- 扩展点
  - 定义在基本用例的哪些位置插入扩展用例
  - 包括一个名称和对用例事件流中一个或多个位置的引用
  - 一个扩展点可以引用基本用例内的两个行为步骤之间的单个位置，也可以引用一组不连续的位置

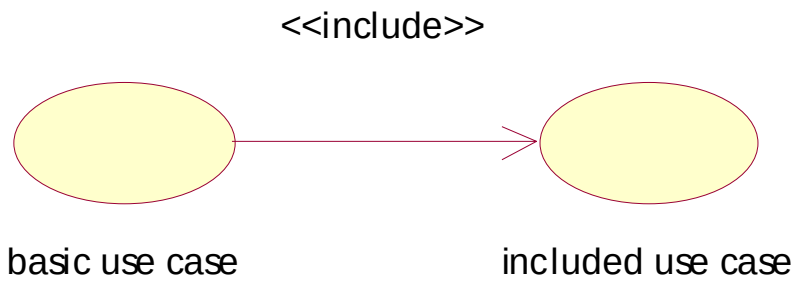
2.12 扩展关系的实例

- 在电话系统中，为用户提供的主要服务通过用例“打电话”来表示。可选服务的示例包括
  - 能让第三方加入通话（召开电话会议）
  - 允许接收方看到呼叫方的身份（显示呼叫方身份）
- 我们可以将这些可选服务所需的行为表示为基本用例“打电话”的扩展用例，由于“打电话”本身就具有意义，您无需阅读扩展用例的说明就可理解基本用例的主要目的



2.13 用例之间的包含关系

- 包含关系的双方分别叫做基本用例（Basic use case）（或具体用例, concrete use case）和包含用例(included use case)（或抽象用例, abstract use case）
- 包含用例用来模型化基本用例中
  - 多个用例都包含的路径
  - 复杂的路径
  - 包含用例要能生成一个有意义的结果
- 包含关系用stereotype是“include” 的association关系来表示



2.13 用例之间的包含关系（续）

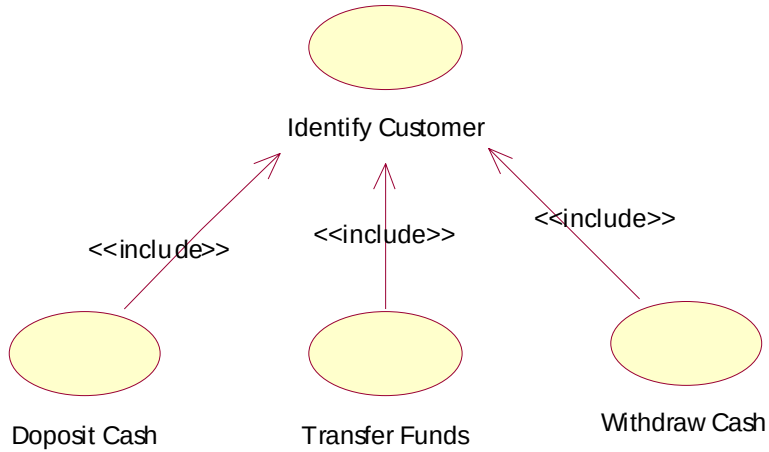
- 包含用例和基本用例
  - 包含用例不能单独执行，必须与包括（也就是执行）它的基本用例一起执行
  - 包含用例没有特定的actor，它的actor实际上包括它的基本用例的actor
  - 包含用例可以被几个其他的用例复用
  - 基本用例的基本流执行时，包含用例一定执行

2.14 扩展关系和包含关系

- 共同点
  - 扩展用例和包含用例都是基本用例的一部分
  - 基本用例不执行，扩展用例和包含用例都不会执行
  - 扩展用例可以扩展多个基本用例；包含用例可以被多个基本用例包含
- 区别
  - 扩展关系中基本用例的基本流执行时，扩展用例不一定执行，即扩展用例只有在基本用例满足某种条件的时候才会执行
  - 包含关系中基本用例的基本流执行时，包含用例一定会执行

2.15 包含关系的实例

- 在 ATM 系统中，用例 Withdraw Cash、Deposit Cash和 Transfer Funds都需要包含系统识别客户的过程。可以将此行为抽取到一个名为 Identify Customer的新包含用例中
- 从基本用例的角度来看，识别客户方法是读取银行磁卡还是执行视网膜扫描并不重要。它们仅依赖于 Identify Customer 的结果，即客户的身份。
- 从 Identify Customer 用例的角度来看，基本用例如何使用客户身份或者在执行包含用例之前基本用例中发生了什么并不重要：识别方法都会完全相同



2.16 术语

缩语、术语	英文全称	解 释
Actor	**Actor**	角色
Use Case	**Use Case**	用例
Use case diagram	Use case diagram	用例图

Neusoft

Beyond Technology