

第 3 章

用例建模

在系统开发的分析阶段,用户对系统的使用方式直接决定了系统的设计方式与构建方式。所以从用户观点出发,对帮助分析人员理解用户需求,建立可用、有用的系统是十分关键的。从用户的观点出发对系统建立模型是用例建模要完成的任务,因此用例建模通常也称为需求建模。本章在本书知识体系中的位置如图 3.1 所示。

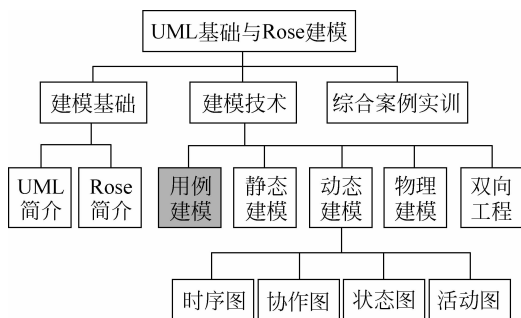


图 3.1 本章在本书知识体系中的位置

在 UML 中,一个用例模型由若干个用例图(Use Case Diagram)描述。用例图是显示一组参与者、用例以及它们之间的关系的图。

3.1 参 与 者

相关知识

1. 系统边界

系统边界(System Boundary)指一个软件系统能够处理的整个问题空间的范围。一个软件系统不可能处理所有问题,开发人员必须给系统定义问题空间的范围。哪些是这个软件可以处理的,哪些则是这个软件不能处理的,也就是项目管理中所说的项目范围。

在 UML 中,系统边界用方框表示,或者省略不做表示。

2. 参与者

参与者(Actor)指的是存在于系统之外,透过系统边界与系统进行有意义交互的任何

事物。参与者可以是一个人、一个其他的系统或一部机器,甚至可以是时间,如图 3.2 所示。例如在“自动售货系统”中,系统有售货、供货、提取销售款等功能,其中启动售货功能的是人,那么人就是参与者;又如“图书管理系统”可能需要和其他应用系统发生联系,比方说可能通过“学生管理系统”验证读者是否为在校学生,那么这里的“学生管理系统”就是一个参与者,只不过该参与者不是具体的一个人,而是另外的一个系统;与一个系统进行交互的人或其他

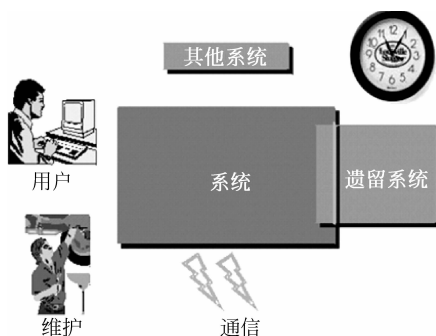


图 3.2 参与者举例

的系统可以是参与者,与系统进行通信的硬件设备也可以是参与者,例如在“自动售货系统”中,顾客购买货品时,最终是货品分配器将货品传送至出货口以便用户提取,此时货品分配器作为硬件设备也就成为该系统的参与者之一;再如在“图书管理系统”中如果读者到期没有归还图书,则读者进入系统时会有“未还书提示”功能,此处时间也就成为参与者,也就是说当经过一定时间后系统中的“未还书提示”事件就会发生。

从参与者在系统中的地位来看,可以将其分成两类,即主要参与者(Primary Actor)和次要参与者(Secondary Actor)。主要参与者指的是执行系统主要功能的参与者,例如在“图书管理系统”中主要参与者是进行借阅管理的图书管理员;次要参与者指的是执行系统次要功能的参与者,次要功能一般指系统维护功能(如管理数据库、备份和通信等),例如在“图书管理系统”中,能够检索该系统中一些基本统计数据的系统管理员属于次要参与者。将参与者分类的主要目的是保证把系统所有功能表示出来,而主要功能是系统最关心的部分。

从参与者对用例的作用来看,可以将其分为主动参与者和被动参与者。主动参与者可以初始化用例、启动用例,而被动参与者则不能,它需要使用用例结果或为用例执行提供数据,被动参与者可参与一个或多个用例,在某个时刻与用例通信。

在 UML 中,参与者表示为一个小人图形的(Stick Man 符号),在小人图形的下方书写参与者的名字,如图 3.3 所示;也可以用类符号(类的具体内容详见第 4 章)来表示参与者,如图 3.4 所示。



图 3.3 参与者

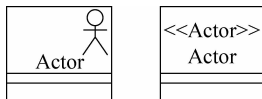


图 3.4 用类符号表示参与者

3. 识别参与者

怎样确定系统的参与者呢? 开发人员可以从如下几个方面来考虑。

从交互识别:

- (1) 谁使用系统的主要功能?
- (2) 谁改变系统的数据?

- (3) 谁从系统中获得信息?
- (4) 谁需要系统的支持以完成日常工作任务?
- (5) 谁(或什么)对系统运行产生的结果感兴趣?

从维护、管理识别:

谁负责维护、管理并保持系统正常运行?

从设备或外部条件识别:

- (1) 系统需要应付(处理)哪些硬件设备?
- (2) 系统需要和哪些外部系统交互?
- (3) 时间、气温等条件是否对系统产生影响?

例如,在“基于 Web 的零件销售系统”中,顾客可以通过 Internet 进行购买。该系统要求顾客先预付一定金额存入内部账户中成为会员,然后才能购买零件。顾客可以根据自己所知道的零件的形状、大小、零件编号等条件,搜索出所需要的零件。结账时使用内部账户支付。系统根据会员提供的送货地址和订购数量,从库存中搜索出离送货地址最近的供应商,并通知供应商发货。另外,内部工作人员不定期地根据供应商方面的价格变动,对某些零件的销售价格进行更新。每个星期各个供应商会把记录自己最新库存情况的 Excel 文件寄来,系统根据这些文件更新库存信息。因简化的需要,折扣、延迟交货等因素略去不考虑。

以该系统为例,针对上述问题进行回答便可以确定系统的参与者。

- (1) 潜在会员和会员使用系统的主要功能。
- (2) 会员、货管员和经理改变系统的数据。
- (3) 潜在会员、会员、经理和货管员从系统中获得信息。
- (4) 经理和货管员需要系统的支持以完成日常工作任务。
- (5) 会员和经理对系统运行产生的结果感兴趣。
- (6) 系统管理员负责维护、管理并保持系统正常运行。
- (7) 系统无须应付(处理)特殊硬件设备。
- (8) 系统可能与供应商的系统交互。
- (9) 忽略时间、气温等条件对系统产生的影响。

综上回答,确定出“基于 Web 的零件销售系统”的参与者有:潜在会员、会员、经理、货管员、系统管理员和供应商系统。



能力目标

能够识别“图书管理系统”中的参与者,并且在 Rational Rose 中绘制图示。



具体要求

- (1) 识别“图书管理系统”中的参与者。
- (2) 在 Rational Rose 中绘制“图书管理系统”的参与者。



实训任务

1. 识别“图书管理系统”中的参与者

遵循前面叙述的识别参与者的方法,可以分析出“图书管理系统”中的参与者有: Administrator(系统管理员)、Librarian(图书管理员)和 Reader(读者)。

- (1) Administrator: 通过使用系统进行用户管理。
- (2) Librarian: 通过使用系统进行读者管理、图书管理、借阅管理等。
- (3) Reader: 通过使用系统进行读者信息查询、预订图书、取消预订等。

2. 在 Rational Rose 中绘制“图书管理系统”的参与者

(1) 创建和保存工程

启动 Rational Rose 后,在菜单栏中选择 File→New 菜单项,创建一个模型,选择 File→Save 或 Save As 保存工程为 Library,如图 3.5 所示。

(2) 创建用例图

在 Rational Rose 左侧视图区域树形列表中选择 Use Case View,右击,在弹出的快捷菜单中选择 New→Use Case Diagram 命令,新建一个用例图,如图 3.6 所示。

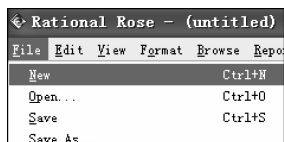


图 3.5 新建工程

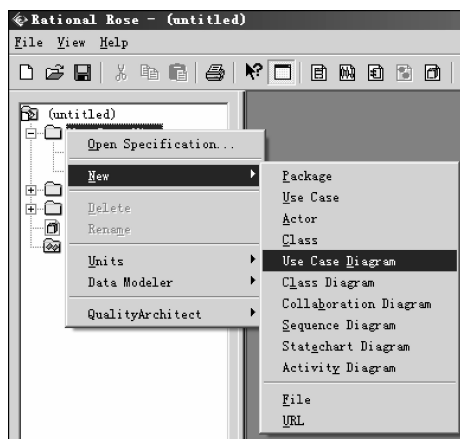


图 3.6 创建用例图

新建用例图默认名称为 NewDiagram,可以更改其名称,更改方法是选择 NewDiagram,右击,在弹出的快捷菜单中选择 Rename 命令,如图 3.7 所示,然后输入用例图的新名称即可,在此可将用例图名称改为 Library UseCase。双击该用例图,在 Rational Rose 窗口内右侧空白处出现相应的编辑区,在编辑区中可进行后续操作。

(3) 新建参与者

在编辑区工具栏中单击 Stick Man 按钮,也就是 Actor,如图 3.8 所示,然后将鼠标指针停放在编辑区任意位置,鼠标指针会变成十字形状,在需要的位置再次单击即可在编辑区中绘制出参与者的图示。新建的参与者默认名称为 NewClass,可将其名称根据具体情况进行修改。简便的修改方法是直接在 NewClass 处输入参与者的新名称;稍复杂的修

改方法是双击该参与者打开参与者属性对话框,或者右击该参与者,在弹出的快捷菜单中选择 Open Specification 命令也可以打开参与者属性对话框,如图 3.9 所示,在此进行参与者名称的修改,同时还可以进行其他方面更为详细的设置。

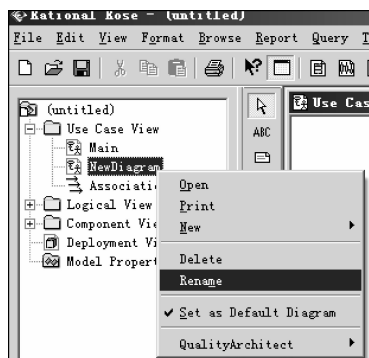


图 3.7 重命名用例图

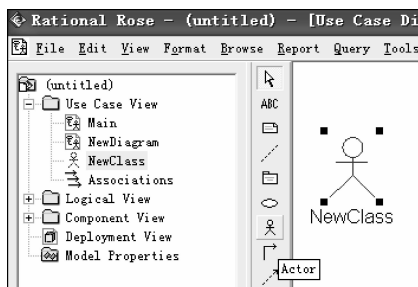


图 3.8 新建参与者

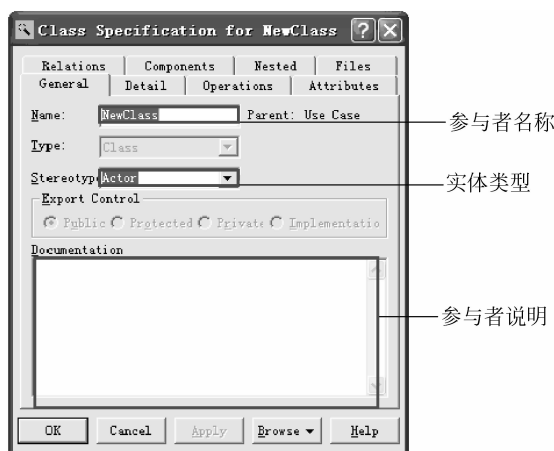


图 3.9 参与者属性对话框

按照以上方法,可以绘制出图书管理系统的全部参与者,即 Administrator、Librarian 和 Reader,如图 3.10 所示。

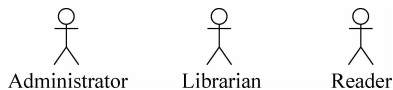


图 3.10 图书管理系统的参与者



能力测试

- (1) 什么是系统边界?
- (2) 什么是参与者? 如何用图示描述参与者?
- (3) 识别参与者应遵循怎样的步骤?
- (4) 在 UML 中,参与者可以是()。

A. 人员 B. 外部系统 C. 硬件设备 D. 时间

(5) 参考以上图书管理系统,捕获短信平台系统中的参与者,并使用 Rational Rose 工具实现。需求描述如下。

用户如果预订了天气预报,那么系统每天定时给用户发送天气信息;如果当天气温高于 34℃,系统还要提醒用户注意防暑。



知识扩展

1. 自定义绘图工具栏

绘图工具栏中给出了常用的绘图工具,如果用户有更丰富的需求,可根据实际情况自定义该工具栏。方法是右击绘图工具栏的任意位置,在弹出的快捷菜单中选择 Customize... 命令,如图 3.11 所示。

然后在弹出的“自定义工具栏”对话框中的“可用工具栏按钮”栏中,选取需要添加的工具,单击“添加”按钮,确认无误后关闭该对话框即可,如图 3.12 所示。

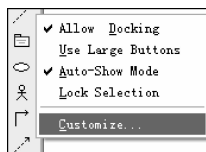


图 3.11 自定义绘图工具栏



图 3.12 添加自定义工具

2. 绘制多个参与者的快捷方法

如果系统用例图中有多个参与者,按照前面介绍的方法绘制相关图示就需要做很多重复劳动,显得相当麻烦。其实可以选择更快捷的方法来完成多个参与者的绘制,首先单击绘图工具栏中的锁定选择按钮 , 然后单击参与者按钮,此时鼠标指针会变成十字形状,接着在编辑区中任意位置多次单击,即可绘制出多个参与者图示,再次单击锁定选择按钮即可取消锁定,如图 3.13 所示。此方法同样适用于其他图示。

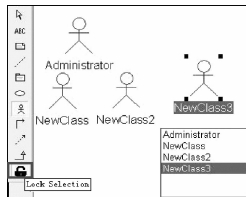


图 3.13 锁定选择

3. 删除用例模型中的参与者

使用 Rational Rose 绘制系统用例模型的参与者时,由于操作不当或需求发生变化可能需要删除已绘制出的参与者图示,删除方法有如下几种。

(1) 在编辑区中选择待删除的参与者,然后单击菜单 Edit→Delete from Model 即可删除指定参与者,如图 3.14 所示。另外,在编辑区中选择待删除的参与者后,直接按 Ctrl+D 键也可删除指定参与者。

(2) 在左侧浏览器窗口的树形结构中选择待删除对象,右击,在弹出的快捷菜单中选择 Delete 命令,即可删除指定参与者,如图 3.15 所示。另外,在浏览器窗口的树形结构中选择待删除对象后,直接按 Delete 键也可删除指定参与者。

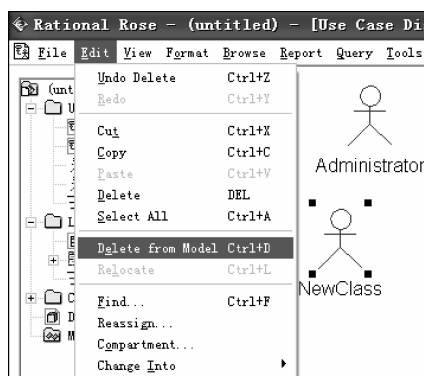


图 3.14 彻底删除 NewClass 参与者

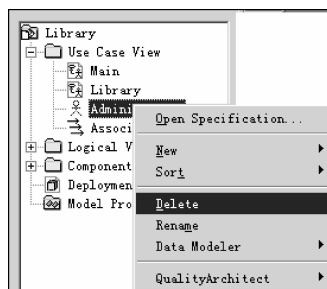


图 3.15 彻底删除 Administrator 参与者

(3) 在编辑区中选择待删除的参与者,右击,在弹出的快捷菜单中选择 Edit→Delete 命令,可以将指定参与者从编辑区中删除,但该参与者在用例模型中仍然存在,即在浏览器窗口树形结构中该参与者仍然显示,如图 3.16 所示。另外,在编辑区中选择待删除的参与者后,直接按 Delete 键也可达到相同效果。

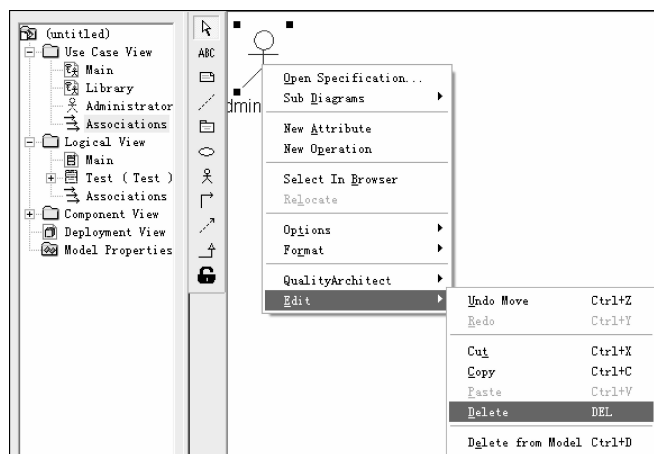


图 3.16 在编辑区中删除参与者

3.2 用 例

相关知识

1. 用例

用例(UseCase)是系统执行的一系列动作,这些动作生成特定参与者可观测的、有价值的结果值。一个用例定义一组用例实例。

用例是参与者要求系统提供的服务,是参与者使用系统期望达到的目标,是从用户角度描述的系统行为。通俗地说,用例侧重于目标,而不是内部处理。用例没有表示任何关

于系统内部的东西,只是表示系统将达到什么样的目标及由哪些参与者操作、负责。

用例具有以下特征。

(1) 每个用例都有相应的信息输入,而输入信息一般由参与者提供。

(2) 每个用例都有相应的信息输出,而输出信息一般被参与者接收。

(3) 每个用例都是对一项系统功能的完整表述,对应具体的用户目标。

在 UML 中,用例被表示为一个椭圆图形,在椭圆的下方或内部书写用例名称,如图 3.17 所示。



图 3.17 用例

2. 识别用例

对于用户来讲,用例可以帮助他们明确未来怎样使用系统;而对于开发者来讲,就需要在不关注细节的情况下,快速收集系统需求,识别出系统用例。识别用例是系统开发中的重要组成部分,一个新的系统在研发前首先要进行系统的功能分析,弄清楚业务流程,这样才能更好地为系统服务,借助用例进行描述便可以解决上述问题。

怎样确定系统的用例呢?识别用例最好的方法是从参与者着手,考虑每个参与者是怎样使用系统的。应用这种策略可能还会找出一些新的参与者,这对完善整个系统用例模型大有益处。用例建模的过程就是反复迭代和逐步精化的过程。在此过程中,通过回答如下几个问题可以帮助识别用例。

- (1) 参与者要为系统提供哪些功能和信息?
- (2) 参与者需要系统提供哪些功能和信息?
- (3) 系统在哪些方面存在问题,在哪些方面亟待完善?

进而,确定用例的一般步骤如下。

- (1) 确定哪些参与者为系统提供输入信息。
- (2) 确定哪些参与者需要从系统中获取输出信息。
- (3) 确定输入信息与输出信息之间的映射关系,即用例。

例如,在“仓库管理系统”中,通过与系统用户沟通,分析人员可以识别出系统主要参与者有操作员、管理员、供应商、商品领料人和商品退料人。针对以上参与者,识别出系统用例有仓库进货、仓库退货、仓库领料、仓库退料、商品调拨、仓库盘点、库存查询、业务分析、仓库历史记录查询、供应商信息维护、仓库信息维护、用户管理等。

3. 用例阐述文档

用例阐述文档是针对每个用例,描述该用例各个场景(Scenario)的文档。所谓场景是指参与者和系统之间的一系列特定的活动和交互。每个用例是一组场景的集合,而每个场景又是一组步骤序列的集合。具体应用时,可采用如下用例阐述的模板。

用例编号:

用例名称:

涉及的参与者:

用例概述:

前置条件:

后置条件:

基本事件流:

1.

2.

备选流:

1a.

1a1.

1a2.

2a.

2a1.

2a2.

补充说明:

其中前置条件约束用例开始前系统的状态,后置条件约束用例执行后系统的状态。

例如,在“图书管理系统”的 Borrow Book(借书)这个用例中,包含着如下几个相关的场景。

Scenario-1: 顺利地借到书。

Scenario-2: 该种书刊不存在。

Scenario-3: 物理书刊都已借出。

Scenario-4: 没有该读者信息。

为描述该用例的各个场景,可采用如下用例阐述文档加以说明。

用例编号: 001

用例名称: Borrow Book

涉及的参与者: Librarian(图书管理员)

用例概述: 读者凭读者证借阅图书。

前置条件: 图书馆正常开放时间。

后置条件: 如果读者借阅成功,则该读者可借阅书刊数量减少,馆藏减少;如果读者借阅失败,该读者可借阅书刊数量不变。

基本事件流(Flow of Events):

1. 读者进入图书馆。
2. 读者查找图书。
3. 读者将读者证提交给图书管理员,并提交借阅图书请求。
4. 图书管理员检查读者可借阅书刊数量和馆藏数量,如果满足条件则借出图书。
5. 图书借给读者,读者可借阅书刊数量减少,馆藏减少。
6. 系统显示读者当前的借阅信息。

备选流:

- 1a. 没有该借阅者信息。
 - 1a1. 系统提示借阅者信息不存在。

- 2a. 查询图书信息不存在。
 - 2a1. 系统提示该图书不存在。
- 3a. 馆藏书刊均已借出。
 - 3a1. 系统提示无馆藏。



能力目标

能够识别“图书管理系统”中的用例,并在 Rational Rose 中绘制图示。



具体要求

- (1) 识别“图书管理系统”的用例。
- (2) 在 Rational Rose 中绘制“图书管理系统”的用例。



实训任务

1. 识别“图书管理系统”的用例

根据 3.1 节中分析出的系统主要参与者(系统管理员、图书管理员、读者),可以分析出“图书管理系统”中主要用例包括 Manage User(用户管理)、Manage Book(图书管理)、Manage Reader(读者管理)、Borrow-Lend(借阅管理)等,详细说明如下。

- (1) Manage User: 完成系统用户的增加、删除、修改、查询等功能。
- (2) Manage Book: 完成基本信息设置(图书类型设置、借阅种类设置)和图书信息管理(图书信息设置、图书信息查询)功能。
- (3) Manage Reader: 完成读者办证、读者信息查询、读者证挂失功能。
- (4) Borrow-Lend: 完成借书、还书、续借、超期罚款、图书预订、取消预订、图书挂失等功能。

综合对“图书管理系统”中参与者和相关系统功能的分析,得到该系统的全部用例见表 3.1。

表 3.1 “图书管理系统”的用例

编号	参与者	用例名称	用例说明
1	Administrator (系统管理员)	Add User	增加系统用户
2		Delete User	删除系统用户
3		Update User	修改系统用户
4		Query User	查询系统用户