

一个用例层次模型及工具开发

A Hierarchical Use Case Model and Tool Development

刘洪辉 (湖南理工学院 计算机与信息工程系 湖南 岳阳 414006)

摘要:用例描述通常采用单一层次结构,不能充分实现用例描述的复用。文章在深入研究已有的用例描述方法的基础上,提出了一种用例描述层次模型。该模型将用例分为服务层、结构层和事件层三个抽象层次来描述,容易保证用例间的一致性,且可以灵活实现系统间、用例间和用例内部用例描述的重用。文章最后还讨论了有关工具的开发。

关键词:用例描述 层次模型 服务 结构 事件 重用

1 引言

自 I. Jacobson 提出用例及其有关概念以来,用例被广泛地引入到各种软件开发方法中^[1],在开发的早期阶段描述系统功能需求^[2]。文献^[3]给出了若干用例描述方式。自然语言是大多数使用者采用的描述方式。但自然语言有它的弱点,如表达不精确、易产生不一致性,冗余大,修改困难等。模板是用例描述的另一常用方式。模板描述一般是在一个抽象层次上进行,对于复杂用例的描述篇幅往往较长。同时,模板描述方式也不便于直观表现用例间的关系,仍需要借助图形来进行描述。图形描述直观、简洁,但现有的图形描述不能完全描述用例,需要借助自然语言。UML 对用例的描述没有明确规定,除采用自然语言或模板外,也可使用活动图和顺序图来描述^[4]。用 UML 的活动图和顺序图来描述用例的主要困难之一是如何保持图形内容之间的一致性。文献^[2]给出了一种层次化的图形用例描述,该层次图形描述的主要问题是难以描述用例间的关系。

用例描述可能随时间的推移变得越来越复杂^[3]。在用例描述的早期,用例的描述要便于管理者和一般用户理解,这要求用例描述要比较直观、粗略;在用例描述的后期,要提供精确的信息便于软件开发者使用,对用例描述要求比较形式化。同时,用例描述的内容和用例间关系的描述不仅随时间的推移不断增多,而且也会随时间的推移不断变化。因此,一个好的用例描述方式应该具有如下特点:

(1) 层次化。用例描述的层次化符合用例描述由粗到细的过程,也便于不同的用户使用。

(2) 模块化。用例描述的内容往往较多,修改也往往比较频繁。用例描述模块化有利于用例描述的重用,也便于将用例修改局部化。

(3) 规范化。用例描述的规范化有利于用例描述方式和内容的统一,也便于用例间关系使用的统一。

(4) 图形化。图形化的描述直观、简洁,易于一般的用户理解和掌握。在此,作者提出一种基于重用的可视化用例描述层次模型,规范用例间关系的描述,从而简化用例描述的复杂性。

2 用例描述的三层模型

2.1 服务层

服务层主要描述系统能给用户哪些服务,有哪些主要参与者使用这些服务。系统提供的服务由用例来描述。系统通过该层用例向主要参与者提供业务价值。最好的用例是为系统的业务提供最大价值的用例。为业务提供很少价值或无价值的用例不是真正的用例,即无用用例(*useless case*)。功能相关的用例可以组成服务包。主要参与者通过使用用例来从系统中得到所需的价值。服务层只描述用例与主要参与者之间的关系。图 1 描述了服务层各元素之间的关系。

显然,服务层对于一般的用户和管理人员来说非常直观。该层元素非常简单,而且描述符号也不复杂(如表 1 所示),一般用户易于掌握。同时,该层元素可

以作为用例描述的一个索引,从而便于不同兴趣的用户通过该层元素找到相应的用例细节描述,也便于不同系统用例描述的重用。

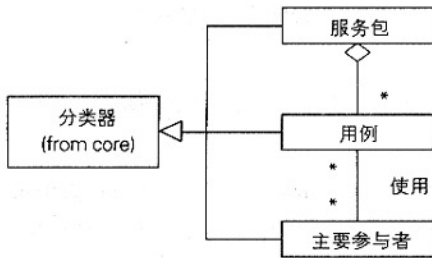


图 1 服务层各元素之间的关系

表 1 服务层元素的图形表示

| 元素名称 | 图形符号 |
|------|------|
| 服务包 | |
| 用例 | |
| 参与者 | |
| 使用 | |

用例间的关系有三种:包含(include)、扩展(extend)和泛化(generalization)。在描述单个用例时,用例的前提条件表示用例前必须满足的系统状态,而用例的成功保障则表明用例成功执行后系统应该所处的状态。用例的前提条件和成功保障对确定用例的边界很有帮助。图 2 描述了结构层各元素之间的关系。表 2 给出了结构层各元素的图形表示。

在用例描述中,最难处理的是用例之间的三个关系。包含关系一般用于提取两个或多个用例共有的部分。描述时,被包含用例没有包含用例的信息。扩展关系一般用于在不修改或很少修改已有用例的情况下扩展已有用例的功能。描述时,扩展用例必须知道基用例的扩展点。扩展用例一般只用来扩展一个用例。泛化关系描述两个或多个相似用例间的共同行为。为了描述的方便,我们一般只描述单重继承。特殊用例必须将一般用例的抽象行为具体化,并且可以重写一般用例的有关行为,重写的用例片断以“#”开头。

2.3 事件层

事件层主要用来细化用例片断。该层描述五种事件:消息、活动以及时间的设置、重置和终止。每个片断由若干相关的事件组成。事件之间的关系主要有:顺序、选择、循环和并行。图 3 描述了事件层各

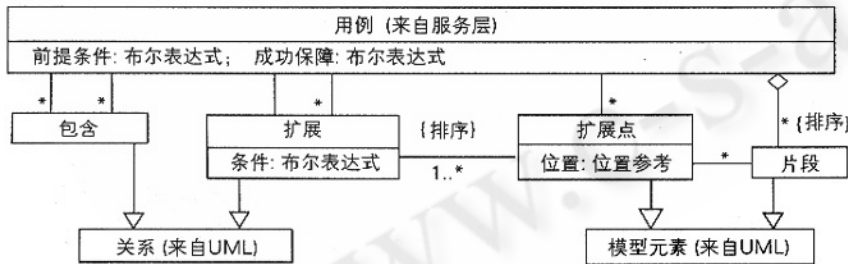


图 2 结构层各元素之间的关系

2.2 结构层

结构层主要描述用例之间的关系以及用例内部用例片断(episode)的组成。用例片断是一个事件流中功能相关的部分事件流。一个用例涉及的事件可能非常多,将功能相关的一系列事件用用例片断来模块化,不仅便于用户理解,而且便于用例间描述的重用。有关用例片断的细节将在事件层进行描述。该层描述的

元素之间的关系。表 3 给出了事件层各元素的图形表示。从表 3 可以看出,事件层的描述图形类似于 UML 的消息图,其扩展主要在两个方面:一是时钟事件描述符,另一个是控制流描述符。时钟约束往往是对系统的行为要求的一个重要方面。比如,用户在输入密码时,必须在一个规定的时间段内完成,否则输入无效。控制流描

述可以使得图形描述更加简洁。除此外,扩展点和条件也是对 UML 的消息图的一个扩充。条件除可以用来表示控制流的条件描述外,也可用来描述系统的状态。扩展点主要是为了满足用例扩展关系描述的需要。同时事件层描述的用例片断便于用例内部描述的重用。

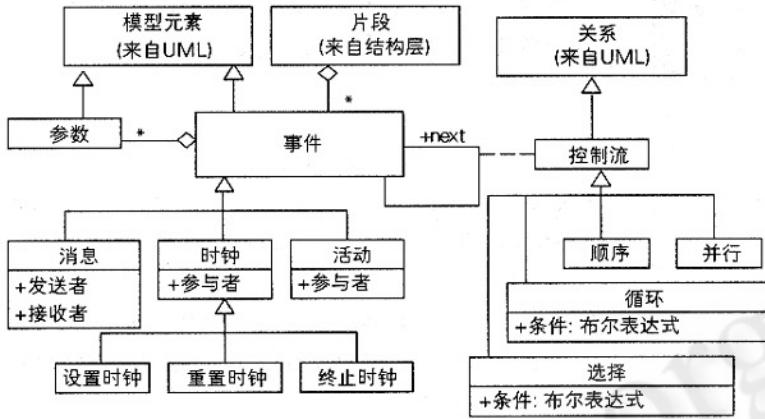


图 3 事件层各元素之间的关系

表 2 结构层各元素的图形表示

| 元素名称 | 图形符号 |
|------|--------------------------------------------------------------------|
| 用例 | |
| 包含关系 | |
| 扩展关系 | |
| 泛化关系 | |
| 扩展点 | {<扩展点名称>} |
| 用例片段 | . <片段名称> [({<扩展点名称> })] 或 # <新片段名称> \$ <旧片段名称> 或 INCLUDE <用例名称> |
| 前提条件 | ?<布尔表达式> |
| 成功保障 | !<布尔表达式> |

3 工具开发

虽然可以采用手工的方式进行用例描述,但效率会很低。用例描述工具的主要用途就是让用例工程师能够灵活、方便、快捷地对用例进行描述和维护,同时

便于用户和软件开发人员使用。为此,我们选用 Drawlet 框架^[5]来开发工具的图形表示界面,用 XML 来存储图形表示结果。为了绘制用例层次模型,我们在 Drawlet 框架的基础上做了两个方面的工作:一是添加新的图形以便绘制参与者、用例及它们间的关系;另外将图形信息采用 XML 文档格式保存。图 4 给出了建立服务层的用户界面,有关工具在进一步完善之中。

4 结束语

本文针对用例描述及应用特点,提出了一种基于重用的用例描述层次模

表 3 事件层各元素的图形表示

| 元素名称 | 图形符号 |
|------|------|
| 消息 | |
| 活动 | |
| 设置时钟 | |
| 重置时钟 | |
| 终止时钟 | |
| 并行 | |
| 循环 | |
| 选择 | |
| 扩展点 | |
| 条件 | |
| 参与者 | |

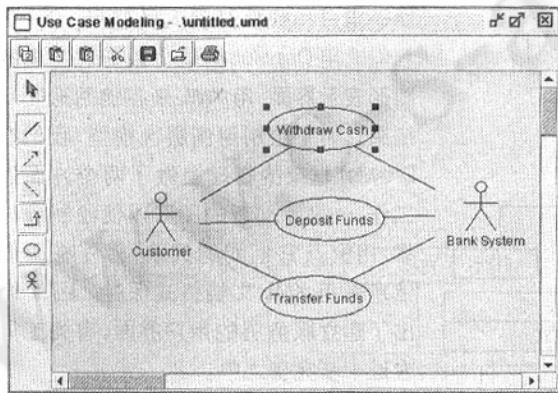


图 4 服务层描述用户界面

型,明确定义了用例的描述方式、内容以及粒度,规范了用例间关系的描述。该模型将用例的描述分为三个不同的抽象层次,便于用例内部、用例之间和不同系统间用例描述的重用,降低了用例描述的复杂性,而且模型简单、可视,方便用户理解、使用和维护用例描述。

目前,该用例描述模型对应的原型工具已在 Drawlet 框架的基础上开发成功。

参考文献

- 1 I. Jacobson. Use Cases and Aspects – working Seamlessly Together [J]. Journal of Object Technology, 2003, 2(4): 7 – 28.
- 2 B. Regnell. Requirements Engineering with Use Cases [D]. Lund: Department of Computer Science, Lund University, 1999.
- 3 F. Aamour. Advanced Use Case Modeling (影印) [M], 北京 科学出版社, 2003。
- 4 I. Diaz, F. Losavio, A. Matteo, et al. A Specification Pattern for Use Cases [J]. Information & Management, 2004, 41: 961 – 975.
- 5 RoleModel Software. <http://www.rolemodelsoft.com/drawlets/index.htm>.