

基于面向方面和 UML 的实时系统建模研究

邓惠敏, 张立臣, 邓建波

(广东工业大学 计算机学院, 广东 广州 510090)

摘要:利用面向方面的编程思想,在实时系统设计的需求分析阶段分离系统的功能性核心关注点和非功能性横切关注点,并着重对系统的非功能性等横切性关注点进行分析。通过扩展 UML 对实时系统时间切面建模,讨论了时间方面织入过程,利用 AspectJ 对建模的结果进行简单的编码实现。通过实际 ATM 自动提款机的实际例子说明如何在实时系统开发中运用这些技术进行系统建模和实现,充分显示了这些技术对实时系统进行分析建模的优越性。

关键词:面向方面;实时系统;横切关注点;统一建模语言

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2010)12-0118-04

Research of Real-Time System Modeling Based on Aspect-Oriented and UML

DENG Hui-min, ZHANG Li-chen, DENG Jian-bo

(Faculty of Computer Science, Guangdong University of Technology, Guangzhou 510090, China)

Abstract: Discussed how to separate the functional core concerns and the non-functional crosscut concerns based on aspect-oriented during requirement and analysis phase, and focus on the non-functional analysis of crosscut concerns, and then also discussed how to model and weave the time aspect of real-time system by extending UML and implement it simply by the AspectJ. Last, it gave an example of Automatic Teller Machine to show how to use these technologies for modeling real-time system, and also show the advantages of these technologies when model and implement real-time system.

Key words: aspect-oriented; real-time system; crosscutting concern; unified modeling language

0 引言

实时系统是一类对时间性要求十分严格的系统,系统的正确性不仅依赖系统计算的逻辑结果的正确性,还依赖于产生结果的时间。实时系统广泛应用于银行、军事、交通控制、核发电、工厂控制设备等领域。传统的对实时系统的开发研究一般是利用面向对象编程(Object-oriented Programming, OOP)的思想对系统的功能进行分解,然后对分解出来的各个功能模块分别进行开发,最后整合为一个完整的系统。这样各个功能性和非功能性关注点就可能分散到不同的类中,造成开发出来的系统存在代码的纠缠,重用性差等缺点,对系统的开发和维护造成困难。面向方面编程

(Aspect-oriented Programming, AOP)^[1]在实时系统开发的需求分析阶段实现系统的关注点的有效分离,很好解决了以上的问题。而且, AOP 能将实时系统最重要的时间特性作为一个独立的方面来进行开发,建立一个时间模型来进行时间方面的管理。统一建模语言 UML 已经成为了事实上的建模标准,但 UML 缺乏形式化描述的精确性,尤其对软件非功能特性的描述不够准确。因此通过对 UML 进行扩展,使其适用于对实时系统建模的描述是一个十分有意义的研究方向。

1 基本概念

1.1 关注点

IEEE 将系统的关注点定义为^[2]“……那些与系统开发相关的兴趣点,它的操作或者其他方面对于一个或多个使用者非常关键或非常重要。”因此可以认为关注点就是一个所要解决的需求问题,它是软件分析、设计、演化以及重用过程中所关心或感兴趣的事物,它可以分为核心关注点(Primary Concern)和横切关注点(Crosscutting Concern)。核心关注点封装了软件系统

收稿日期:2010-05-08;修回日期:2010-08-04

基金项目:国家自然科学基金重大研究计划资助项目(90818008);国家自然科学基金项目(60774095,60474072Z);广东省自然科学基金项目(07001774,04009465)

作者简介:邓惠敏(1983-),男,广东新兴人,硕士研究生,研究方向为面向方面,分布式实时系统;张立臣,博士,教授,研究方向为分布式处理和实时系统等。

的功能成分,它描述的是系统“做什么”的概念;横切关注点封装了软件系统的非功能或约束成分,它描述了系统“如何做”的概念,一般地,系统的非功能关注点主要包括系统响应时间、日志记录、异常捕获、安全性等。

1.2 AspectJ 主要术语

AspectJ^[3~5]是 eclipse 开源组织开发的面向方面 Java 编程的实现,到文中完稿时其版本已经到了 AspectJ1.6.8。AspectJ 现在技术已经比较成熟了,因此这里只介绍 AspectJ 的一些基本术语,以便对面向方面进行论述和研究。

“切面(aspect)”：系统要实现的交叉功能,是系统实现横切关注点的模块化单元。如日志记录。

“通知(advice)”：也可以叫做“装备”,指切面在程序流程运行到某个连接点所触发的动作。目前 AOP 定义了五种通知:前置通知(Before advice),返回后通知(After returning advice),环绕通知(Around advice),异常通知(After throwing advice),引入通知(Introduction advice)。

“连接点(join point)”：应用程序执行过程中激活 advice 被执行的触发点,可以是方法调用,异常抛出,或者要修改的字段。

“切点(pointcut)”：一系列连接点利用与或非("&&"|"和"!")组成的切点。

“引入(introduction)”：提供修改基类的特性,为类添加新的方法和属性。

1.3 UML 扩展机制

UML Profile 扩展机制主要包括以下三种^[6]：

“构造型(Stereotype)”：指对于一个已经存在的模型元素定义一种新的模型元素,新的模型元素是已有模型元素的子类,通过泛化关系继承了原模型的特性,并对原模型进行了扩展,构造型一般用双尖括号(<<>>)中的字符串标识;

“标记值(Tagged value)”：用于对元素增加新的语义,是一种指定模型元素性质的机制;

“约束(Constraint)”：则是对元素进行语义的限制。

可以通过这些机制来扩展 UML 使其满足建模的需要。

2 关注点分离

2.1 捕获关注点

捕获关注点首先要理解问题域,系统能够正确反映出来的问题域就是一些十分重要的涉众关注点^[7]。这可通过“类”有效实现。例如,如果你要开发一个 ATM 自动取款机系统,就必须了解客户、账户、打印、

取款、转账等概念。其次就是要抽取系统的特性,生成关键特性列表。表 1 列出了 ATM 自动取款机的关键特性。

表 1 ATM 自动取款机的关键特性表

1. 客户可以查询、取款、转账、存款、修改密码、挂失、打印小票
2. 银行可以启动系统、检测系统、关闭系统、冻结或者解冻账户
3. 在关系数据库中存储所有账户记录
4. 所有事件必须记录操作日志
5. 只有授权的用户才能使用
6. 输入错误密码次数超过 3 次,系统自动吐卡
7. 输入密码时间为 30 秒,超时自动退卡
8. 转账时,客户在输入转账金额之前超时,自动取消操作
9. 存款时,客户在插入现金之前超时,自动取消操作
10. 2 分钟内没能从银行计算机获得响应,则将卡退出并显示错误信息
11. 只有当账号提取现金处理完成,被银行接收后,ATM 才弹出现金

2.2 分离核心关注点

一般与业务逻辑相关的关注点都可以看作核心关注点,核心关注点可以从用例中很好分离出来。如图 1 所示为 ATM 自定提款机系统终端用例图,从用例图中可以清楚地看到系统存在的核心关注点。

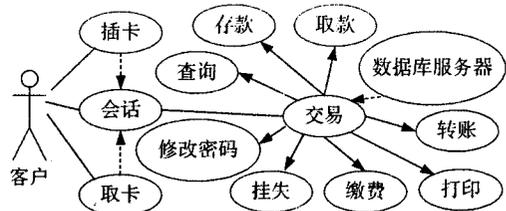


图 1 ATM 终端用例图

从以上用例中可以得到如下的核心关注点:查询、存款、取款、转账、缴费、挂失、修改密码、打印。同样针对银行人员,可以提取出冻结账户、预放现金、启动系统、关闭系统和检测系统这几个关注点。

2.3 分离横切关注点

系统的横切关注点主要通过分析用户需求,从中抽取横切整个系统的关注点。横切关注点可以作为扩展用例处理,最终合并到需求模型中去。

从表 1 中可以看出第 4 至第 10 条记录均为系统的非功能特性,记为 R₄、R₅…R₁₀。因此可以将这些特性总结为日志记录、安全性、时间性这 3 个横切性关注点。表 2 为 ATM 系统的时间性横切关注点的描述。

3 关注点建模

关注点建模目前还停留在一个抽象的程度,但已经有一些方法在研究对它进行处理。文献[8]提出了一种基于概念的方法 CoCompose,它能够对早期方面

进行建模。文献[9]为关注点建模提出一种超空间方法的实现:能用非正交的关注点簇代替正交的关注点维度,并且在一个维度上一个单元允许被赋予多个关注点。主要维度基于软件构件,而次要维度代表了用户的兴趣。以上都是早期对关注点建模技术的一些研究。

表 2 时间性横切关注点

横切关注点	时间管理
描述	用户操作或服务器响应超时处理
属性	$T \geq T_{max} > 0$ (T_{max} 为时间上限)
需求集合	$R_4, R_5 \dots R_{10}$
模型	输入密码、转账、存款、取款

3.1 扩展 UML 表达方面概念

扩展之后的 UML 应该能够表达方面、核心功能性类、方面与核心类之间的关联以及方面和核心类之间的关联,并描述这些元素的结构和行为。文献[6]给出了扩展方面构造型的规范,这里不再赘述。

3.2 时间切面模型

实时系统的面向方面模型把时间管理作为一个方面进行独立的处理^[10],因此可以建立一个时间模型来表达系统时间管理。ATM 实时系统时间模型的时间约束条件如下:

[T_1]:用户插入 CreditCard 卡后输入密码,若输入时间超过最大上限时间(如 30s),则系统自动退卡, $T_{in} > 30s \rightarrow Card_{Out}$

[T_2]:用户存款时候,若用户在插入现金前超过最大上限时间(如 30s),系统自动取消操作, $T_{put} > 30s \rightarrow Card_{Cancel}$

[T_3]:用户在转账时候,若用户在输入转账帐号前超过最大上限时间(如 30s),系统自动取消操作, $T_{sys_in} > 30s \rightarrow Card_{Cancel}$

[T_4]:2 分钟内没能从银行计算机获得响应,则将卡退出并显示错误信息, $T_{max} > 2 * 30s \rightarrow Card_{Out}$

4 实例分析

ATM 自动取款机是一个典型的实时系统,通过上面对系统特性的分析,对系统关注点的捕获以及对时间切面的建模分析的基础上可以对该系统进行模拟建模。

4.1 ATM 终端静态模型

通过分析可以建立如下的 ATM 终端总体类图,如图 2 所示,并表示出时间方面模型。图中只建立基本的键类图关系,对像登陆验证、异常处理、日志记录等都可以作为一个横切的方面来处理,这里不作讨

论。

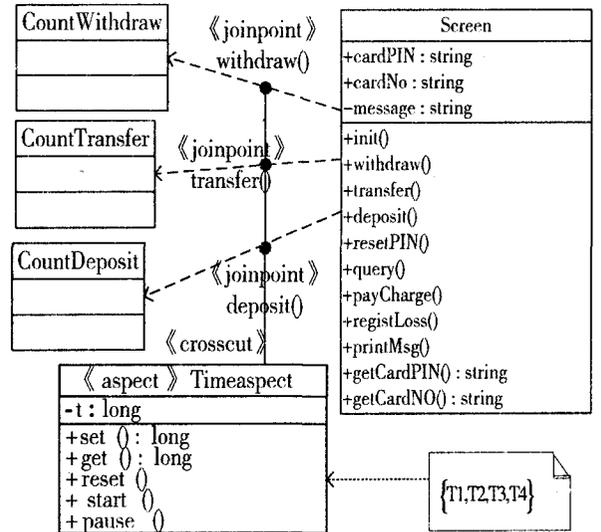


图 2 ATM 终端总体类图

4.2 ATM 时间方面织入模型及其 AspectJ 实现

方面的织入过程在本质上来说是寻找切入点的过程,下面分析 ATM 系统登陆阶段的密码验证顺序图为例说明时间方面的织入过程。

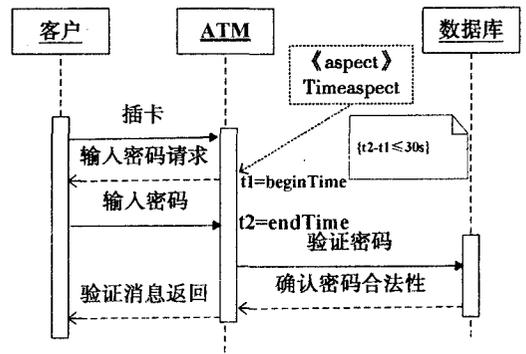


图 3 密码验证(未超时)

图 3 中 ATM 读取用户的信用卡信息后,发出要求用户输入密码的请求 `getCardPIN()`,同时触发时间管理切面 `Timeaspect`。经过切面内部算法的处理,若用户未超时则交互过程如图 3 所示;若用户超时则按图 4 过程进行。

下面简单介绍上述过程的 AspectJ 的实现过程:

```

aspect Timeaspect {
    pointcut timeCut():
    call(public? *? getCardPIN());
    after timeCut():checkTime();//时间检测及控制操作
}

```

上面一段小小的代码,它就可以方便实现时间方面的织入,使时间管理从真正的业务代码中分离出来,有利于提高代码的可重用性及其模块化程度。这里列举的只是部分代码,它代表了本实例如何利用 AOP 进行系统开发的一个典型例子。

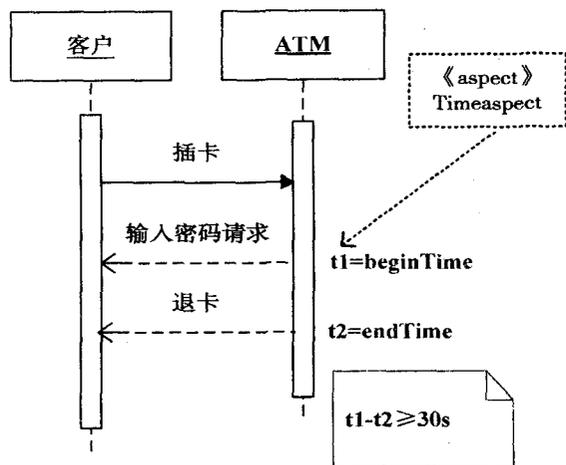


图 4 密码验证(超时)

4.3 方面织入时冲突的解决

方面织入时很容易出现关注点组合的各种冲突问题,例如系统的响应时间和系统安全性的冲突,冲突行为可以通过确定优先权的办法来解决,如果冲突的非功能需求优先权相同则可以通过与用户协商进行解决。

文献[11]给出了一种基于时序逻辑的关注点冲突解决办法,文献[12]给出了一些基于优先权矩阵的关注点冲突的解决办法,因此这里不再讨论。

5 结束语

文中基于面向方面思想讨论了实时系统关注点建模的技术,实时系统还有许多可以抽取的方面,比如异常处理、日志记录、安全检测等等都可以通过上面介绍的方法对其进行建模分析。面向方面编程现在比较成熟了,如何很好地利用这一技术为实际应用开发还是要解决的问题,在实时系统中利用这种技术进行开发还是比较少人研究,其难点还是如何解决好切面的分

离和组合的问题,这由实时系统本身的特点决定。文中的后期主要工作是横切关注点的形式化描述和验证。

参考文献:

- [1] Filman R E, Elrad T, Clarke S, et al. 面向方面的软件开发[M]. 莫倩, 王恺, 刘冬梅, 等译. 北京:机械工业出版社, 2005.
- [2] IEEE Std 1471 - 2000 IEEE Recommended Practice for Architectural Description of Software - Intensive Systems Description[S]. New York: ISO, 2000.
- [3] 王海龙. What is AspectJ [EB/OL]. 中国, 2003. <http://www.ibm.com/developerworks/cn/java/1-aspectj/index.html>.
- [4] Gradecki J D, Lesiecki N. 精通 AspectJ [M]. 王欣轩, 吴东升, 等译. 北京:清华大学出版社, 2005.
- [5] Aspect - Oriented Programming with AspectJ [M]. [s. l.]: Ivan Kiselev, 2002.
- [6] 刘瑞成. 基于 UML 和形式化方法的面向方面实时系统模型[D]. 广州:广东工业大学, 2006.
- [7] Jacobson I, Pan - wei NG. AOSD 中文版——基于用例的面向方面软件开发[M]. 徐锋, 译. 北京:电子工业出版社, 2005.
- [8] Wagelaar D. A Concept - based approach for early aspect modeling [EB/OL]. Boston, 2003. <http://www.cs.Bilkent.edu.tr/AOSD-EarlyAspects/Papers/Wagelaar.pdf>.
- [9] 方义秋, 冉华锋, 葛君伟. 基于用例的面向方面需求建模[J]. 计算机工程, 2009, 35(12): 44 - 46.
- [10] 刘瑞成, 张立臣. 基于 UML 的面向方面建模方法[J]. 计算机科学, 2005, 32(10): 204 - 209.
- [11] 张琳琳, 应时, 倪友聪, 等. 一种软件体系结构关注点分析方法[J]. 计算机学报, 2009, 32(9): 1782 - 1791.
- [12] 郑旭飞. 一种面向方面的非功能需求框架 AONFRF 建模研究[D]. 重庆:西南师范大学, 2005.

(上接第 117 页)

- algorithms[M]. New York: Plenum Press, 1981.
- [3] 李春生, 王耀南, 陈光辉, 等. 基于层次分析法的模糊分类优选模型[J]. 控制与决策, 2009, 24(12): 1882 - 1884.
 - [4] 孔峰. 模糊多属性决策理论方法及应用[M]. 北京:中国农业科学技术出版社, 2008: 78 - 100.
 - [5] 李玲娟, 豆坤. 层次分析法中判断矩阵的一致性研究[J]. 计算机技术与发展, 2009, 19(10): 24 - 26.
 - [6] 王雪华. 两种层次结构化决策方法的理论与应用研究[D]. 大连:大连理工大学, 2005.
 - [7] 钱刚, 冯向前, 徐泽水. 区间数互补判断矩阵的一致性[J]. 控制与决策, 2009, 24(5): 723 - 728.
 - [8] 徐泽水, 吴应宇, 达庆利. 一种改进的行和归一化排序方

- 法[J]. 东南大学学报(自然科学版), 2004, 34(4): 518 - 521.
- [9] Saaty T L. Highlights and critical points in the theory and application of analytic hierarchy process[J]. European Journal of Operational Research, 1997, 74: 426 - 447.
 - [10] 吕跃进. 指数标度判断矩阵的一致性检验方法[J]. 统计与决策, 2006, 24(18): 31 - 32.
 - [11] 左军. 层次分析法中判断矩阵的间接给出法[J]. 系统工程, 1988, 6(6): 56 - 63.
 - [12] 徐泽水. AHP 中两类标度法的关系研究[J]. 系统工程理论与实践, 1999, 19(7): 97 - 101.