

AOP 在 HPMIS 中的应用

卢 娟, 朱 光, 张辉飞, 张云华

(浙江理工大学信息电子学院, 杭州 310018)

摘 要: 结合医院人员管理信息系统(HPMIS)中权限管理的开发过程,分析采用 OOP(object oriented programming)技术处理横切关注点引发的问题和困难,研究在此情况下采用 AOP(aspect oriented programming)技术来解决 HPMIS 中横切关注点模块化问题的解决方案。通过 AspectJ,给出了 AOP 技术在 HPMIS 权限控制开发中具体的应用示例,实现了非主要功能的集中模块化,并在此基础上讨论了 AOP 技术的优越性。

关键词: 管理系统; 面向方面编程; 横切关注点

中图分类号: TN912.35

文献标识码: A

0 引 言

目前,面向对象编程(OOP)已经成为了主流程程序设计范型,是大多数软件项目开发所采用的编程技术。OOP 在对系统的功能模块(纵向)进行建模时,能够很好地完成对核心关注点的设计与开发。但是在现实中,常常会遇到的横跨多个不相关模块的公共行为,例如日志记录、异常处理、安全控制以及性能统计等等“横切关注点(crosscutting concern)”,采用 OOP 技术进行处理时,就会造成代码的分散以及模块内部的代码混乱,从而导致系统开发难度增大、系统维护性差以及不易扩展。

当前应运而生的 AOP(aspect oriented programming 面向方面编程)技术,为处理跨模块横切行为问题提供了很好的解决方法,它通过对横切关注点进行模块化,实现关注点的分离(separation of concerns),有效弥补了 OOP 技术在处理这方面问题的不足,提高了软件的设计水平及可复用性。本文结合医院人员管理系统中权限控制管理的开发过程,分析 OOP 处理横切问题的不足,拟采用 AOP 技术的解决方案,通过 AspectJ 给出 AOP 技术封装权限控制该横切关注点的示例。

1 基于 OOP 的 HPMIS 及其权限控制管理

1.1 HPMIS 以及权限控制管理

医院人员管理信息系统 HPMIS (hospital personnel management information system)是现代化医院运营有力的技术支撑和基础技术环境,有利于以更现代化、科学化、规范化的手段来加强对医院工作人员的管理,提高医院的工作效率、降低消耗,推进医院信息化管理的发展步伐^[1]。图 1 是 HPMIS 的系统结构。

从图 1 中可以看出,作为一个综合、复杂的人员管理信息系统,医院人员管理信息系统 HPMIS 的核心业务表现在各业务子系统中,而权限控制(access control)管理模块是各个业务子系统都涉及到的公共需要,它在各子系统中的功能是基本相同的,即为一个横切关注点。

权限控制简单地说,就是 Who(谁)对 What(什么)有 How(怎样)的权限的问题。对于 HPMIS 这样一个多用户的软件系统,需要进行权限的限制,规定每个用户只能操作其职责范围内的功能而不能使用其他功

能。这样可以保证数据的安全,防止非授权的访问,由此可见权限控制起着很重要的作用。

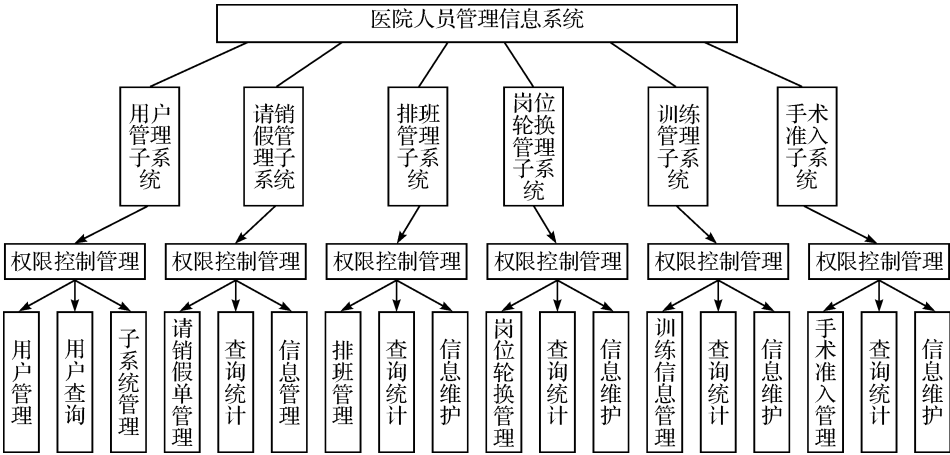


图1 HPMIS的系统结构

1.2 OOP 处理横切关注点的局限性

所谓关注点(concerns),一般理解为软件需要解决的问题。根据特征不同,一个典型的软件系统的关注点可以分为核心关注点(core concern)和横切关注点(cross-cutting concern)^[2]。在 HPMIS 中,核心关注点包括医院人员请销假的查询统计、排班轮岗的安排以及手术准入的申请等;而横切关注点,从图 1 可以看出,分布在多个核心模块中的权限控制管理就是一个横切关注点。

OOP 技术的出现改变了人们设计和编写软件的方法,它可以很优秀地完成对核心关注点的设计和开发,但是在处理横切关注点时却显得力不从心。在使用 OOP 方法开发 HPMIS 的过程中,横切关注点跨越了多个业务功能模块,是多维的公共行为;而 OOP 的设计方法却是一维的,如果使用 OOP 编程方法,就需要把多维的需求映射到一维上,这样就造成了核心关注点和横切关注点之间的强耦合。当需要增加新的横切特性,或者修改现存的横切功能时,则需要对所有相关的核心模块都进行操作,这显然会对软件的开发带来不利影响。

如果在 HPMIS 核心业务功能中都进行权限控制操作,那么利用 OOP 的方法进行编程代码是这样的:

```
public class HandleClass
{
    //core data members
    //other data members;for access control
    public static void Approve()
    {
        //ensure authentication
        //=====perform business logic=====
    }
    Public static void Refuse()
    {
        //ensure authentication
        //=====perform business logic=====
    }
}
```

由以上代码,可以看出:第一,该类中的其他数据成员不是核心关注点;第二,函数实现中加入的权限验证代码与核心功能无关,这样会造成代码混乱的问题;第三,处理权限验证的代码分散到了多个核心模块中,如上例所示函数 Approve 和函数 Refuse 中都加入了类似的代码,这样就造成了代码分散的问题。

代码混乱和代码分散会导致模块耦合度高、代码可读性差、代码重用率低、代码的质量低、系统的扩展和

维护困难。这些问题是我们在开发和设计系统的时候应该尽量避免的。

2 面向方面编程(AOP)

2.1 AOP 实现方法和步骤

面向方面编程技术 AOP 可以解决上述问题。AOP 是由施乐公司帕洛阿尔托研究中心(Xerox PARC)的 Gregor Kiczales 等在实践基础上提出的,并开发了第一个 AOP 开发环境 AspectJ。AOP (aspect oriented programming) 是 OOP 的延续,它实际上是 GoF 设计模式的延续,将那些与业务无关,却为业务模块所共同调用的逻辑或责任,例如事务处理、日志管理、权限控制等,封装起来,实现系统关注点的分离,如图 2 所示。

AOP 技术的出现为开发者提供了一种描述横切关注点的方法,人们可以通过它实现横切关注点的模块化^[3]。从而使代码的修改范围得到很好的控制,减少代码修改对系统稳定性的影响,减少代码的冗余度和耦合度,增强可读性,提高软件质量。

通过划分 Aspect,横切关注点问题可以迎刃而解。基于 AOP 的程序开发过程示意图如图 3 所示。

利用 AOP 进行开发分为三个步骤组成:

- a)方面分解 需求分析,将一个系统中的核心关注点和横切关注点分离开来。
- b)方面实现 OOP 方法实现一般关注点,AOP 方法对横切关注点封装。
- c)方面织入 方面编织器通过创建一个模块单元(方面)来指定重组的规则。通过编织器将通过 AOP 实现的模块单元织入 OOP 实现的基础代码,进行组合生成满足系统需求的代码^[4]。

2.2 ApectJ 语言元素

a)Aspect(切面):通知和切入点的组合,称作切面,或者方面。实现了 cross-cutting 功能,是针对切面的模块,它包括所有的切点、通知和类型间声明。

b)Jointpoint(连接点):它是在运用程序执行过程中需要插入切面模块明确定义的点,它可以定义在方法调用、条件检测或是赋值动作处。连接点有一个与之相关联的上下文。

c)Pointcut(切入点):一个切入点是用来定义某一个通知(advice)应该如何执行的一组连接点(即,切入点是一组连接点),它可以精确地控制程序中什么组件接到什么通知。

d)Advice(通知):Advice 是对 aspect 的一个具体实现,在某一个特定的连接点处执行的代码,用来实现 AOP 的业务逻辑。典型的有前置通知(before advice)和后置通知(after advice)^[5]。

e)Introduction(引入):允许添加新的方法和属性到类中,以改变它的结构。

f)Weaving(织入):将 Aspects 真正加入到程序代码的过程。静态 AOP 在编译时完成织入^[6],动态 AOP 是在程序运行时动态织入的。

g)Target(目标):Target 是正在被 advice 的类,可以是你自己写的一个类或者是你想增加新的功能的第三方提供的类。

h)Proxy(代理):Proxy 是被应用了 advice 的 target 实例。

图 4 为 AspectJ 语言元素的图例。

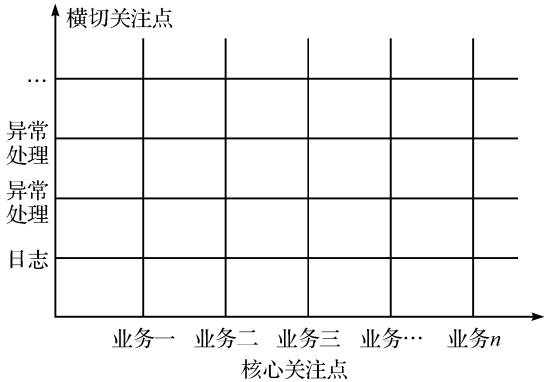


图 2 典型的系统关注点分离

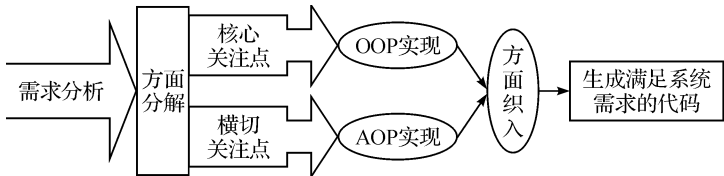


图 3 面向方面编程的开发步骤

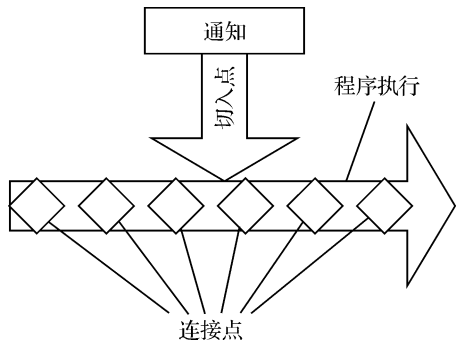


图 4 AspectJ 语言元素图例

3 AOP 技术在 HPMIS 中的应用

从本文上述内容可见,使用 OOP 技术处理横跨多个功能模块的公共行为会造成代码混乱和代码分散,增大了系统扩展和维护的难度。AOP 利用一种称为“横切”的技术,剖解开封装的对象,并将那些影响了多个类的行为封装到一个可重用模块,称之为方面(aspect)。它实现了核心业务级关注点与横切关注点分离开,而 AOP 的核心思想是将应用程序中的业务逻辑(功能逻辑)同对其提供支持的 非功能逻辑进行分离。使用 AOP 技术解决 HPMIS 权限控制问题的示意图如图 5 所示。

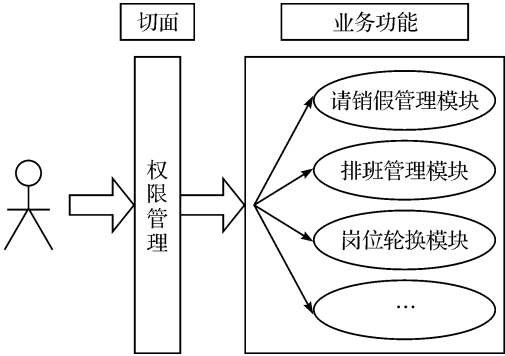


图 5 AOP 在 HPMIS 中的应用示意

以下是上文的例子采用 AOP 技术在 AspectJ 里的实现^[7]:

//权限控制方面模块

```
Public aspect AuthenticationAspect{
    .....
    Public pointcut authorizationExecution():execution(public void
    NewBussinessClass. Approve ())|| execution(public
    void NewBussinessClass. Refuse())||...;
    Before():authorizationExecution()
    {
        //Ensure authentication 进行身份验证
        if(! permissions. Verify(Permission. ADMIN))
        {
            Throw new UnauthorizedException();
        }
    }
    .....
}
```

//核心业务逻辑模块

```
Public class NewHandleClass extends HandleClass{
    Public static void Approve()
    {
        //=====perform business logic=====
    }
    Public static void Refuse()
    {
        //=====perform business logic=====
    }
    .....
}
```

以上代码定义了 AuthenticationAspect 方面,来处理需要进行的身份验证,方面用关键字 Aspect 标识。在该方面中,定义了一个切入点 authorizationExecution,用关键字 pointcut 来标识。由代码可以看出,如果执行 NewHandleClass 的 Approve()或者 Refuse ()等函数时,切入点 authorizationExecution 就会触发相应的动作。接下来的代码是关注点的具体实现,称作通知(Advice),表示切入点被触发后系统执行的动作,这是 AOP 的关键之处。Before 表示切入点之前做什么事,此处为身份验证。

从核心业务逻辑模块的代码可以看出,引入 Aspect 后,业务逻辑方法中不再包括身份验证等公共事务^[8],只有处理核心业务的功能代码,而横切关注点则被封装在了 Aspect 中,再通过 AOP 语言编译器的组织,就可以得到系统所需要的代码。

由此可见,引入 AOP 使程序结构清晰,并获得更好的关注点的分离,体现了软件工程中高内聚、低耦合的要求,从而使开发出来的系统结构清晰,易被重用、维护和扩展^[9]。

4 结 论

面向方面编程思想的提出,是为了降低软件的复杂性,更好地分离核心关注模块和横切关注模块。通过以上分析与研究可以看到,对于传统的 OOP 很难处理的横切关注点方面,AOP 技术提供了一个很好的解决方法。AOP 通过对横切点进行封装,可以解决 HPMIS 中的权限控制这一横切关注点模块化的问题,解耦了程序功能和权限检验。AOP 作为一种编程技术,是对 OOP 的扩展和完善。随着未来系统变得越来越复杂,横切关注点越来越成为一个大问题,会有越来越多的软件系统采用 AOP 面向方面的编程思想来进行开发、测试,以构建更好的松散耦合、易维护扩展的软件系统,AOP 也将会得到更大的发展。

参考文献:

[1] 张泳,苏健. AOP 在 HIMS 系统管理中的应用[J]. 计算机应用与软件, 2008, 25(10): 190-192.

[2] 郭东亮,张立臣. 面向方面软件开发研究[J]. 计算机应用研究, 2005(8): 36-38.

[3] 张广红,陈平. 关于 AOP 实现机制和应用的研究[J]. 计算机工程与设计, 2003(8): 14-17.

[4] 李志纯,张南平. 面向 Aspect 编程的应用研究[J]. 计算机技术与发展, 2006, 16(5): 217-222.

[5] Ivar Jacobson, Pan-Wei Ng. 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.

[6] 谭韩成,牛争艳,邓宏文,等. 基于 AOP 的 Web 系统权限管理[J]. 电脑开发与应用, 2006, 19(4): 20-22.

[7] 董华山,孙济庆,吉久明. AOP 在数据访问中间件中的应用[J]. 计算机应用与软件, 2005, 22(1): 53-54.

[8] Miles R. AspectJ Cookbook 中文版[M]. 北京:清华大学出版社, 2006.

[9] 石丹丹,王静,熊前兴. 面向方面编程模式的探讨[J]. 武汉理工大学学报, 2005, 27(1): 93-95.

Applying AOP to HPMIS

LU Juan, ZHU Guang, ZHANG Ye-fei, ZHANG Yun-hua

(School of Informatics and Electronics, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: This paper is based on the developing of access control management in a hospital personnel management information system. After analyzing the drawbacks of the conventional OOP means in solving the problems of crosscutting concern, the paper introduces the solution using AOP technology at the same situation. In the end the paper presents an implementation of AOP technology for solving access control in HPMIS using AspectJ, and achieves modularization of non-main functions. Finally the advantages of AOP technology are illustrated.

Key words: management system; AOP; crosscutting concerns

(责任编辑:陈和榜)