

AOP 技术在数据交换与共享系统中的应用

钱慎一, 付中举, 林青

(郑州轻工业学院 计算机与通信工程学院, 河南 郑州 450001)

摘要:针对数据交换与共享系统通用模块存在代码冗余的问题,引入 AOP 技术,对系统通用服务进行代码植入操作,实现了 AOP 框架在数据交换与共享系统的应用.试验结果证明,AOP 技术能够有效减轻系统负担,提高系统的时效.

关键词:数据交换;数据共享;AOP 技术

中图分类号:TP39 **文献标志码:**A **DOI:**10.3969/j.issn.2095-476X.2012.06.004

Application of AOP technology in data exchange and sharing system

QIAN Shen-yi, FU Zhong-ju, LIN Qing

(College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China)

Abstract: Aiming at the problem of code redundancy in data exchange and data sharing system, AOP technology was introduced in general service of system by code woven operation, the application of AOP frame was realized in data exchange and data sharing system. The experiment results showed that AOP technology can effectively reduce the burden and improve security of system.

Key words: data exchange; data sharing; AOP technology

0 引言

数据交换与共享系统的核心功能是可以用户通过系统的服务器端统一进入、统一访问、统一管理,用户登录时由系统自动进入相应的角色界面进行管理,系统服务器端管理员负责登记前置机的相应信息等.这要求该系统访问控制机制必须严谨高效.

目前,在用户的访问与系统基础管理、维护等方面,大多数数据交换与共享系统采用的是基于角色的访问控制(RBAC)^[1-2].但是,单纯地使用 RBAC 机制,并不能很好地解决在访问控制、部门管理、共享管理与日志管理等模块中代码大量重复、分散以及效率低下等问题.

AOP(aspect-oriented programming)是面向方面(切面)的编程 OOP(object-oriented programming)的补充与延续^[3].AOP 作为一种新的软件开发思想,是为了更好地解决对象中的方法具有通用性而代码冗余的问题^[4-5].鉴于此,本文在数据交换与共享系统中引入 AOP 技术,以解决系统通用模块代码的冗余问题^[6],并解除代码在数据交换与共享系统中的强耦合性,从而提高系统效率.

1 AOP 在数据交换与共享系统中的应用设计方案

1.1 数据交换与共享系统中的 AOP 技术路线

数据交换与共享系统应用面较宽,现已广泛应用于银行机构、金融债券公司、数字认证中心等领

域. 其核心功能是对用户访问各前置机的数据资源进行统一控制,通过登录与权限机制来访问获取资源列表,保证数据的安全性及时性. 值得注意的是,系统前置机上装载的是统一的数据交换标准,为引入 AOP 技术提供了便利. 采用 AOP 后,系统能够调用方面编程中的代码,识别存储在前置机中的特殊字段中的数据结构,并抽取日志、安全、事务等非业务代码为一个独立模块,插入到各执行业务中. 这样不仅解决了非核心模块业务逻辑与核心模块业务逻辑的代码需解耦合的问题,也降低了系统维护的难度. 数据交换与共享系统的整体系统结构设计如图 1 所示.

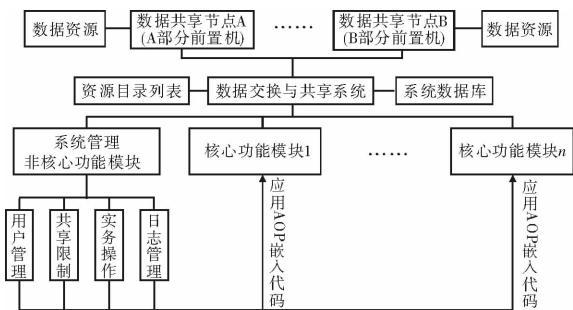


图 1 基于 AOP 的数据交换与共享系统结构图

1.2 AOP 技术在数据交换与共享系统中的设计理念

AOP 是在 OOP 的基础之上发展起来的,这并不能说明 AOP 的发展对 OOP 有取代性的作用^[7]. 作为对 OOP 的补充,AOP 针对 OOP 中代码强耦合的问题,提出将应用程序中的商业逻辑和对其进行支持的系统通用服务进行分离的思想^[8].

AOP 把软件系统分成 2 部分,即核心关注点和横切关注点. 核心关注点是业务处理的主要流程,也就是说这个解决方案要做的事;横切关注点是与核心关注点无关的部分. 笔者结合数据交换与共享系统的总体设计,明确系统非核心模块(如共享限制、事务操作与日志管理等)都可以很方便地使用 AOP 来实现(此处用户管理子模块的安全交由 RBAC 机制保障). 这种将影响多个类的公共行为封装到一个可重用方面的编程思想,具体表现在:在涉及用户调用核心模块时,使用 AOP 技术对非核心模块子模块的核心关注点进行操作,紧接着以模块化在横切关注点处采用与应用程序无关的方式处理相应的操作^[9]. 图 2 给出了用户在操作之前、之中、之后捕获目标的应用程序并进行相应操作的过程.

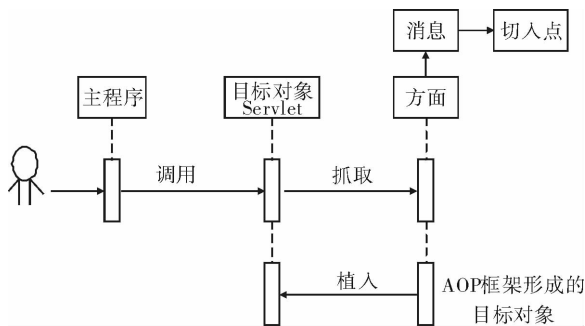


图 2 系统操作顺序图

当用户调用需要响应的 Servlet 程序后,执行响应的操作,aspect 将会通过 advice 捕捉到切入点,并进行植入操作,处理方面编程的相关操作.

2 AOP 技术在数据交换与共享系统中应用的实现

目前,宣称能够支持 AOP 的架构平台已达近百种,实现了基于 Java 语言的 AOP 框架也有 20 多种,其中较为完善的是 Spring AOP 和 Aspect J. 综合考虑系统的构架,本文使用 Aspect J 作为面向方面编程的一个框架,使用 myeclipse 作为编程工具,在 2.5 GHz 主频的实验环境下进行数据交换与共享系统的软件开发.

数据交换与共享系统被分为 2 部分:核心功能模块所处理的业务流程将被设计为核心关注点;非核心功能模块(不包括用户权限的设计与用户管理部分)被设计成横向关注点. 以日志管理为例,部门管理员角色的用户在管理资源时存在对文件与数据的增加、删除与更新操作,这时系统需要将这些操作过程记录在案,因此需要引入 AOP 编程技术. 通过用户的增删改操作,可以插入一个横切面,形成一个切入点,从而实现跟踪操作.

下面给出的是在项目中部门管理员对相应的文件或者数据进行增加、删除与更新操作时如何插入切入点进行日志记录的部分代码:

```
public class DepManServlet
{
    public void insert ( HttpServletRequest request, HttpServletResponse response )
    {
        /* insert the corresponding data or filestreams into database */
    }
    public void update ( HttpServletRequest request, Http
```

```

tpServletResponse response)
{
    /* update the corresponding data or filestreams from
    database */
}
public void delete ( HttpServletRequest request, Http
ServletResponse response)
{
    /* delete the corresponding data or filestreams from
    database */
}
}

```

使用 Aspect J 定义一个名为 Logging 的方面,以
实现用户对日志方面的管理,部分代码如下:

```

public aspect Loggin
{
    pointcut Logcap( HttpServletRequest request, HttpS
ervletResponse response ) :
        execution ( * DepManServlet. insert ( HttpServle
tReque, HttpServletResponse )
            &&args ( request, response ) ;
        after ( HttpServletRequest request, HttpServletResponse
response response ) ;
        returning : Logcap ( request, response )
}
/* do logging */
}
}

```

以代码执行相同行数的时间为度量,分别测试
原系统与应用 AOP 框架后的系统性能,具体的系统
时效图如图 3 所示.

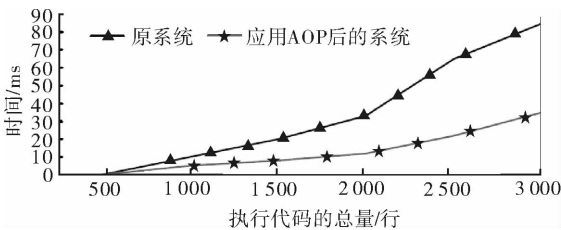


图 3 使用 AOP 技术的系统时效图

从图 3 可知,未采用 AOP 方面编程策略的数据
交换与共享系统在代码执行量为 500 行、1 000 行、
1 500 行、2 000 行、2 500 行与 3 000 行的耗时分别
是 8 ms, 15 ms, 28 ms, 52 ms, 74 ms 与 88 ms. 但在原
系统应用 AOP 后,系统耗时降低 50%, 时效性得到
增强. 当在 Java 代码中添加类或者添加新的方法后,

程序员不必加入新的横切面来处理相应的操作,只需
要改动类中的连接点标识,日志管理等模块就会自动
应用到相应的位置上. 从图 3 可知,AOP 的方面编程
以其独有的优势不仅解决了用户在非核心模块上代
码冗余的问题,而且提高了系统的效率.

3 结论

本文研究表明,原始开发模式的通用业务中非
核心部分的代码强耦合性、冗余性可以通过引入
AOP 技术得到很好的处理. 试验结果证明,将 AOP
应用于数据交换与共享系统中,不仅能够很好地协
助 RBAC 访问机制控制对用户的身份认证,而且能
够高效分离核心业务逻辑与非核心业务逻辑,使得
访问控制过程更加严密清晰. 通过对相应权限用户
安全记录、事务处理、共享限制与日志的管理,可以
跟踪后台数据库的存储情况,进一步提高系统安全
分析员对系统安全的审计情况,符合数据交换与共
享系统的总体设计思想.

参考文献:

- [1] Gradecki J D, Lesieckin N. Mastering Aspect J: Aspect-Oriented Programming in Java [M]. Indianapolis: Wiley Publishing Inc, 2003.
- [2] 伍建晖. 基于角色的访问控制在 J2EE 中的研究及扩展 [D]. 南京: 南京理工大学, 2006.
- [3] 张英捷, 刘万军. SpringAOP 技术在 J2EE 系统安全性验证中的应用研究 [J]. 计算机工程与科学, 2008, 30 (8): 137.
- [4] 钟秀琴, 符红光, 余莉, 等. 基于本体的几何学知识获取及知识表示 [J]. 计算机学报, 2009, 33 (1): 167.
- [5] 李森, 白勇, 张波. 基于领域特征的 AOP 编织实现方法 [J]. 计算机科学, 2009, 36 (2): 299.
- [6] 郑汉雄, 郑汉英, 周晓聪. 在 AOP 中使用标注改进日志功能的实现 [J]. 计算机工程, 2009, 34 (17): 71.
- [7] Havinga W, Nagy I, Bergmans L, et al. Detecting and resolving ambiguities caused by inter-dependent introductions [C] // Proc of the 5th International Conference on Aspect-oriented Software Development, New York: ACM Press, 2006: 214 - 225.
- [8] Kiczales. An overview of aspectJ [C] // ECOOP, Object-Oriented Programming, Budapest: Springer-Verlag, 2001: 327 - 353.
- [9] Laddad R. Aspect J in Action: Practical Aspect Oriented Programming [M]. Greenwich: Manning Publications Co, 2003.