

# 基于 AOP 的 Web 系统事务管理的应用研究

周改霞<sup>1</sup>, 高尚<sup>1</sup>, 杨镇江<sup>2</sup>

(1. 江苏科技大学, 江苏 镇江 212003; 2. 郑州大学, 河南 郑州 451000)

**摘 要:**面向方面的程序设计(Aspect-Oriented Programming, AOP)是在继承 OOP 基础之上进行改进而来的一种创新的软件设计方法,通过引入方面概念,将应用程序中的核心业务逻辑同其提供支持的通用服务分离开来,从而降低代码之间的耦合度,增加程序的灵活性和可重用性。通过简介 AOP 在事务处理中的应用,介绍了 Spring 框架和 AOP 相结合的事务处理方法。通过采用 Spring 的 AOP 框架,使系统的核心业务逻辑和事务管理逻辑分开,从而系统的耦合性大大降低且易于扩展,面向方面编程提高了系统的模块性。

**关键词:**AOP; OOP; Web 系统; 事务管理; 横切关注点

**中图分类号:**TP311.52

**文献标识码:**A

**文章编号:**1673-629X(2010)04-0246-04

## Research on Application of Transaction Management of Web System Based on Aspect-Oriented Programming

ZHOU Gai-xia<sup>1</sup>, GAO Shang<sup>1</sup>, YANG Zhen-jiang<sup>2</sup>

(1. Jiangsu University of Science and Technology, Zhenjiang 212003, China;

2. Zhengzhou University, Zhengzhou 451000, China)

**Abstract:** The Aspect-Oriented Programming is an innovative software design method based on OOP, through introducing the aspect concept, separating the core business logic and its universal service support from the application to decrease the code coupling degree and increase the flexibility and reusability of procedure. By presenting AOP in the transaction processing application introduce the method of Spring framework and AOP in resolving the transaction. Through the use of Spring's AOP framework, the core business logic and transaction management logic of the system has been separated, thereby the coupling degree of system is much decreased and it is easy for expansion. Aspect-oriented programming improves the system's modularity.

**Key words:** AOP; OOP; Web system; transaction management; crosscutting concerns

### 0 引言

计算机软件设计的重要原则,就是要清晰分离各种关注点(separation of concerns),然后分而治之,各个击破,最后形成统一的解决方案,AOP实现的就是设计模式所追求的目标——分离关注,也就是将某一通用的需求功能从不相关的类之中分离出来,同时能够使得很多类共享这一个通用的行为,一旦行为发生变化,而不必费力修改很多类,只要修改这个行为就可以了<sup>[1]</sup>。

关注点是指一个特定的目标、概念或者兴趣域。从技术角度,软件系统分别包括核心级和系统级的关

注点<sup>[2]</sup>,核心级关注点是系统要完成的业务功能;系统级关注点是完成核心级关注点所必须的配套设施,这些配套设施通常被认为是整个系统的系统特性,或者是业务功能的功能约束。从本质上讲,AOP是用一种松散耦合的方式来实现独立的关注点,然后组合这些关注点来实现最终系统<sup>[3]</sup>。通过简介 AOP 在事务处理中的应用,利用 Spring 的 AOP 框架可以在不使用 EJB CMT 的情况下采用申明式事务处理,为普通的 Java 类提供事务处理的能力,从而简化在 Web 系统中事务处理的方式。

### 1 传统的事务管理实现及问题

#### 1.1 事务的概念

事务是一组原子操作的工作单元,以数据库存取实例来说,就是一组 SQL 指令,这一组 SQL 指令必须全部执行成功,若因为某个原因未全部执行成功(例

收稿日期:2009-07-15;修回日期:2009-10-15

基金项目:江苏省自然科学基金研究课题(08KJB520003)

作者简介:周改霞(1982-),女,河南郑州人,硕士研究生,主要研究方向为计算机软件与设计;高尚,副教授,硕士生导师,主要研究方向为蚁群算法、图像处理。

如其中一行 SQL 错误),则先前所执行的 SQL 指令都会被撤消<sup>[4]</sup>。事务有以下四个特性<sup>[5]</sup>:

(1)原子性:事务必须成功提交,或者回滚。

(2)一致性:事务操作将导致一致性的结果,即使得其操作的资源处于一致性的状态。

(3)隔离性:在事务执行过程中,其操作资源的状态信息不能够被其他事务操作到,因此,不同事务不能够并发对同一数据资源进行读取或写入操作,否则保证不了隔离性。

(4)持久性:一旦事务成功提交,则事务所导致的结果应该是持久性的。

## 1.2 传统的事务管理及问题

通常,对于应用 J2EE 的 Web 应用程序而言,J2EE 应用服务器支持 JDBC 事务,JTA(Java Transaction API)事务(一般由容器来进行管理)和 JTS 事务。

JDBC:JDBC 是一种可用于执行 SQL 语句的 Java API (Application Programming Interface, 应用程序设计接口)。JDBC 给数据库应用开发人员、数据库前台工具开发人员提供了一种标准统一的数据库设计接口,使开发人员可以用纯 Java 语言编写完整的数据库应用程序。但使用 JDBC 连接事务管理的代码不能用于全局的 JTA 事务中。另一个缺点是局部事务趋向于入侵式的编程模型。

JTA:Java 事务 API(JTA)及其同门兄弟 Java 事务服务(Java Transaction Service JTS) 为 J2EE 平台提供了分布式事务服务。一个分布式的事务涉及一个事务管理器和一个或者多个资源管理器。一个资源管理器是任何类型的持久性的数据存储器。事务管理器负责协调所有事务参与者之间的通信。

JTS: JTS 是一个组件事务监视器 CTM (Component Transaction Monitor), 是底层服务提供者的一层接口。它的功能由应用程序通过 JTA 访问。JTA 则由组件开发者和应用程序开发者用来计划性地控制事务边界。

JTA 和 JTS 通常只能在 J2EE 应用服务器环境中才能使用,使用全局事务的首选方式是通过 EJB CMT。其重大的缺陷是 CMT 绑定在 JTA 和应用服务器环境上,并且只有选择使用 EJB 实现业务逻辑,或者至少处于一个事务化 EJB 的外观(Facade)后才能使用它。

## 2 AOP 技术

### 2.1 AOP 的基本思想和基本概念

面向切面编程(AOP)<sup>[6]</sup>是施乐公司帕洛阿尔托研

究中心(Xerox PARC)在 20 世纪 90 年代发明的一种编程范式,它使开发者更好地将那些本不应该纠缠在一起的任务(横向关注点)分离开,从而为程序提供了更好的封装性和互操作性。其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<sup>[7]</sup>,如图 1 所示。

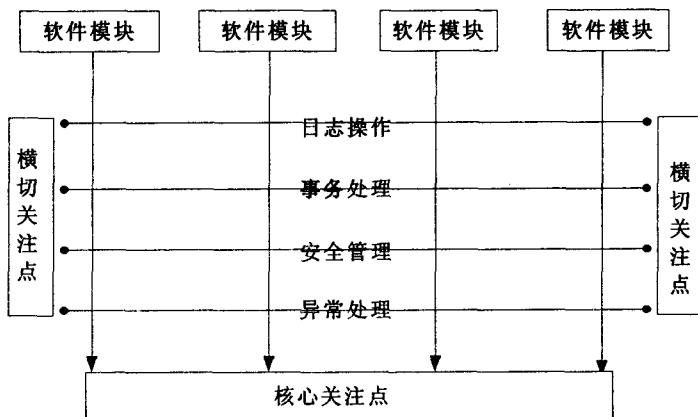


图 1 关注点的分离

对 AOP 的基本概念解释如下<sup>[8]</sup>:

(1)联结点(Joint Point):一个联结点是程序执行过程中的一个特定点,如调用一个方法,类初始化、对象初始化等。

(2)通知(advice):在某一个特定的联结点处运行的代码称为通知。

(3)切入点(Point Cut):用来定义某一个通知该何时执行的一组联结点。

(4)方面(Aspect):通知和切入点的组合,定义了一段程序中应该包括的逻辑,以及何时应该执行该逻辑。

(5) 植入(Weaving): 将方面真正加入程序代码的过程。

(6)引入(Introduction):通过引入,可以在一个对象中加入新的方法或属性,以改变它的结构。这样,即使该对象的类没有实现某一个接口,也可以修改它,使之实现该接口。

## 2.2 Spring AOP 在事务处理中的优势

据 AOSD (Aspect - Oriented Software Development) 统计, 目前基于 Java 的 AOP 实现机制有 20 多种<sup>[9]</sup>, Spring AOP 是其中之一, Spring AOP 与 Aspect J 和 JBoss 具有很大的不同, 首先, Spring AOP 针对的是方法层次上的实现, 而其他两者对字段也提供了支持。

利用 Spring AOP 优势主要在于:

(1) 不像 EJB CMT 绑定在 JTA 上, Spring 声明式事务管理可以在任何环境下使用。只需更改配置文件, 它就可以和 JDBC、JDO、Hibernate 或其他的事务机

制一起工作。

(2)Spring 的声明式事务管理可以被应用到任何类上,不仅仅是像 EJB 那样的特殊类。

(3)Spring 提供了声明式的回滚规则,EJB 没有对应的特性。

(4)Spring 允许你通过 AOP 定制事务行为。你可以在事务回滚中插入定制的行为。你也可以增加任意的通知,就像事务通知一样。

行事务方面的管理,否则将造成难以预料的后果,在该项目的 AOP 实现中利用 xml 配置文件进行配置,因此对于 Aspect 而言,这些操作都是相同的,只要定义了配置文件,将其读入即可,从而将大大提高系统的可扩展性和可维护性。

下面以 Hibernate 事务管理为例,用户在进行技术资料的添加、修改、删除操作时,数据持久层工作方式,如图 3 所示。

### 3 AOP 在 Web 系统事务管理中的应用

以一个舰船的后勤保障计算机辅助管理系统的技术资料模块作为实例,基于 AOP 的思想,采用 J2EE 技术和 JSF、Spring、Hibernate 框架并结合 MVC 等设计模式进行系统设计和研究探索,从事务处理方面挖掘 AOP 的优势,力图更好地解决需求变更所带来的系统开销和耦合度等问题。

下面仅以该模块的添加、修改、删除技术手册目录为例,介绍 AOP 在系统活动流程中事务处理方面的优势,在用户输入账号和密码都正确的情况下,允许合法用户进入技术资料管理模块,进行对技术资料的添加、修改和删除工作,如图 2 所示。

图 2 中的横切面定义为事务管理,因为合法用户在对技术资料模块进行添加、修改、删除操作之前,为了保护数据的一致性,支持多用户并发访问,使用户可以可靠地更新和查询数据库,都会对所操作的数据进行保护,即事务方面的管理。因此横切面把事务管理从流程图中分离开来,在方法操作之前设置好触发点,对所操作的方法进行事务管理,以保护数据的完整性、一致性。

图 2 中的横切面即横切关注点定义把事务处理从流程图中分离出来,用户在操作底层数据库进行添加、修改、删除技术手册等方法操作时都要设置触发点进

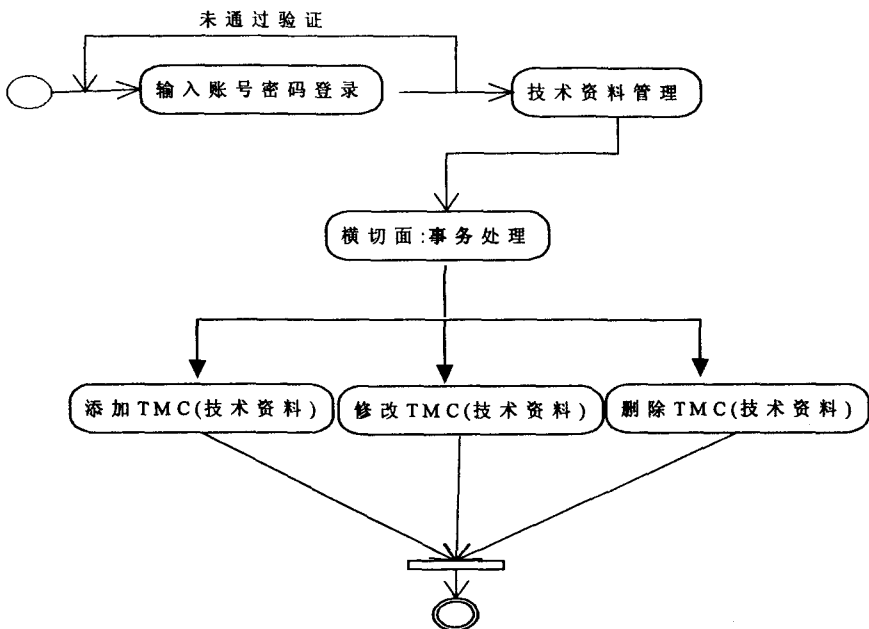


图 2 模块流程图

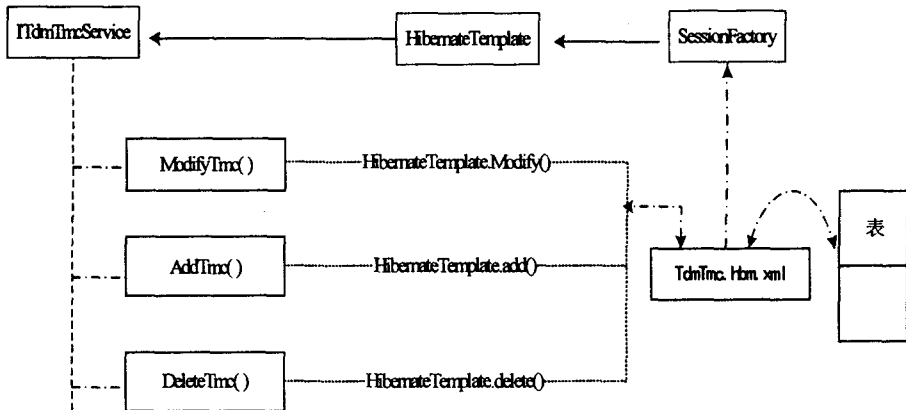


图 3 持久层工作方式

定义配置文件如下所示:

```

<bean id="transactionManager"
class="org.springframework.orm.hibernate3.Hi-
bernateTransaction Manager">
<property name="sessionFactory">
<ref bean="sessionFactory"/>
</property>
</bean>
<tx:advice id="txAdvice" transaction-manager

```

```

=>transactionManager">
  <tx:attributes>
    <tx:method name="add" propagation="PROPA-
GATION_REQUIRED"/>
    <tx:method name="modify" propagation="
PROPAGATION_REQUIRED"/>
    <tx:method name="delete" propagation="
PROPAGATION_REQUIRED"/>
  </tx:attributes>
  </tx:advice>
</aop:config>
<aop:pointcut id="tdmtmcPointcut" expression="
execution(* com.service.tdm.ITdmTmcService.*
(..))"/>
<aop:advisor advice-ref="txAdvice" pointcut-ref="
tdmtmcPointcut"/>
</aop:config>

```

上面的配置将由‘transactionManager’定义的 bean 创建一个代理对象,要把这个代理对象做成事务性的,所以当它的相应方法被调用时,一个事务将被启动、挂起、被标记为只读,或者其它(根据该方法所配置的事务语义)。要把一个服务对象(‘transactionManager’ bean)做成事务性的,将需要施加的事务语义封装在<tx:advice/>定义中,其中如果‘transactionManager’属性指向的 bean 的名字仍为‘transactionManager’的话,事务通知<tx:advice/>中的‘transactionManager’属性可以忽略,否则你则需明确指定。该 bean 将实际上实施事务管理。上例<tx:advice/>把所有以‘add’,‘modify’,‘delete’开头的方法执行在默认语义的事务上下文中,指定应用程序的 propagation 用来设置事务的传播行为。配置中最后一段是<aop:config>的定义,它确保由‘txAdvice’ bean 定义的事务通知在应用中合适的点被执行。首先定义了一个切面,它匹配 transactionManager 定义的所有操作,把该切面叫做‘tdmtmcPointcut’。然后用一个通知器(advisor)把这个切面与‘txAdvice’绑定在一起,表示当‘tdmtmcPointcut’执行时,‘txAdvice’定义

的通知逻辑将被执行。

#### 4 结束语

AOP 是一种新的开发方式,通过上述模块的演示,足以说明传统的事务处理方式和利用 Spring AOP 事务处理方式的不同,但它不是一种要取代传统编程方法的技术,而是对传统编程方法的有力补充,和传统编程技术相辅相成,虽然目前 AOP 仍处于发展阶段,但已经出现了不少的实际应用和学术研究项目, AOP 的发展必将为软件开发带来历史性的变革。

该文的创新点:将 Spring 框架提供的 AOP 技术与 Hibernate 框架相结合,实现了事务处理,并应用于轻量级 Web 系统的开发之中,给出了 Spring AOP 在 Web 系统中开发的具体应用实现,从而提高了系统的可扩展性、可维护性,并降低了系统的耦合度。

#### 参考文献:

- [1] 王蓉,周骏,周维,等. Spring AOP 技术在轻量级 Web 系统中的应用[J]. 微计算机信息, 2007, 9(3): 231 - 232.
- [2] 余维. 基于 AOP 的 MIS 系统设计与实现[J]. 郑州大学学报: 理学版, 2008, 40(2): 67 - 68.
- [3] Johnson R, Hoeller J. Expert One-on-One J2EE Development without EJB[M]. Indianapolis, Indiana: Wiley Publishing, Inc, 2004.
- [4] 林信良. Spring 2.0 技术手册[M]. 北京: 电子工业出版社, 2007.
- [5] 罗时飞. 精通 Spring[M]. 北京: 电子工业出版社, 2006.
- [6] Walker R J, Baniassad E L A, Murphy G C. An Initial Assessment of Aspect-oriented Programming[C]//In: Proceedings of the 21st International Conference on Software Engineering. Los Angeles, CA, USA: [s. n.], 1999.
- [7] Johnson R. J2EE Development Without EJB[M]. Indianapolis: Wiley Publishing, 2004: 144 - 184.
- [8] Filman R E. 面向方面的软件开发[M]. 莫倩, 译. 北京: 机械工业出版社, 2006.
- [9] 杨帆. 基于 AOP 技术的 Web 应用开发与研究[J]. 软件导刊, 2008, 7(4): 60 - 62.

中国计算机学会会刊、中国科技核心期刊

《计算机技术与发展》欢迎订阅, 邮发代号: 52-127