文章编号:1000-582X(2007)09-0071-05

一种基于访问安全性MVC的Web系统开发模式

陈渝侠[a],刘 胜[b],司慧萍[b],刘 飞[b]
(重庆大学 a.软件工程学院;b.机械工程学院,重庆 400030)

摘 要:针对基于MVC的Web应用系统由于访问安全性代码和其他模块强耦合导致系统可维护性差、可重用性弱等问题,提出了一种适用于Web系统的基于访问安全性MVC的开发模式。给出了该模式的总体架构和运行流程,对实现该模式的关键技术进行了深入研究,并将该模式应用于企业网络化分销管理系统的实际开发中。

关键词:访问安全性;Web系统;MVC;拦截器
中图分类号:TP311 **文献标志码**:A

在Web应用系统中,存在代码有可重用程度低、维护工作繁琐以及程序应变能力较弱等不足。采用MVC模式的体系架构设计能有效的解决这些问题并使系统结构清晰,提高了软件的灵活性和复用性[1],但该模式在访问安全性方面欠缺必要的考虑,其访问安全性代码与其他功能代码的紧密耦合导致了整体代码的可维护性、可重用性降低,达不到快速响应企业需求变化的要求[2]。因此文中探索了一种基于访问安全性MVC的Web系统开发模式,通过分离访问安全性的控制代码,实现系统体系结构的清晰可扩展。

1 MVC模式及存在问题

随着Web系统规模和复杂程度的不断增大,给系统开发带来了许多新的问题。一方面,代码的藕合度增大,代码的可维护性、应用框架及组件的可重用性等都在降低;另一方面,对复杂系统难以实现模块化,不利于开发人员之间的分工合作。基于MVC的体系架构能有效的解决这些问题。MVC是Model-View-Controller的简称,即模型-视图-控制器。MVC包括3类对象:模型Model是应用对象,用于存储状态及更新视图;视图View是模型数据的表现形式;控制器Controller抽象用户的交互和应用的语义映射,传递用户输入给应用程序,根据用户的输入和上下文信息选择适当的视图显示数据[1]。通过将应用程序的输入、处理和输出合理的分离,有利于团队分工合作,实现代码的可重用,有利于整个项目的管理和维护。

虽然MVC模式有很多优点,但是由于MVC在访问安全性方面欠缺必要的考虑,造成访问安全性控制部分往往和控制器或者模型紧密耦合在一起,使得系统存在如下两个问题。

1)代码交织(CodeTangling):软件系统的模块可能同时与数个需求交互,如此多的需求会导致最终的代码混乱。

2)代码分散(Codescattering):由于贯穿特性遍布于多个模块的性质,使得相关的实现也遍布于其中。

由于这两个问题的存在,导致模块耦合度高、代码重用率低、质量不高等问题,从而使得开发出来的程序难于维护和扩展。

2 基于访问安全性的MVC模式

基于访问安全性的MVC模式体系架构如图1所示,它由安全、模型、视图和控制器组成。该模式明确的将访问安全控制和其他模块分开,降低了层与层之间的耦合度,从而使得逻辑结构更为清晰。如果安全策略有所改变时,只需要修改安全中的资源权限定义,而无需改动核心业务模块;同样,如果业务需求发生变

收稿日期:2007-04-23
基金项目:重庆市重大科技攻关项目(AA2002)
作者简介:陈渝侠(1980-),男,重庆大学硕士研究生,主要从事企业信息化研究,(Tel)023-60639528;(E-mail)cyx2005@163.com。

化,也只需要关注与业务流程相关的部分即可;另外,当加入新的业务时,编程人员不必考虑新的安全问题,而只需添加相应的资源权限定义。这种体系结构使得系统可以灵活适应业务和安全策略的变化,因此系统的可重用性和可维护性大大提高。

基于访问安全性的MVC模式应用于Web应用程序,其整个流程如下:当Web客户端向服务器提交请求时,服务器端的安全(Security)判断客户端请求是否受到保护。如果受到保护,则对这些提交请求的发起者进行审查。如果审查通过,调用需要的控制器(Controller)。这个控制器(Controller)根据提交的业务,向相应的模型(Model)中的业务方法发出调用请求,安全(Security)判断所调用的业务方法是否受到保护,如果受到保护,则对调用请求的发起者进行审查。如果审查通过,调用所需要模型(Model)的业务方法。然后该模型(Model)将业务的处理结果再传递给视图(View),视图(View)在服务器上处理之后以HTML的方式回显给客户端。

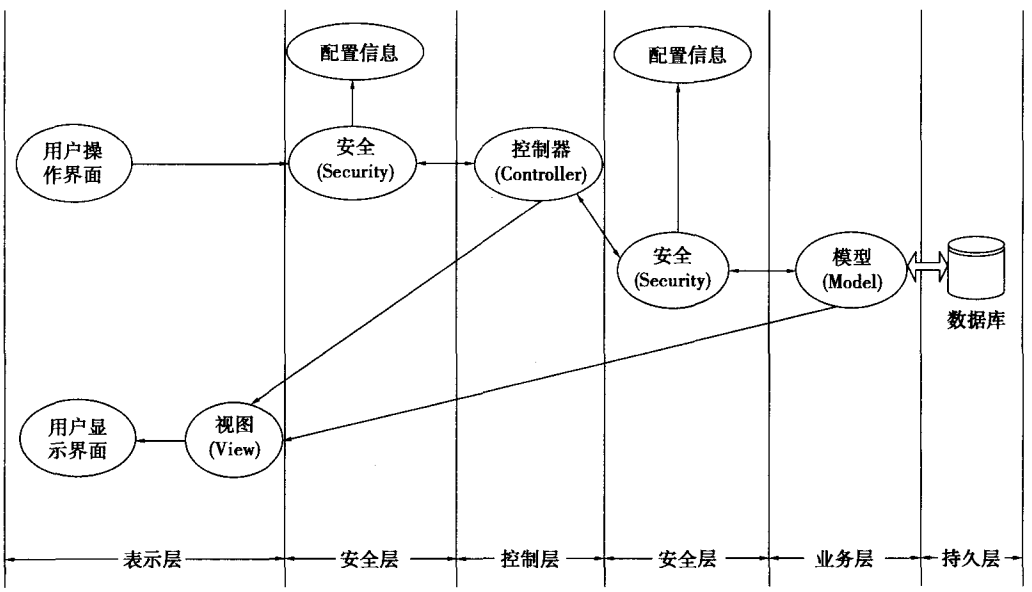

图1 基于访问安全性MVC体系架构示意图

3 实现基于访问安全性MVC的关键技术

基于访问安全性的MVC模式为Web系统提供以下几个特性:1)访问安全性代码与其他功能模块完全分离;2)不仅能够为Web系统提供URL资源请求的安全管理,而且也能够通过保护方法调用在更底层的级别上强制系统访问安全性。3)在保障Web系统访问安全的同时,也保证系统的访问速度。

该模式重点是突出访问安全性代码和其他模块的分离,以拦截器思想[3]为理论依据,设计Web请求过滤拦截器与方法过滤拦截器两个关键组件,这2个组件实现了前两条特性,而资源权限定义和用户权限信息则以缓存的形式存储保证了第3点系统的访问速度特性。针对这些重要特性,在J2EE平台下对这两个关键组件和缓存技术进行研究。

3.1 Web请求过滤拦截器

Web请求过滤拦截器设计中的一个重要问题是如何在Web请求到达控制器之前将其拦截,并且Web请求过滤拦截器要完全独立于控制器。基于这种考虑选择了java的Servlet过滤器来实现该组件。Servlet过滤器是小型的Web组件,它拦截请求和响应,以便查看、提取或以某种方式操作正在客户机与服务器之间交换的数据。通过提供一种面向对象的模块化机制,用以将公共任务封装到可插入的组件中,这些组件的实现只需要在Web.xml文件中配置如下内容[4]:

```
<filter>
<filter-name>HTT PRequest Security Filter</filter-name>
<filter-class>security. FilterInterceptor </filter-class>
</filter>
<filter-mapping>
<filter-name>HTT PRequest Security Filter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>
```

Web请求过滤拦截器使用Servlet过滤器获得

Web 请求,并根据资源权限定义和用户权限信息按照权限验证规则对用户权限验证,完成对 Web 请求的权限控制(见图 2)。

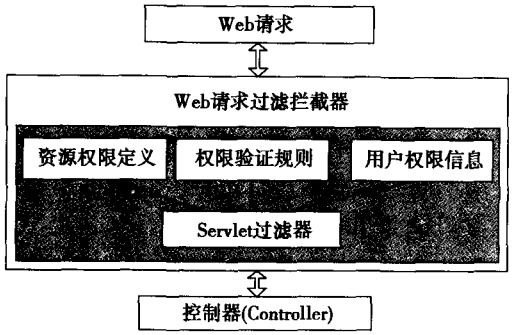

图 2　Web 请求过滤拦截器结构图

组件中定义了 Web 请求的走向,针对 Web 请求的访问安全就可以通过该组件进行控制,彻底把安全控制代码从控制器中分离出来。如果用户的安全需求发生变动,只需要修改资源权限定义,从而大大提高了系统的可维护性、可扩展性和可重用性。组件根据用户权限自动调度合适的控制器,并由控制器返回用户操作的界面,使用户操作界面变的更加友好。

3.2　方法过滤拦截器

方法过滤拦截器的目的是方法级的权限控制,通过采用面向方面编程 AOP(Aspect Oriented Programming)技术,将无法封装在单个模块中的功能与行为,集中于一个横切面上,然后在这个横切面上进行代码的编写。编写完成之后,通过织入机制(Weaving)将这个横切面引入到系统的业务功能模块中,使得业务功能的实现都需要通过该横切面,从而在实现业务功能之前或者之后,也同时实现了该横切面所提供的功能和行为。以 Spring AOP 为例实现方法拦截的配置文件[5]如下所示:

```
< bean id = " bean " class = " org. springframework. aop.
framework. ProxyFactoryBean" >
<property name = "proxyInterfaces" >
<value > dbdao. IndentDAOImp </value >
</property >
<property name = "target" >
<ref local = "beanTarget"/ >
</property >
<property name = "interceptorNames" >
<list > <value > theAdvisor </value > </list >
</property >
</bean >
<bean id = "beanTarget"
class = " dbdao. IndentDAO"/ >
<bean id = "theAdvisor"
class = " org. springframework. aop. support.
RegexpMethodPointcutAdvisor" >
<property name = "advice" >
<ref local = "theBeforeAdvice"/ >
</property >
<property name = "pattern" >
<value > dbdao。IndentDAOImp。
theMethod
</value >
</property >
</bean >
<bean id = "theBeforeAdvice"
class = " test. TestBeforeAdvice"/ >
```

按照上述的配置文件,类 TestBeforeAdvice 可以实现对 IndentDAOImp 类的 theMethod 方法的拦截,在实际项目中需要拦截的方法和对应的权限可以存放在数据库中,以便进行快速的权限修改。

该组件拦截控制器向模型的调度请求并判断用户权限,如果满足用户权限,则调度需要的模型,不满足则调度特定的控制器,返回用户权限不足的操作界面。组件基本结构和 Web 请求过滤拦截器相同,差别在于 Web 请求过滤拦截器是通过 Servlet 过滤器对 Web 请求拦截,而方法过滤拦截器是通过 AOP 实现对方法调用请求进行拦截。该组件应用 AOP 技术彻底地把安全控制代码从模型中分离出来,如果企业的业务流程发生变动,只需要修改资源权限定义,对业务组件复用也更加容易,大大提高了系统的灵活性,增强了系统的快速响应能力。

3.3　缓存信息

如果用户每次请求受保护资源时,都从数据库查询资源权限定义和用户权限信息,会明显降低系统的访问速度。为解决此问题,采用开源项目 Ehcache 来实现缓存用户信息。Ehcache 是一个简单快速的针对 Java 的缓存解决方案,可作为进程范围内的缓存,存储数据的物理介质可以是内存或硬盘。

4　应用实例

根据上述对基于访问安全性 MVC 模式的研究,笔者将其应用到重庆市某日用消费品企业的网络化分销管理系统中。网络化分销管理系统是指利用先进制造技术和 IT 技术对分销系统进行管理的信息系统。随着市场竞争的加剧,企业的业务流程与业务范围在不断的变化,因此企业要求系统能快速响应这些需求变

化,降低企业在信息化方面的投资及其风险[6]。下面以该模式在企业现有订单处理流程中的应用为例对该模式进行说明。图3标识了该模式实现订单处理流程的类间关系图:

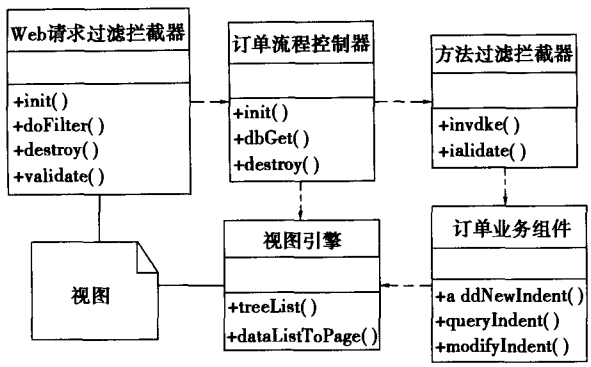

图3 基于访问安全性MVC类间关系图

1)订单流程控制器:作为一个订单流程的控制程序,负责接收来自客户端的订单处理请求,调用订单业务组件进行新增、查询或维护订单数据。

2)订单业务组件:它接受订单流程控制器的控制,负责处理订单业务逻辑的简单JavaBean。

3)视图引擎:负责生成订单处理结果的响应页面。

4)Web请求过滤拦截器和方法过滤拦截器:它们用来实现安全控制请求的拦截,针对不同的安全控制请求使用不同的拦截器进行拦截。它们都封装了各自的资源权限定义和权限验证规则。

Web请求过滤拦截器:遵循Servlet过滤器设计规范。实现了强制拦截Web请求,并对Web请求的发起者进行权限验证。

方法过滤拦截器:借助于spring AOP,分离了访问安全性代码,对订单业务组件的方法提供安全保证。

图4所示的顺序图表现了在新增订单过程中该模式各个应用组件之间的交互关系。

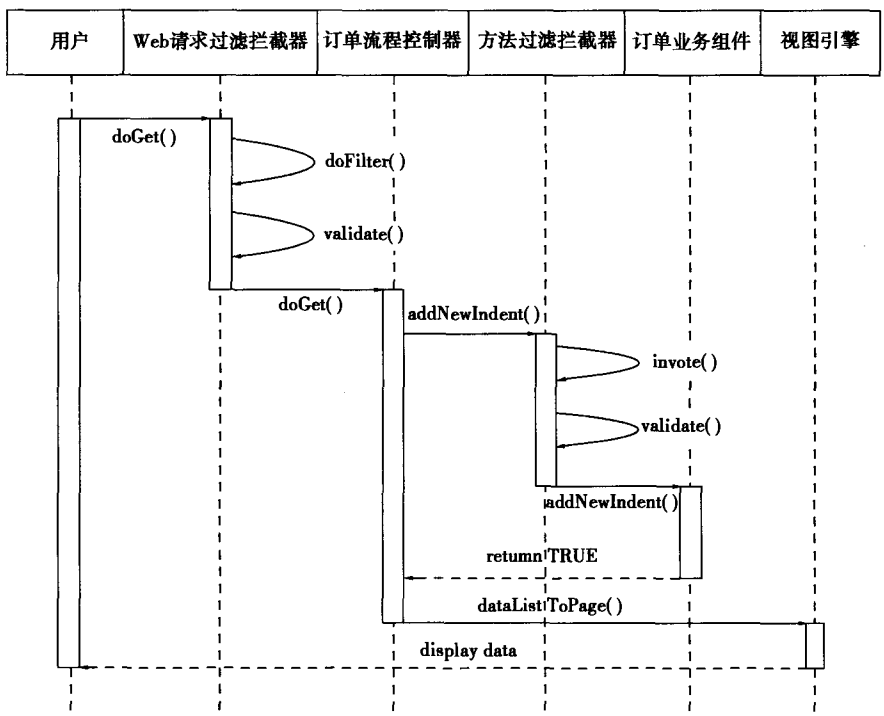

图4 新增订单顺序图

1)用户提交新的订单数据,客户端向订单流程控制器发出调用请求。

2)Web请求过滤拦截器调用doFilter()拦截客户端操作请求并判断客户端请求是否受到保护。如果受到保护,调用validate()对用户进行权限验证。如果通过权限验证,则执行客户端请求操作,否则返回用户权限不足信息。

3)订单流程控制器接受到客户端请求后,调用订单业务组件的addNewIndent()方法。

4)方法过滤拦截器调用invoke()拦截控制器的调用请求并判断addNewIndent()是否受到保护,判断权限流程和Web请求过滤拦截没有区别。

5)订单业务组件的addNewIndent()方法完成业务逻辑后,把结果传送给视图引擎,生成客户显示

页面。

目前,基于访问安全性的MVC模式已经在5家快速消费品企业的网络化分销管理系统中成功应用,大大提高了这些系统的可重用性和可维护性。

5 结 论

在基于MVC模式的基础上,从访问安全性角度提出了基于访问安全性的MVC模式,研究了实现该模式的关键技术。并通过实例说明了该模式的具体实现。实践证明,该模式通过分离访问安全性代码,降低了层与层之间的耦合度,使得系统具有更好的可维护性和可扩充性。在构建不同的Web系统时该模式也具有良好的可重用性,能有效缩短开发周期,降低开发成本。

参考文献:

[1] 姚慧广,赵岳松.Web编程中MVC模型的应用[J].微机发展,2002,11(3):9-10.

[2] 吴春明,强保华,余建桥.基于AOP的安全程序设计[J].重庆大学学报:自然科学版,2004,27(11):39-42.

[3] 唐玲.CORBA拦截器机制的研究[J].科技广场,2005(12):16-18.

[4] 孙卫琴.Tomcat与Java,Web开发技术详解[M].北京:电子工业出版社,2004.

[5] WALLS C,BREDIENBACH R. Spring in action[M].北京:人民邮电出版社,2006.

[6] 刘胜,王诚,郭亮,等.基于虚拟分公司模式的销售管理系统开发[J].重庆大学学报:自然科学版,2005,28(6):15-18.

Web System Development Model Based on Access Security MVC

CHEN Yu-xia[a], *LIU Sheng*[b], *SI Hui-ping*[b], *LIU Fei*[b]

(a. College of Software Engineering;

b. College of Mechanical Engineering, Chongqing University, Chongqing 400030, China)

Abstract: In order to reduce the inconvenience of maintainability and reusability of web application system based on MVC, which is brought by the code of access security and coupling tightly with other modules, A web system development model based on access security MVC is put forward. The main framework and running process are provided and the key techniques of this model are researched. The model is applied to the practical development of the network distribution management system.

Key words: access security; Web system; MVC; interceptor

(编辑　陈移峰)