WEB工作流的访问控制研究与实现

郑文煜,葛 玮
(西北大学 计算机科学系 软件工程研究所,陕西 西安 710069)

摘 要:由于WEB技术本身所具有的分布式、广域访问的特点以及企业现在普遍比以往更加重视自身的信息安全问题,在基于WEB的工作流的设计和实现过程中,访问控制问题成为一个至关重要的议题。文中从一个企业级的WEB工作流系统基础出发,探讨在一个典型的基于WEB的工作流应用系统中,如何对来自广域环境下的用户访问进行有效的控制并提供安全的授权机制。同时能提供与企业组织结构相适应的动态的角色权限分配管理机制。

关键词:工作流;访问控制;JAAS;J2EE
中图分类号:TP393.08 **文献标识码**:A **文章编号**:1005-3751(2006)02-0235-03

Research and Realization on Access Control of WEB Workflow

ZHENG Wen-yu, GE Wei
(Inst. of Software Eng., Dept. of Computer Science, Northwest University, Xi'an 710069, China)

Abstract: Because Web technology has property of distributed, wide area access, and more and more enterprise focus on information security and efficiency access control architecture. So during the design and implementation of Web-based workflow application, access control and authority technology becoming important issue. This paper introduces a solution of access control architecture for Web-based workflow application, and gives an implementation of this architecture.

Key words: workflow; access control; JAAS; J2EE

1 基于WEB的工作流

工作流管理联盟(WorkflowManagementCoalition, WfMC)给出的工作流定义是:工作流是一类能够完全或者部分自动执行的经营过程,它根据一系列过程规则、文档、信息或任务能够在不同的执行者之间进行传递与执行。由于网络技术的飞速发展和快速普及,WEB已经成为信息交流和共享的一个基础性平台。WEB应用程序的开放、跨平台的特性使基于WEB的管理和协同工作系统成为一种必然的发展趋势。但是,由于基于WEB的工作流应用系统的运行环境是Internet,因此在局域网环境下容易解决或不存在的诸如安全性、共享性、私有性、防火墙等问题显现出来[1,2]。同时由于工作流管理本身的特点,人们也希望通过有效的访问控制机制来屏蔽WEB操作用户权限之外的操作项,将系统资源和数据对象共同作为对象来制定权限控制策略,在页面中动态生成用户的可操作菜单,一方面屏蔽了用户无权访问的系统资源,防止非法用户访问资源和合法用户越权使用资源;另一方面还使得应用系统的开发和维护更加方便而快捷。这就要求:

①每个登陆用户都要经过严格的身份认证。

②根据登陆用户的身份信息动态生成该用户权限范围。在页面上就是不同的登陆用户可操作的菜单项不同。这就要求对于权限认证控制的粒度要小。

③能够对用户试图绕过访问控制机制的企图进行识别和屏蔽。由于使用了细粒度的控制策略,就可以防止绕过认证机制而使用URL直接非法访问系统资源。

图1是一个基于WEB的工作流系统的架构。此处问题集中在其访问控制机制,而不去管其工作流引擎和工作流引擎外围的工作流运行环境的设计和实现机制。因为从工作流引擎与与业务分离的思想出发,工作流引擎的工作机制对研究这种架构下的访问控制机制来说是透明的。

* 通过在用户访问界面与工作流引擎之间添加Filter及请求预处理层。所有操作项的执行请求或者对于系统资源的访问请求全部被拦截。被拦截的请求都要经过访问控制机制的检查,如果合法才能通过。同时在这一层使用Decorator模式,针对用户的身份信息对于用户所能看到的返回信息也作了相应的过滤和权限控制。这样下一步的操作者只能看到与其权限相符的可操作菜单,从而对他屏蔽了不可访问的资源。由此避免了其有意或者无意访问未授权资源的情形。

* 数据访问层提供了对于底层数据持久层进行访问的统一接口。在这一层可以根据业务的需要进行另外一层的数据访问控制机制。通过采用分散、多层次的访问控制机制,使整个系统的访问控制更加灵活高效。

收稿日期:2005-05-23
作者简介:郑文煜(1980—) 男,陕西宝鸡人,硕士研究生,研究方向为网格计算、工作流、中间件计算;葛 玮,副教授,硕士生导师,主要研究方向为软件工程、分布式计算、工作流技术。

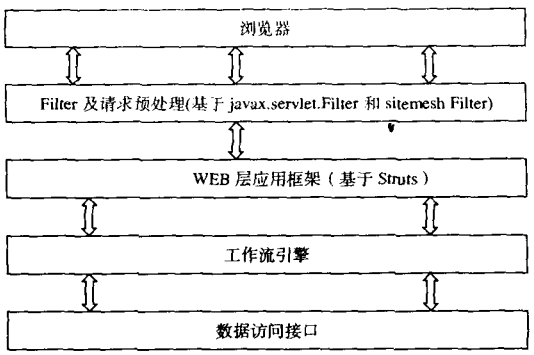

图 1 WEB 工作流应用系统的典型架构

2 基于角色的访问控制

RBAC 的核心思想就是将访问权限与角色关联,通过给用户分配合适的角色,让用户与访问权限联系[3]。角色是根据企业内为完成各种不同的任务需要而设置的,根据用户在企业中的职权和责任来设定他们的角色。用户可以在角色间进行转换,系统可以添加、删除角色,还可以对角色的权限进行添加、删除。这样通过应用 RBAC 将安全性放在一个接近组织结构的自然层面上进行管理[4,5]。

图 2 是基本的 RBAC 模型。

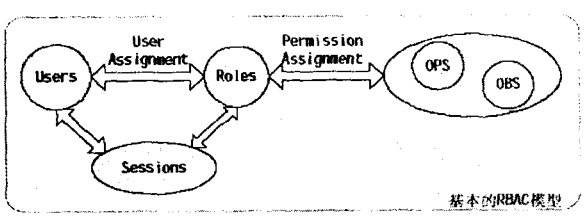

图 2 基本的 RBAC 模型

在该模型中,用户和角色之间、角色和权限之间均为多对多的关系,即:同一用户可以有多种角色,同一角色可以赋予多个用户;同一角色可对多个资源有访问权限,同一资源可赋权给多个角色。级别高的角色可继承低级角色的权限。角色和角色之间、权限和权限之间有限制关系。当用户有多个角色而这些角色又分属不同会话组时,只可激活一个会话,从而控制角色冲突[1]。定义角色集合为 $R: r_1, r_2, r_3, \cdots, r_l$,用户集合为 $U: u_1, u_2, u_3, \cdots, u_m$,权限集合为 $P: p_1, p_2, p_3, \cdots, p_n$,角色与权限的关系集合用($R \propto P$)表示,用户与角色的关系集合用($U \propto R$)表示,用户与权限的关系集合用($U \propto P$)表示。某个用户 u_i 的角色集是集合($U \propto R$)的一个子集 ($u_i \propto R_i \mid u_i \in U$),某个角色 r_i 的权限集是集合($R \propto P$)的一个子集($r_i \propto P_i \mid r \in R$),($U \propto P$)集合中的元素则通过集合($U \propto R$)和集合($R \propto P$)中的元素间接确定[2,4,6]。因此,只要能建立关系集合($U \propto R$)和($R \propto P$),则关系集合($U \propto P$)中的元素就可确立,即用户的权限可确立。

3 实现策略

由于在基于 WEB 的工作流系统中,其访问控制机制不仅仅是实现用户登陆的认证和授权的机制,而且还包括动态职责分离、动态职责绑定等问题的控制和管理。此外笔者也期望通过访问控制机制所提供的用户的动态角色信息动态地生成用户的可操作项,表现在具体的 WEB 页面中就是在页面中动态生成用户的可操作菜单。同时又要保证这种动态性得到了有效的控制和管理而且是安全的。图 3 是一个基于 WEB 的访问控制机制的实现框架。

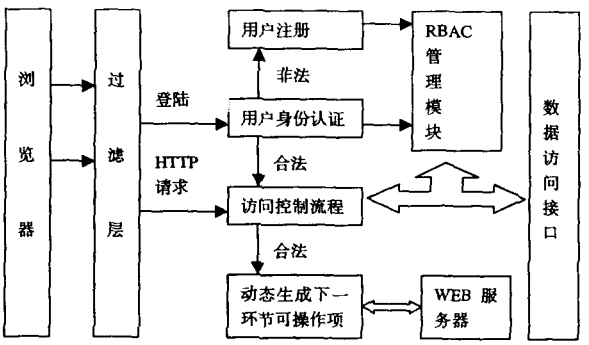

图 3 基于 WEB 的访问控制机制实现框架

在具体的实现中使用了 XML 来作为角色和权限以及角色和权限之间映射关系的配置文件,大大增强系统对于不断变化和调整的组织机构的适应能力,增强了可移植性和可扩展性。而且 XML 易于操作且具有可读性;易于维护和管理。其中,Authorities.xml 定义了系统中可用的所有权限,将每个 authority 定义为一个三元组(name,object,action),表示对于权限 name 来说它对于 object 具有执行 action 的权限。Group - Role - Authority.xml 定义了系统中所有定义的角色,以及角色和其所具有的所有权限的映射关系。通过这两个配置文件建立了集合($R \propto P$),而集合($U \propto R$)关系的建立则是建立用户的时候绑定其所对应的角色,而且该用户的角色可以维护也就是随着组织机构的变迁用户所具有的角色也可以动态变更。通过在之前的分析知道只要能建立关系集合($U \propto R$)和($R \propto P$),则关系集合($U \propto P$)中的元素就可确立,即用户的权限可确立。下面给出它们的两个样例:Authorities.xml,Group - Role - Authority.xml。

3.1 Authorities.xml

```
<? xml version="1.0" encoding="UTF-8"? >
<cudes xmlns:my="http://schemas.microsoft.com/office/infopath/2003/myXSD/2004-11-18T07:10:38" xml:lang="zh-cn">
  <authorities>
    <authority name="nms.show" object="删除操作员" action="del"></authority>
    <authority name="cms.show" object="进入综合管理子系统" action="any"></authority>
    <! -其他的定义- - >
  </authorities>
  <actions>
    <action actionname="del" displayname="删除"></action>
```

```
        <action actionname="any" displayname="所有"></action>
        <!--其他的操作的定义-->
    </actions>
</cudes>
```

3.2 Group-Role-Authority.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<cudes xmlns:my="http://schemas.microsoft.com/office/infopath/2003/myXSD/2004-11-18T09:23:06" xml:lang="zh-cn">
    <admin>
        <role name="系统管理员">
            <authority>person.any</authority>
            <authority>self.any</authority>
            <authority>cms.show</authority>
        </role>
    </admin>
    <company class="0" id="">
        <role name="总公司系统管理员">
            <authority>person.any</authority>
            <authority>self.any</authority>
        </role>
    </company>
    <company class="1" id="">
        <role name="分公司系统管理员">
            <authority>sortclass.any</authority>
            <authority>sortclass.list</authority>
            <authority>person.any</authority>
            <authority>mms.show</authority>
            <authority>cms.show</authority>
        </role>
    </company>
</cudes>
```

3.3 基于JAAS的RBAC管理模块

Java Authentication Authorization Service(JAAS, Java验证和授权API)提供了灵活和可伸缩的机制来保证客户端或服务器端的Java程序的安全。其提供了一个可插拔的(pluggable)和富有弹性的(flexible)框架(framework)允许开发者混合不同的安全机制和丰富的已经存在各种安全方面的资源。在JAAS中,登陆是一个两阶段(two-phase)的处理过程。第一阶段是"登陆(login)"阶段。这个阶段唯一的任务是认证。只要处理过程成功通过这个阶段,认证处理过程就进入了"提交(commit)"阶段,这一阶段LoginModule的commit方法被调用去关联所属子项相关的规则和标识。在JAAS中一个所属子项表示一个认证实体,比如一个人或者一台设备。它包含了一整套法则和安全相关的属性诸如密码和加密密钥。在JAAS体系结构中,所属子项和其所附属的相关权限,在认证过程中扮演了重要的角色。所有的认证模块当中,LoginModule是事实上的认证机制的借口。虽然LoginModule决没有得到直接调用客户应用的机会,但是他经由一个可插拔的模块提供了一个认证的具体类型,其实现了认证的算法并且决定实际的认证过程是怎样被执行的[7]。

在基于JAAS的RBAC管理模块中,包含登陆、授权、存取控制等子功能模块。在登陆模块中提供了定制的认证策略。在登陆用户通过"登陆"阶段后,开始给该用户分配相应的角色。在分配角色后每个登陆用户被映射到一个四元组(Group:Role:Object:Action),表示Group中的角色Role对系统所定义的Object具有执行Action的权利。此时该用户已经具有了初始的登陆权限。但由于在一个工作流应用系统中,一个用户所具有的权限可能是动态变化的,所以RBAC管理模块必须提供伴随工作流的执行能根据某种策略动态地获得用户的动态角色的能力。

3.4 动态生成可操作项

RBAC管理模块不仅负责了初始登陆用户的认证和权限分配,而且能够伴随工作流的执行动态地处理用户的角色变更,所以所有的用户请求在通过访问控制层以后可以认为它是可信赖的。此时要处理该用户的请求并将处理的结果返回。从文中的实现框架可以看到,在返回之前会根据用户动态角色信息和工作流处理流程的相关信息动态生成下一环节可操作项,也就是对用户请求所返回的WEB页面进行过滤,将其权限之外的页面对象屏蔽掉。

4 结束语

在基于WEB的工作流系统中,综合使用基于角色的访问控制机制实现动态地角色绑定已经动态的生成下一步的可操作项。既提供了安全可靠的访问控制机制,同时也使WEB工作流的实现,特别是WEB层的实现大大简化。该实现策略已经在一个企业级的WEB工作流应用系统中成功应用。

参考文献:

[1] 段云所.信息安全概论[M].北京:高等教育出版社,2004.50-80.

[2] 张 纲,李晓林,游赣梅,等.基于角色的信息网格访问控制的研究[J].计算机研究与发展,2002,39(2):953-956.

[3] 蒋 涛.信息安全模型研究[J].小型微型计算机系统,2000,21(10):1080-1081.

[4] 高正宪,李中学.WEB环境下基于角色的访问控制策略及实现[J].计算机工程,2004,30(8):133-134.

[5] 吕宜洪,宋瀚涛,龚圆明.大型应用系统用户权限构成分析及访问控制策略研究[J].小型微型计算机系统,2004,25(2):195-198.

[6] 李孟珂,余祥宣.基于角色的访问控制技术及应用[J].计算机应用研究,2000(10):45-46.

[7] Deitel H M, Deitel P J, Santry S E.高级Java2大学教程[M].钱 方,梅 皓,周 璐,吴志英,等译.北京:电子工业出版社,2003.346-385.