第34卷 第20期 计 算 机 工 程 2008年10月
Vol.34 No.20 Computer Engineering October 2008
·软件技术与数据库· 文章编号：1000—3428(2008)20—0052—03 文献标识码：A 中图分类号：TP311.5

基于RBAC的细粒度访问控制方法

吴江栋[1]，李伟华[2]，安喜锋[2]
(1. 西北工业大学软件与微电子学院，西安 710065；2. 西北工业大学计算机学院，西安 710072)

摘 要：分析基于角色的访问控制模型，提出一种基于RBAC模型的细粒度权限管理方法。引入细粒度权限管理的概念，把资源的访问权限按尽量小的粒度分解，并把分解后的权限分配给角色，通过给用户分配角色以及角色之间的继承关系简化权限的管理。基于RBAC模型的细粒度权限管理系统验证了该方法的正确性。该方法的设计和实现过程对于同类软件的开发具有参考价值。

关键词：角色；访问控制；RBAC模型；细粒度访问控制

Method of Finely Granular Access Control Based on RBAC

WU Jiang-dong[1], LI Wei-hua[2], AN Xi-feng[2]
(1. College of Software and Microelectronics, Northwestern Polytechnical University, Xi'an 710065;
2. School of Computer, Northwestern Polytechnical University, Xi'an 710072)

【Abstract】A method of finely granular access control based on RBAC is brought forward after the discussion of the access control model based on role. This paper proposes the idea about finely granular access control, decomposes the access privilege of sources to less granularity, and the privilege is assigned to role, then access control can be managed easily by defining the user of the role and the inherit of roles. The validity of method is proved by the successful system of finely granular access control based on RBAC. Design and implementation process of the method have referenced value to similar software's development.

【Key words】role; access control; RBAC model; finely granular access control

1 概述

基于Web的管理信息系统成为软件开发的一个方向，其信息安全受到人们的关注。特别是在Browser/Server(B/S)模式的管理信息系统中，强健的权限管理对于保证信息系统的安全性是必需的。权限管理为解决信息系统安全性问题提供了重要保障。

目前的权限管理大致上可以分为两类：

(1)系统级的安全管理

该部分包括：操作系统级的安全管理，数据库级的安全管理等。

(2)应用级的安全管理

该部分权限的控制主要取决于具体的系统。

在应用级的安全管理方面，基于角色的访问控制方法(Role-Based policies Access Contro1, RBAC)是公认的解决大型企业统一资源访问控制的有效方法，特征如下：

(1)减小授权管理的复杂性，降低管理开销。

(2)灵活地支持企业的安全策略，并对企业的变化有很大的伸缩性[1]。

本文结合RBAC授权模型，针对基于J2EE平台的信息系统提出了一种细粒度权限管理方法，并在实际项目中进行了验证。

2 RBAC模型及基本思想

1992年，Ferraiolo等人提出了基于角色的访问控制模型RBAC[1]。在RBAC模型中，权限(permission)直接通过角色(role)起作用，用户(user)被赋予一个或多个角色(role)进而获得这些角色的权限(permission)。这很大程度上简化了权限的管理。在这种权限管理模式下，用户的角色转换变得很容易实现。角色针对组织中的各种单一功能进行创建，用户依据他们的责任和资历被指派一个或多个角色。用户对信息的访问在指派角色的基础上被管制。

1996年，Sandhu[2]发布了RBAC96的RBAC通用模型家族。文献[3]介绍了一些基于角色系统的开发框架。图1显示了家族中最通用的模型。一个用户是一个人或一个独立的个体，一个角色是一项在组织中的工作功能或工作头衔。而权限是对系统中一个或多个对象(object)的特定访问模式的许可或执行某些动作的特权。角色以偏序关系≥组织，如$x \geqslant y$，那么角色x就继承了角色y的权限。x的成员也意味着是y的成员。每次会话(session)把一个用户和可能的许多角色联系起来。用户建立一次会话，激活了用户的角色子集。一个用户可以拥有一个或多个角色，一个角色可以同时赋予多个用户。类似地，一个角色可以有多个权限，同一个权限可以被指派给多个角色。每个会话把一个用户和可能的许多角色联系起来。一个用户在激发本身所属角色的某些子集时，建立了一个会话。用户可用的权限是当前会话激发的所有角色权限的并集。每个会话和单个用户关联。这个关联在会话的

作者简介：吴江栋(1978－)，男，硕士研究生，主研方向：软件工程；李伟华，教授、博士生导师；安喜锋，博士研究生

收稿日期：2007-11-28 **E-mail：**wu-summer@163.com

生命期间保持不变。一个用户在同一时间可以打开多个会话，例如，可以通过多个 IE 同时打开。每个会话可以有多个不同的活动角色。会话的概念相当于传统的访问控制中主体(subject)的标记。一个主体是一个访问控制单位，一个用户在同一时间可以拥有多个不同活动权限的会话。

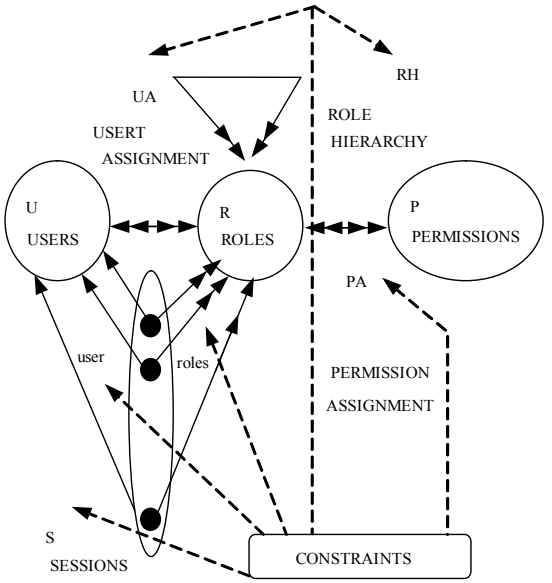

图 1　RBAC 模型

3　细粒度权限管理系统的实现

一个系统的权限控制的最小单位是原子按钮或链接，也就是不可再分解的单一功能的按钮或链接。如果把权限都分解为可以进行访问控制的原子按钮或链接的集合，并按这样的粒度进行权限管理，这样的权限管理就是最小粒度的权限管理。当然，最小粒度是理想化的分解结果，在实际工程中，由于各式各样的原因，这样的分解是很难做到的，因此只能做到尽量小的分解，即比较折中的细粒度分解。

细粒度是相对粗粒度而言的。在粗粒度权限管理系统中，存在如下缺点：

(1)在应用粗粒度权限管理的信息系统研发过程中，由于授权的粗粒度，为了在信息系统中体现原子按钮或链接权限的分配，必须编写大量代码进行控制，其中有许多代码是重复或类似的。

(2)在应用粗粒度权限管理的信息系统使用过程中，由于权限的粗粒度，权限树太宽而不够深，每个权限所表示的意思都比较接近，没有层次性，不利于授权管理。

细粒度权限管理方法通过在传统的粗粒度权限管理基础上加入“菜单-按键”对应关系使权限得以细化，解决了粗粒度权限管理系统存在的缺点。

细粒度权限管理系统分为 3 层：

(1)客户端表示层；

(2)应用服务层，由一台或多台服务器组成，该层具有良好的可扩充性，可以随着应用的需要增加服务器的数目；

(3)数据层，由数据库系统和遗留系统组成。系统的动态权限管理是以数据库作为基础的。

整个系统是基于 B/S 模式的，具体应用生产环境如下：

(1)数据库采用 Orcale 10g；

(2)应用服务器采用 Weblogic v8.0；

(3)Web 服务器层服务器端编程语言采用 Java 语言；

(4)Web 页面端(客户端表示层)使用 HTML(超文本标志语言)和 Javascript 脚本语言。

3.1　数据库的设计

授权系统数据库逻辑结构见图 2。用户与角色不直接相关，而是通过“用户-角色”对应表建立起动态的对应关系。角色与权限也不直接相关，而是通过“角色-菜单-按键”对应表建立动态的对应关系。通过“用户-角色”对应表和“角色-菜单-按键”对应表就可以方便、灵活地建立用户与权限之间的对应关系。

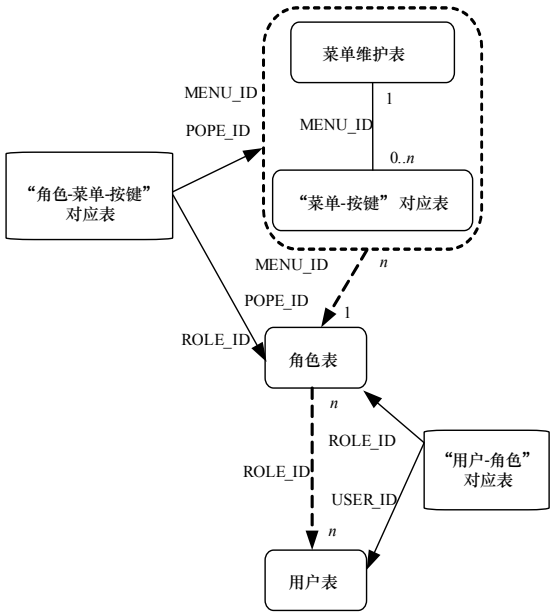

图 2　授权系统数据库逻辑结构

在进行信息维护时，用户表、角色表都是可以独立维护的；菜单维护表和“菜单-按键”对应表的记录组成权限集，这两个表可放在一起维护，有利于系统的操作。

3.2　用户授权的实现

用户授权的实现包括：菜单维护，角色维护和用户维护。其中，这 3 项维护功能所包括的所有权限也是权限管理能进行授权的一部分。也就是说，菜单维护、角色维护和用户维护也是需要相应权限才能进行操作的。

(1)菜单维护

具有菜单维护权限的用户(包括子菜单权限和添加、修改、删除等按键权限)通过菜单维护界面，对信息系统的菜单、按键等权限进行维护，菜单是有层次性的，一级菜单下面可能会有二级菜单、三级菜单或者按键。

(2)角色维护

具有角色维护权限的用户(包括添加、修改、删除和授权等按键权限)通过角色维护界面，维护信息系统得角色信息；并且按实际情况，将信息系统中已经分解为最细粒度的所有独立的权限，按单一任务分组，并分配给相应角色。

(3)用户维护

具有用户维护权限的用户(包括添加、修改、删除和分配角色等按键权限)通过用户维护界面，维护信息系统用户基本信息；并且按实际情况，将信息系统中一个或若干个角色分配给指定的用户。

用户授权静态类图见图 3。

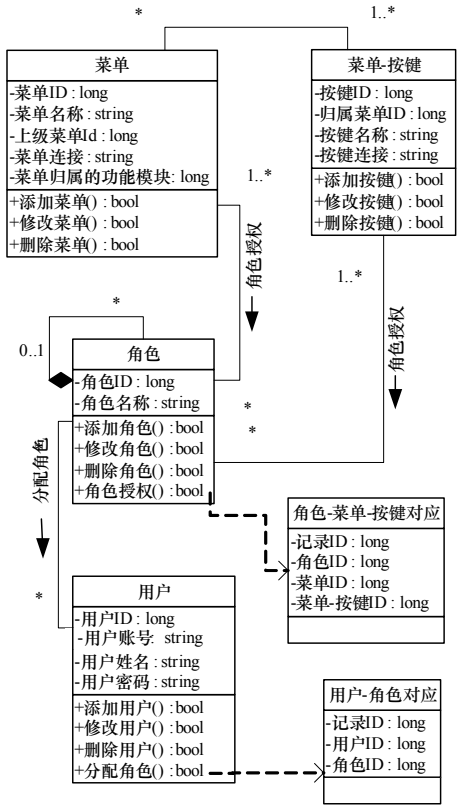

图 3　用户授权静态类图

3.3　用户权限的控制

所谓用户权限就是授予用户的所有角色所具有的权限项的总和。用户具有的所有权限项组成一棵权限树，根为系统名，根的孩子为一级菜单。获得用户权限项的流程如图 4 所示，流程如下：

(1)系统使用者(用户)通过系统登录页面输入账号和口令，通过验证后获得用户 ID 及属于用户的所有角色 ID，并都存入 Session 变量中。

(2)根据属于用户的所有角色 ID“roleids”，获得用户在系统中具有权限的一级菜单，SQL 语句为：

```
Select MENU_ID,MENU_NAME,MENU_URL,
MENU_BELONG from MENUINFO where MENU_UP_ID=1 and
MENU_ID in(select distinct MENU_ID from ROLE_MENU_POPE
where ROLE_ID in (" + roleids + ")) order by menu_Id
```

(3)根据一级菜单号“headmenu”及属于用户的所有角色 ID“roleids”，获得以相应一级菜单为根的用户所有权限项节点，并根据节点间的父子关系组成的菜单树，SQL 语句为：

```
select MENU_ID, MENU_UP_ID, MENU_NAME, MENU_URL
from MENUINFO where MENU_BELONG='" + headmenu + "' and
MENU_ID in(select distinct MENU_ID from ROLE_MENU_POPE
where ROLE_ID in (" + roleids + ")) order by menu_Id
```

(4)根据菜单 ID“menuid”及属于用户的所有角色 ID“roleids”，获得相应页面及用户在本页面下具有权限的所有按键及链接，SQL 语句为：

```
select distinct mp.POPE_NAME from MENU_POPE mp,ROLE_
MENU_POPE  rmp where mp.POPE_ROW_ID= rmp.POPE_ID and
rmp.MENU_ID=" + menuid + " and rmp.ROLE_ID in (" + roleids + ")
```

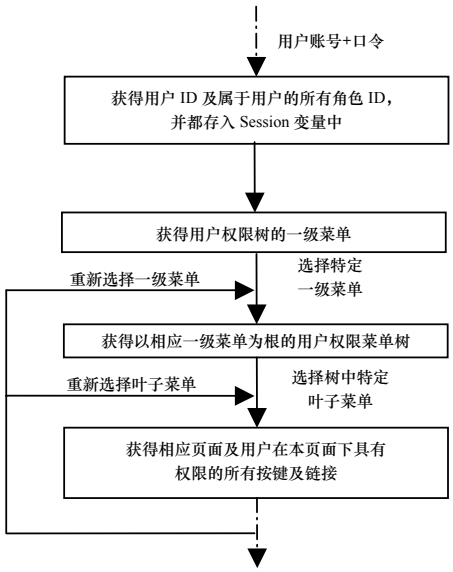

图 4　获得用户权限的流程

另外，对于页面来说，具有相应页面链接权限的用户不一定都具有本页面上所有按键和链接的权限。为了提高开发效率和程序的灵活性，将用户不具有访问权限的按钮或链接设置为不可见，用户具有访问权限的按钮或链接设置为可见，从而更好地实现对权限的细粒度控制。这样做就可以尽量减少系统中实际页面文件的个数。

在教育部学位与研究生教育发展中心使用该方法实现了权限管理系统。在实际使用过程中，用户一致反映，该系统符合学位中心业务处室的实际业务需求，使用灵活方便。

4　结束语

权限管理是信息系统的重要组成部分，权限管理的技术和策略对系统的信息安全影响很大。本文提出的基于 RBAC 的细粒度权限管理方法是在 RBAC96 模型[2]理论基础上提出，并在实际工程中进行了验证，除了满足信息安全性的需要，还降低了权限管理和维护的复杂性，简化了用户的授权。对开发类似的基于 J2EE 的信息系统权限管理功能模块具有一定的参考价值。

参考文献

[1] 周文峰, 尤军考, 何基香. 基于RBAC 模型的权限管理系统设计与实现[J]. 微计算机信息, 2006, 22(3-5): 35-36.

[2] Sandhu R S, Coyne E J, Feinstein H L, et al. Role-based Access Control Models[J]. IEEE Computer, 1996, 29(2): 38-47.

[3] Epstein P, Sandhu R. Towards A UML Based Approach to Role Engineering[C]//Proceedings of the 4th ACM Workshop on Role-based Access Control. [S. l.]: AVM Press, 1999-10: 28-29.