一种基于改进RBAC模型的权限管理系统

文骁一
（厦门大学 信息科学与技术学院 福建 厦门 361005）

摘 要： 针对传统RBAC模型在现今企业应用中的不足，提出一种改进的RBAC模型。该模型引入用户组的概念，减少授权的工作量，在角色授权中利用角色之间的优先级解决不同角色互斥的问题，并且采用基于细粒度的权限管理方式，将资源的控制从菜单级别分解到原子按钮级别。该模型采用ASP.NET技术，集成Spring.Net和NHibernate开源框架，实现一种高复用，低耦合，易维护的权限管理系统。

关键词： RBAC；访问控制；细粒度；Spring.Net
中图分类号： TP393.08 **文献标识码：** A **文章编号：** 1671－7597（2012）0210039－02

0 引言

随着计算机技术的高速发展和普及，web信息系统已经和人们的日常生活、学习、工作紧密相关。但互联网给予人们方便的同时也带来了许多的安全隐患。信息安全随即成为人们讨论的重点。

访问控制（Access Control）技术作为众多信息安全技术[1]中的一种被广泛计算机信息系统中。访问控制技术已经从最早的自主访问控制（DAC）和强制访问控制（MAC）发展为现在普遍流行的基于角色的访问控制（RBAC）。然而RBAC模型也有相应的不足。比如，随着系统科目和数据的增加，角色的数量会越来越大，不便于管理；用户和角色的对应关系取决于系统本身的逻辑，这势必会造成冗余。对于上述问题，国内外学者进行了大量的研究[3]-[6]，但始终没有提出一个最优方案。本文基于ASP.NET技术，并采用Spring.net+NHibernate框架搭建权限管理系统框架，同时在基于原有RBAC的模型上引入了用户组的概念，并且在不同角色之间加入了优先级的概念，解决了不同角色之间权限冲突的问题，同时实现了权限的细粒度控制，可以对菜单页面下一级的功能按钮进行控制。

1 RBAC模型及改进

1.1 传统RBAC模型及其基本思想

RBAC[2]（Role Based Access Control）模型是由NIST组织的Ferraiolo等人在1992年提出来的，它的基本思想是：将用户与权限分离开来，由角色来充当中间桥梁，在访问过程中先将权限与角色关联，再将角色与用户关联，角色与用户是多对多的关系，即一个角色可以被多个用户使用，一个用户也可拥有多个角色。同样，权限与角色之间也是多对多的关系，在RBAC诸多模型中最具代表性的是1996年乔治麻省大学SandHu教授提出来的RBAC96模型。它是一个四层模型，包含了RBAC0，RBAC1，RBAC2，RBAC3，其中RBAC3模型又是前面三个模型的综合。RBAC模型包括用户、角色、权限、会话、用户指派（UA）、权限指派（PA）、角色层次（RH）等基本元素，它的基本结构图如图1所示。

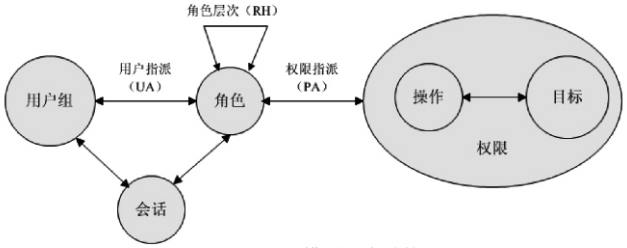

图1 RBAC模型基本结构

1.2 新改进的RBAC模型

1.2.1 在用户与角色之间加入了用户组概念

用户组是由若干个具有某一类相同特性的用户组成，在web信息系统中，大多数用户都会具有许多相同的权限，如果一个一个对用户进行授权的话，无疑工作量会变得巨大，而且容易出错。引入用户组后，可以将权限或者角色相同的一类用户放在一起，统一授予权限。用户组之间可以存在继承与包含的特性，不同用户组权限不一样，这样管理员可以实现对用户权限的动态管理。

1.2.2 引入角色优先级概念

另外在实际模型中可能存在这样一种情况，如，角色A拥有某项权限P，角色B没有这项权限，A与B具有互斥关系[8]，在传统的RBAC模型中用户是不能同时被赋予A和B这两个角色的。为解决这个问题，本文在定义角色的同时给出角色的优先级，若出现上述权限的互斥现象，则根据角色的优先级来决定用户是否具有当前权限，对于没有互斥的情况，则按照角色权限的并集来处理。

1.2.3 基于细粒度的权限管理

在传统的RBAC模型中，只将权限划分到菜单项一级，这样管理者只能控制用户的访问权限而不能控制用户的操作权限，而本文降权限细分到每个访问菜单下的功能按钮，实现了对用户操作权限的控制，能更好的应付各种情况的授权。改进后的RBAC模型结构图如图2所示。

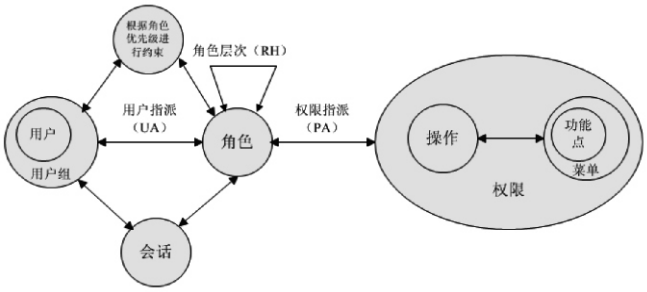

图2 改进后的RBAC模型

2 改进RBAC模型的权限管理系统的设计

2.1 权限管理系统总体框架设计

本权限管理子系统采用基于ASP.NET(C#)的B/S结构设计，整个系统使用现今比较流行的三层架构，由上至下分别为表现层（UI），业务逻辑层（BLL），数据访问层（DAL），在整个系统框架的开发过程中，采用了Spring.Net和NHibernate的技术。Spring.Net实现了框架的控制反转[7]和依赖注入，可以提供一种容器来进行有效的管理数据操作及相关的服务，当需要某个对象或者方法时，只需要通过容器将对象注入到程序中，这样对象之间的耦合度低，复用度高。在数据访问时采用了NHibernate技术，NHibernate是常用的开源ORM数据映射框架，NHibernate不仅管理类到数据库表的映射，还提供数据查询和获取数据的方法，可以大幅度减少开发时程序员使用SQL和ADO.NET处理数据的时间。

2.2 权限管理系统的数据库设计

数据库是一个信息系统的骨架，几乎用户所有的操作都和数据库有着紧密的联系，一个优秀的数据库设计会让整个系统的开发事倍功半，本系统的实现与数据库的设计有着紧密的联系。本权限管理系统在基于改进的

RBAC模型上结合业务需求共设计了七张数据表：用户表、角色表、用户角色关联表、菜单表、角色菜单关联表、功能表、角色功能关联表。用户与角色之间，角色与权限之间都是多对多的关系，用户表中定义用户组属性来确定相同权限的用户，角色表中定义角色优先级属性解决互斥角色授权问题，数据库设计E-R图为图3所示。

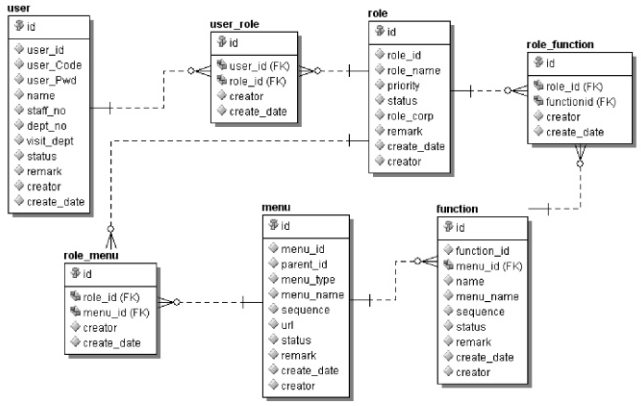

图3　权限系统数据库设计E-R图

3 基于改进RBAC模型的权限管理系统的实现

3.1 角色授权模块的实现

在基于RBAC模型的权限管理系统中，权限的授予和实现是两个核心的模块，为角色授权的具体过程是：管理员登陆系统进入角色管理模块，选中某个角色后，在系统资源树中勾选想要授予的权限，勾选完成后保存，数据库权限和菜单表将会自动插入选中的记录，该模块的关键点是怎么将权限保存和更新。本系统采用C#语言，核心代码如下：

```
// 更新角色关联的菜单和功能点到关联表
if (oldmenuIds.Contains(checkmenu[i]))
oldmenuIds.Remove(checkmenu[i]);//如果新勾选的权限原来就有，则保持原来的。
ServiceLocator.MenuService.SaveRoleMenu(rm);// 保存新勾选的权限。
ServiceLocator.MenuService.DeleteRoleMenu(rm);//删掉原来有，现在没有的权限。
```

上述代码实现了角色权限的保存和更新，实际应用中界面为图4所示。

图4　实际应用中角色授权界面

3.2 用户权限认证模块的实现

在基于细粒度访问的两级权限管理系统中，权限认证分为两部分，一部分为导航菜单的权限认证：用户在登陆系统后，系统根据用户所拥有的角色权限自动为用户生成首页导航菜单，另一部分是用户对资源进行操作的即时认证。生成导航菜单的实现过程用自然语言描述如下：

1）根据Session中登陆用户的id去user_role关联表找出当前用户关联的所有角色。

2）将用户关联的角色以优先级由高到低排列，放在临时定义的泛型数组中。

3）遍历所有角色，通过角色id找出该角色下所有菜单，其中，优先级高的角色权限覆盖优先级低的角色，将找出的菜单权限存入临时数组内。

4）根据临时数组内的菜单权限，生成导航菜单树，用户通过树节点链接来访问对应的功能模块。

权限认证的另一部分为对资源操作权限的认证，即用户点击进入某个菜单模块，验证用户是否具有当前菜单下的功能点权限，我们定义了ComfirmAuth()函数来验证用户功能点权限，函数的两个参数分别为当前菜单下所有的功能点和用户拥有的所用功能点，如果用户所拥有的功能点权限不包括当前菜单下的功能点权限，则隐藏该按钮来控制用户的功能使用，核心代码如下：

```
protected void ComfirmAuth(IList<string> func, IList<string> funcname)
 {
Control c = FindControl(funcname[i]);
c.Visible = false; //先将菜单下的功能点隐藏
//如果用户的功能点权限列表包括当前菜单下的功能点，则显示该功能点
if (func.Contains(funcname[i]))
    hasfunc.Add(funcname[i]);
Control c = FindControl(hasfunc[i]);
    c.Visible = true;
```

4 结语

本文通过对企业应用需求的分析并综合访问控制的知识，在传统RBAC模型的基础上提出了一种改进的RBAC模型，该模型能够对一组权限相同的用户统一授权，减少了管理员授权工作量，在角色授权时设置了角色优先级，解决了互斥角色授权的问题，能适应不同场合的授权需要，另外该模型将资源的控制细分到按钮级别，将用户的访问权限与操作权限分离出来。基于该模型本文设计并实现了一种通用的权限管理系统，该系统在实际项目中检验可行，能很好的完成各种情况下的权限分配。

参考文献：

[1]D.Ferraiolo,R.Kuhn.Role-based access controls. In:Proceedings of the 15th NIST-NSA National Computer Security Conference.1992: 554-563.

[2]SANDHU R,COYNE E,FEINSTEIN H,YOUMANC.Role-Based access control models[J].IEEE Computer,1996,29(2):38-47.

[3]乔佩利、陈欣，一个基于改进RBAC模型的自适应权限构件[J].自动化技术与应用，2011，30（3）：30-34.

[4]江南、王士同、贺杨成，关于改进的RBAC模型研究及应用实现[J].微计算机信息，2011，3：169-171.

[5]张盈谦、孙斌、刘佳、董宗然，基于AOP的细粒度RBAC模型研究[J].电子设计工程，2011，19（18）：161-163.

[6]汪原祥、李卉，基于角色的访问控制研究[J].计算机应用研究，2005，7（4）：125-127.

[7]Martin Fowler. Inversion of Control Containers and the Dependency Injection[EB/OL].http://martinfowler.com/articles/injection.html.2004-1-23.

[8]胡金柱、陈娟娟，RBAC模型中角色的继承与互斥问题的研究[J].计算机科学，2003，30（11）：160-163.

作者简介：

文骁一（1986-），男，汉族，湖南省长沙人，厦门大学信息科学与技术学院09级系统工程专业硕士，研究方向：信息系统，企业综合自动化平台。