文章编号:1001 - 9081(2008)09 - 2449 - 03

基于 RBAC 权限管理模型的改进与应用

刘建圻, 曾 碧,郑秀璋
(广东工业大学 计算机学院,广州 510090)
(Liujian1818@163.com)

摘 要:在分析现有权限管理系统权限的基础上,给出了新的授权模式,A 用户在其自己的可授权的权限集合内可以直接对 B 用户授权,并且可设定所授权限的时效,B 用户可以根据 A 用户规定时效内拥有 A 用户的权限。该方案采用静态权限分配与动态权限授权相结合,很好的解决了传统权限管理的僵化。改进后的方案在实际的项目中得到了应用,并取的很好的效果。

关键词:权限管理;权限模型;动态授权
中图分类号: TP311.52 **文献标志码**:A

Improvement and application of access control mode based on RBAC

LIU Jian-qi, ZENG Bi, ZHENG Xiu-zhang
(*Faculty of Computer, Guangdong University of Technology, Guangzhou Guangdong* 510090, *China*)

Abstract: A new access authorization mode was established based on the analysis of access of existing access authorization control system, in which user A can authorize directly user B within his available authorization set, as well as set a time of validity, during which user B own user A' s access authorization. This scheme combined static authority allotment with dynamic access authorization, solving the restriction of traditional authority management system. The improved scheme had been well applied in actual projects, getting good effect.

Key words: authorization management; authorization model; dynamic authorization

0 引言

随着应用系统的普及,权限管理的模型和对象也越来越复杂。目前常用的权限管理系统主要采用基于 RABC 的权限管理模型,它的基本思想是根据企业中的职能岗位划分出不同的角色,将系统资源的访问权限附加到角色上,并为用户分配相应的角色,通过控制角色权限来间接地控制拥护对系统资源的访问。现有的应用系统通常采用基于角色访问权限[1](Role-Based Access Control, RBAC)对用户进行权限管理。如图 1 所示。

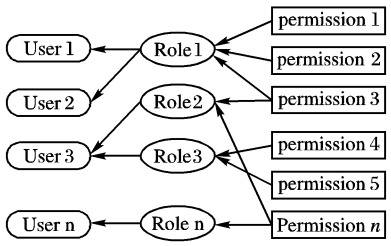

图 1 RBAC 授权模型

RBAC 模型通过用户和角色之间的联系来降低了具有大量用户和权限分配的系统的管理的复杂性[2]。虽然这样的系统设置相对简单,容易管理,但是 RBAC 模型的授权方式缺乏灵活性。例如,公司总经理因公出差半个月,要授权给甲、乙、丙三位副总代为管理公司事务,三位副总要分别获得公司管理系统的部分管理权限。因此,要把 Perm A、Perm B、Perm C 权限分别授权给甲、乙、丙,且时效是半个月。现有的授权模式处理方式如下:系统管理员先在系统中增加新的角色(假定为 Role 甲、Role 乙、Role 丙),然后把 PermA 赋给角色 Role 甲、PermB 赋给角色 Role 乙、PermC 赋给角色 Role 丙,最后把甲、乙、丙三位副总分别设定角色,整个处理过程非常繁杂,而且需要系统管理员的参与。如果在半个月期间,公司的副总甲又要出差,情况会变的更为复杂。半个月后;总经理取消授权,又必须重新完成一遍相反的程序。这样的授权模式缺乏灵活性、不适应现代企业变化的特点,不利于现代公司的管理[3]。

针对这样的情况,需要重新考虑系统的权限授权的问题,根据需求来调整权限管理模型。提高权限授权的灵活性,动态性。因此,本文提出了一种新的动态授权模式。

1 基于角色的权限管理的改进模型

1.1 基本概念的描述

在各类管理系统中,我们把一个用户的所拥有的所有权限称为用户权限集合 P,把权限集合 P 分成三部分:角色权限 Pr(Permission of Role)、个人权限 Pp(Permission of Person)、授权权限 Pa(Permission of Authorization)。

定义 1 用户权限集合 $P = \{Pr \cup Pp \cup Pa\}$

其中角色权限是用户根据授予角色而获得的权限,这类权限角色相同权限也是一样。个人权限是个人通过系统直接授予而获得的具体功能权限[4]。由于现代公司的工作的不断的细化,即使角色相同的情况下,每个人工作也会有所不同,分配的权限也会有所不同。授权权限是其他用户授权而

收稿日期:2008 - 03 - 14;**修回日期**:2008 - 05 - 29。

作者简介:刘建圻(1982 -),男,江西兴国人,硕士研究生,主要研究方向:嵌入式系统; 曾碧(1963 -),女,广东梅州人,教授,主要研究方向:嵌入式系统、智能机器人; 郑秀璋(1983 -),男,广东阳江人,硕士研究生,主要研究方向:智能机器人。

得到的权限。

定义2 用户可授权的权限集合 PA = {Pr∪Pp∪Par}

授权权限又可以分为可重复授权权限 Par(Reapetable Permission of Authorization)和不可重复授权权限 Paur(UnReapetable Permission of Authorization)。可重复授权权限是某用户授权时允许被授权用户再次授权给其他用户的权限。公司总经理授权给副总甲如果允许副总甲再次授权给职能经理,总经理授权给副总甲的那部分权限就是可重复授权权限。不可重复授权权限是在某用户授权时不允许被授权用户再次授权给其他用户的权限。

定义3 授权时效。用户赋予被授权用户使用权限的时长,即 dateline(权限过期时间)减去用户授权时的系统时间。在这段时间内被授权用户可以根据权限的许可,处理有关事宜。总经理授权副总行使总经理权限的时间为半个月,超出半个月,权限自动失效。半个月及为授权时效。

1.2 改进的授权模式的描述

以前的授权模式呆板僵化,而且每次权限的改变都必须有系统管理员的参与,现在已经无法应对现在企业管理的变化[5]。下文根据现在企业管理的特点,设计出一种新的方案。该方案同时提出了三种授权方式:角色授权、个人授权、用户授权,如图2所示

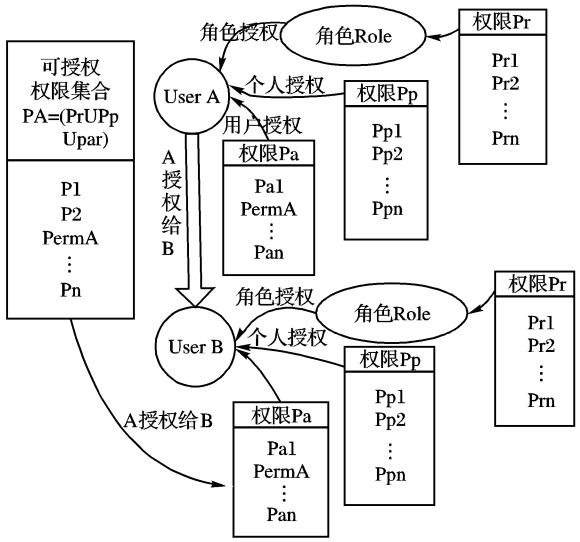

图2 授权模型

1)角色授权:对用户的授权通过授予特定的角色来实现的。角色拥有完成某一特定职能的权限,用户经角色授权后享有被角色所拥有的权限。通过角色授权而获得的权限具有永久时效。

2)个人授权:对用户的授权通过直接授予具体的功能权限。通过个人授权获得的权限具有永久时效。

3)用户授权:对用户的授权通过其他用户授予的权限来实现的。通过其他用户授权而获得的权限的时效由授权用户设定。

在系统的权限的实现中,系统管理员通过把三种授权方式相结合。对不同职责的用户进行分类,创建不同的角色,对不同的角色授予可以完成特定职能的角色。在给不同的用户授予不同的角色。当需要区分细化具有相同角色的用户权限时,这是在通过个人授权来修改用户的具体权限。通过以上步骤可以初步的完成系统的静态权限的分配。实现权限的动态授予必须使用用户授权来实现。

1.3 改进的用户授权与取消用户授权

用户授权:用户在自己的可授权的权限空间中选择需要授权的权限,权限是通过其他用户授权而获得的并且该权限的 dateline 是大于当前系统时间,可把权限授予给该用户,并设定权限的时效 dateline(必须小于用户本身的权限的 dateline)和是否可重复授权,该用户通过授予在设定的时效拥有了该权限,如果是可重复授权的权限,该用户还可以把该权限授予给他人。如果权限是通过角色和个人授权而获得的,设定权限时效和是否可重复授权后,就可以直接授权给其他用户。

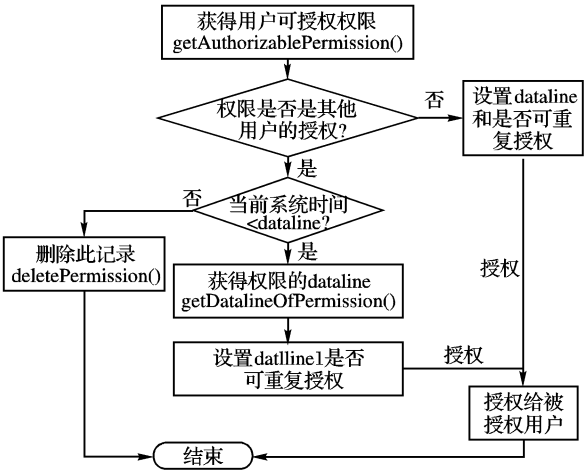

图3 用户授权流程

取消用户授权:取消授权的方式有两种:1)被授权用户主动删除该权限,被授权用户在登录时,遍历用户权限表。当时间超过设定的 dataline 时,被授权用户在读取权限时用当前时间与比较 dateline 大时,调用删除权限函数 deletePermission(),删除该权限记录。2)授权用户主动取消授权,用户在 dateline 未到时,主动调用取消权限函数,在数据库表搜索该权限并且删除。多个用户对同一权限重复授权后,将会形成一个树状权限链,可用树的搜索算法逐个删除用户权限。

2 角色权限管理在实际项目中的应用

在东莞移动巡检系统中,在权限管理模块我们采用新的授权模型,这种新的权限授权模式在实际的项目中得到很好的应用和验证。系统开发环境如下:C#(.net)开发语言,采用 Microsoft SQL Server2000 数据库。以下是实际的权限管理系统的主要开发过程:

2.1 数据库设计

系统的关键表如图4。用户表存储用户基本信息,角色表存储角色信息,用户和角色通过用户角色表实现多对多的关系。功能表存储各具体的功能点,pID 采用编码规则来生成具体的系统菜单,pOperation 表示系统菜单的具体操作权限。角色与功能通过角色功能表实现多对多的关系。

用户权限表是实现用户授权的关键表,具体如表1,dateline 字段定义 pID 功能点权限的到期时间,如果系统当前

的时间大于 dataline,系统将删除此条记录。isRepeat 字段设定此权限是否可以重复授权,其值为 True 时,用户可以将此功能权限再次授予给其他用户,但是在授予时,设定时效时(dateline)必须比自己的 dateline 小。当其值为 False 时是,用户不得再次授权。authorizorID 字段定义了授权用户 ID,此字段是用户授权过程中系统自动添加的,用于用户取消权限。

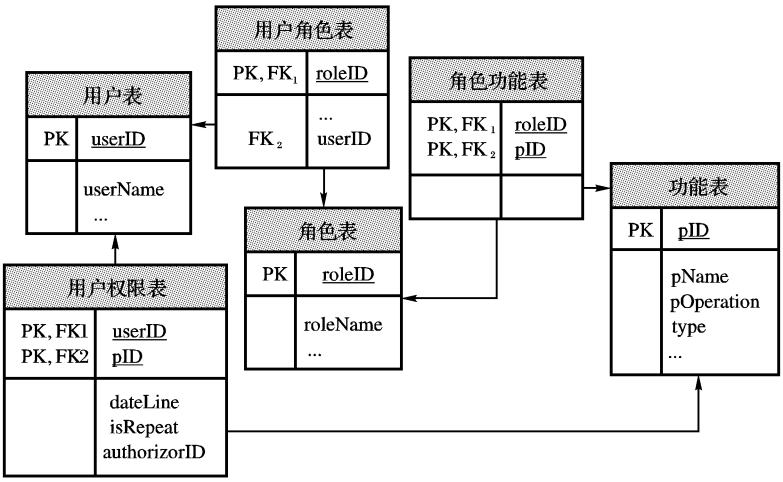

图 4　数据库设计模型

表 1　用户权限表

字段名称	字段类型	字段描述
userID	int(8)	用户 ID
pID	varchar(32)	功能 ID
type	int(8)	功能类型
dateline	date	权限截止时间
isRepeat	bool	是否重复授权
authorizorID	int(8)	授权用户 ID

2.2　实体类设计

本系统使用 AuthorizePermission 类来实现权限的静态分配和动态授权。

```
Class AuthorizePermission {
public:
  //构造函数
  AuthorizePermission ();
  //获得用户的权限空间
  struct getPermission (int userID);
  //授权方法, authorizedID 被授权用户        //ID, pID 授权权限
  bool authorize( int authorizedID, varchar pID, date dateline);
  //用户取消授权, 实现算法参考树的搜索算法删除记录.
  bool unauthorized( int authorizedID, int authorizorID);
private:
  //被授权用户时效过期删除权限
  bool deletePermission ();
  //获得用户可授权权限
  struct getAuthorizablePermission ();
  date getDatelineOfPermission( int userID, varchar pID)
}
```

以上是简单的类的接口,实现代码很长,不一一详述。

2.3　改进的授权模型应用效果与用户反馈

应用改进的授权模型后,大大简化了授权手续,减轻了系统管理人员的工作负担。根据用户使用反馈信息和比较原有的授权模型,本授权模型具体从以下几个方面得到了改进。

1)增加了系统的灵活性。管理人员可以根据工作的实际情况自己授权给不同的人员,并且可以随时取消所授的权限。还可以重复授权,解决了公司由于领导先后出差而引起的授权难题。

2)减轻了系统管理人员的工作负担。因授权问题大大增加了系统管理人员的工作量,使用本授权系统之后,授权和取消授权不需要系统管理员的参与,系统管理员可以专心于系统维护工作,在原有系统的基础上可以适当的裁减工作人员,减轻企业的负担。

3)增加了管理的安全性。原有的系统是通过管理员来进行授权和取消授权,由于各种原因,已经授出的权限没有取消,而引起被授权者一直拥有该权限,增加了管理的不安全性。本模型可以根据授权者的意愿设定时效,系统可以自动取消所授的权限。从而避免了此类事件的发生,增加了管理的安全性。

3　结语

目前,该改进的方案在实际的项目中得到了应用,并且获得了较好的效果。它不但继承了角色授权和个人授权的传统静态权限管理的优点,同时引入了用户授权,用户可以动态的根据自己需要授权给其他用户并且可以设置时效,同时也可以随时收回自己的授权。这个方案更能够接近实际应用系统的需要,更能在实际应用中解决客户的需求。但是在被授权用户在时效过期时要遍历整个用户权限表,对系统的运行速度有一的影响。如何更好的提高执行速度是我们下一步着重考虑的问题。

参考文献:

[1] OSBORN S, SANDHU R. Configuring role-based access control to enforce mandatory and discretionary access control policies[J]. ACM Transactions on Information and System Security, 2000, 3(2): 123 - 132.

[2] LIN A, BROWN R. The application of Security policy to rose-based access control and the common data security architecture[J]. Computer Communication, 2000, 23(17): 1584 - 1593.

[3] 王培康, 胡访宇, 袁平波. 一种信息系统授权实现方法[J]. 计算机工程, 2001, 27(1): 135 - 136.

[4] 杨强, 王忠民. ERP 系统用户权限分配问题的实现[J]. 微机发展, 2004, 14(7): 16 - 17.

[5] 朱磊, 周明辉, 刘天成, 等. 一种面向服务的权限管理模型[J]. 计算机学报, 2005, 28(4): 677 - 685.