基于角色的访问控制在教务管理系统中的应用

Application of role – based access control in educational administration system

王 成　王世波　王 铁　(齐齐哈尔大学经济管理学院 161000)

摘要:基于角色的访问控制(RBAC)模型是目前主流的访问控制模型。本文主要论述了基于角色的访问控制模型基本理论,并在此基础上系统论述了该模型在教务管理系统中的应用,同时根据教务管理系统权限管理的特点,在 RBAC 模型的基础上增加了部门权限控制,弥补了 RBAC 只能实现功能级别访问控制的不足。

关键词:角色 访问控制 基于角色的访问控制模型 教务管理

1 访问控制方法概述

随着计算机技术、通信技术和互联网的飞速发展,计算机安全性已经越来越引起人们的重视。访问控制作为一种重要的安全技术,已经渗透到操作系统、数据库、网络的各个方面。访问控制是众多计算机安全解决方案中的一种,是最直观最自然的一种方案。信息安全的风险(Information security risks)可以被宽泛的归结为 CIA:信息机密性(Confidentiality)、信息完整性(Integrity)和信息可用性(Availability)。访问控制主要为信息机密性和信息完整性提供保障。访问控制方法一般有三种:

(1) 自主访问控制模型(DAC Model,Discretionary Access Control Model),是根据自主访问控制策略建立的一种模型,允许合法用户以用户或用户组的身份访问策略规定的客体,同时阻止非授权用户访问客体,某些用户还可以自主地把自己所拥有的客体的访问权限授予其它用户。

(2) 强制访问控制模型(MAC Model:Mandatory Access Control Model),在此模型中,每一个数据对象被标以一定的密级,每一个用户也被授予某一个级别的许可证。对于任意一个对象,只有具有合法许可证的用户才可以存取。强制访问控制相对比较严格,这种方式常用于多层次安全级别的军事应用。

(3) 基于角色的访问控制模型(RBAC Model,Role – based Access Control Model),是目前公认的解决大型企业的统一资源访问控制的有效方法。其显著的两大特征是: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。

2 基于角色的访问控制模型

基于角色的访问控制是近年来发展起来的一种访问控制模型,它由美国 George Mason 大学的 Ravi Sandhu 等人提出。RBAC 模型在用户和访问权限之间引入了角色的概念,它的基本特征就是根据安全策略划分角色,对每个角色分配操作许可,为用户指派角色,用户通过角色间接地对信息资源进行访问。根据对不同复杂度权限的需求,RBAC 参考模型定义了三部分组件:

2.1 核心 RBAC

核心 RBAC 定义了 RBAC 模型的最基本的五个元素:用户、角色、对象(资源)、操作、权限(许可),各元素基本含义如下:

用户(User):是一个访问计算机系统中的数据或者用数据表示的其它资源的主体。我们用 U 表示全体用户的集合。用户一般情况下指人,也可为 Agent 等智能程序。

角色(Role):是指一个组织或任务中的工作或位置,代表了一种资格、权利和责任。我们用 R 表示全体角色的集合。角色是一种语义综合体,可以是一种抽象概念,也可以对应于具体应用领域内的职位和权利。管理员角色(Administrator Role)是一种特定的角色,

用来对角色的访问权限进行设置和管理。在集中式管理控制模型中，管理员角色由一个系统安全管理员来完成；而在分布式管理控制模型中，可以采用制定区域管理员来对系统进行分布式管理，每个管理员可以管理该区域内的角色权限的配置情况。当然区域管理员的创建和权限授予则统一由顶级的系统安全管理员完成。

对象(Object)：任何访问控制机制都是为了保护系统的资源，在不同应用背景下对象可以有相当广泛的定义，比如在操作系统中可以是一段内存空间，在数据库里可以是一个表中的记录，在 Web 上可以是一个页面。

操作(Management)：是程序的可执行的反映，被用户调用和执行。操作的类型取决于实现系统的类型。例如文件系统可能的操作是读写、执行等，数据库系统则是创建、修改、删除等。

权限(Permission)：是对计算机系统中的数据或者用数据表示的其它资源进行访问的许可。我们用 P 表示全体权限的集合。权限一般是一种抽象概念，表示对于某种客体资源的某种操作许可。因此有的模型中将权限细化为二元组(操作，对象)，其中对象是访问控制系统中的真正客体，操作是作用在该对象上的一种访问方式。由于该二元组中操作一般是与具体的对象相关的，我们在下面的论述中认为权限是一个语义统一体。

RBAC 的关注点是角色和用户、权限的关系，称为用户分配(User Assignment)和权限分配(Permission Assignment)。用户分配是用户集 U 到角色集 R 的一种多对多的关系，即有 $UA \subseteq U \times R$，也称为角色授权(Role Authorization)。$(u,r) \in UA$ 表示用户 u 拥有角色 r，从语义上来说就表示 u 拥有 r 所具有的权限。权限分配是权限集 P 到角色集 R 的一种多对多的关系，即有 $PA \subseteq P \times R$。$(p,r) \in PA$ 表示权限 p 被赋予角色 r，从语义上来说就表示拥有角色 r 的用户拥有权限 p。

2.2 继承 RBAC

继承 RBAC 组件引入了角色继承的概念。这里首先介绍两个与角色继承相关的基本概念：

2.2.1 角色继承关系(Role Inheritance)：是角色集 R 中元素间的一种偏序关系 $\geq$，满足：

(1) 自反性：$\forall\ r \in R, r \geq r$；

(2) 反对称性：$\forall\ r_1, r_2 \in R, r_1 \geq r_2 \cap r_2 \geq r_1 \Rightarrow r_1 = r_2$；

(3) 传递性：$\forall\ r_1, r_2, r_3 \in R, r_1 \geq r_2 \cap r_2 \geq r_3 \Rightarrow r_1 \geq r_3$。

从语义上来说，两个角色 r1≥r2 是指前者比后者级别更高，具有更大的权利。形式化的说，$r_1 \geq r_2$ 蕴涵 r_2 对应的权限指派 r_1 也拥有，同时 r_1 对应的用户指派 r_2 也拥有，即有

$r_1 \geq r_2 \Rightarrow Permission(r_2) \subseteq Permission(r_1) \cap User(r_1) \subseteq User(r_2)$

其中 Permission(r)表示 r 对应的权限集，User(r)表示 r 对应的用户集。角色继承关系允许存在各种形式，包括多重继承。在这个偏序的意义下，角色集中并不一定存在最大角色和最小角色。

2.2.2 角色层次图(Role Hierarchies)：是给定了角色继承关系之后整个角色集形成的一个层次图。如果 $r_1 \geq r_2$，那么在图上就存在 r_1 到 r_2 的一条有向边。根据不同的角色偏序定义，角色层次图可以是树、倒装树、格，甚至极为复杂的图。

继承 RBAC 包括限制继承和一般继承两种，其区别是：前者可以有多个父节点，但是只能有一个子节点，是一个反向树结构，而后者是图。角色继承主要用于解决复杂组织机构之间的权限关系。它是一种很自然的对组织机构层次中权限和责任的映射，例如总经理→部门经理→员工。角色继承方向和用户的关系方向是相反的，即权限最小的在顶端：员工→部门经理→总经理。

2.3 限制 RBAC

限制 RBAC 组件引入了限制(Constraints)的概念，它是指在整个模型上的一系列约束条件，用来控制指派操作，指定职责分离(SD，Separation of Duty)以及避免冲突等。这是一个非常抽象的概念，RBAC 模型中并没有给出限制的类型和表述方式。任何独立于前面诸多术语的约束条件都是限制的一种形式。典型的限制包括指定角色互斥关系、角色基数限制。

2.3.1 角色互斥关系(Mutually Exclusive Roles)

用于指定两个角色具有不同的职责，不能让一个用户同时拥有。银行的出纳和会计便是角色互斥的简单例子。角色互斥关系的目的是为了在 RBAC 模型中引入业务逻辑的规则，避免冲突的发生。

2.3.2 角色基数限制(Role Cardinality Constraints)

用于指定一个角色可被同时授权或激活的数目。比如总经理只能由一个用户担任,那么总经理角色的角色基数就为1。根据下面定义的静态和动态的职责分离,角色基数限制有静态角色基数限制和动态角色基数限制两种。

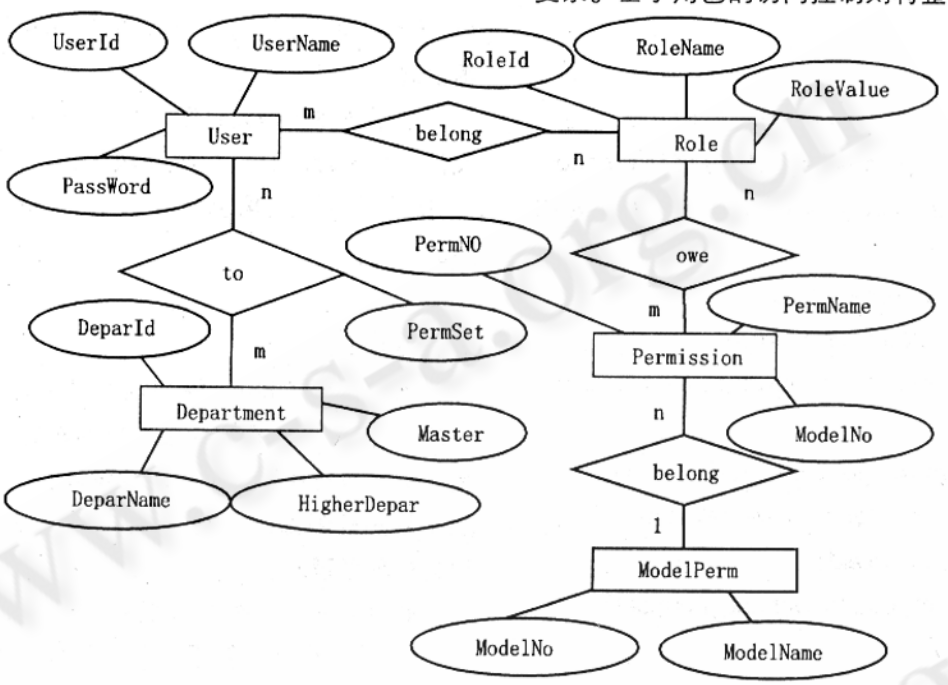

图 1　RBAC 系统各实体的 E-R 图

根据职责分离的不同阶段,限制一般可分为静态职责分离和动态职责分离。

(1) 静态职责分离(Static SD)。是指限制定义在用户指派(角色授权)阶段,与角色激活无关。以角色互斥为例,如果定义两个角色为静态的角色互斥,那么任何一个用户都不能同时被指派到这两个角色。静态职责分离实现简单,语义清晰,便于管理,但是不够灵活,有些实际情况无法处理。

(2) 动态职责分离(Dynamic SD)。是指限制定义在角色激活阶段。仍以角色互斥为例,如果定义两个角色为动态的角色互斥,那么一个用户可以同时被指派这两个角色,但是在任何一个会话中都不能同时激活它们。由此可见动态职责分离更灵活,基本上能处理各种实际情况,但实现略复杂。

3 基于角色的访问控制模型在教务管理系统中的应用

3.1 教务管理系统权限控制需求分析

高校的教务管理系统是一个用户数量多,用户权限分配复杂的系统。利用像自主型、强制型等将主体和客体直接绑定的访问控制方法会导致授权工作极其复杂。基于角色的访问控制则将整个访问过程分成两步:访问权限与角色相关联,角色再与用户相关联,从而实现用户和访问权限的逻辑分离。当用户的部门或权限发生变动时(譬如:某几个专业合并为系,原来的教研室主任现随之上升为系主任,权限发生了变动),系统可以很灵活地将该用户从一个角色转移到另一个角色来实现权限的协调转换。另外,在学校机构发生职能性改变时,应用系统只需要对角色进行重新授权或取消某些权限,就可以使系统重新适应需求。再者,角色之间还可以有继承、限制等逻辑关系,通过这些关系可以更好地完善用户和权限的实际对应。综上看来,对于权限分配复杂的教务管理系统来说,权限管理应用基于角色的访问控制模型是最佳的选择。

3.2 概念结构设计

通过对该系统权限管理的具体分析,我们设计了用户(user)、角色(role)、部门(department)、权限(permission)、权限模块(model_permission)等五个实

体,它们之间的关系及各实体的属性如图1所示。

注:User 与 Role 之间的 belong 关系是指用户属于哪些角色;User 与 Department 之间的 to 关系是指所有用户分别与各部门相对应;Role 与 Permission 之间的 owe 关系是指某一角色所拥有的权限;Permission 与 ModelPerm 之间的 belong 关系是指权限所属的权限模块。

3.3 逻辑结构设计

将概念结构设计中用户(user)、角色(role)、部门(department)、权限(permission)、权限模块(model_permission)各实体的实体关系图转换为关系模式如下:

User(UserId,UserName,Password)

Role(RoleId,RoleName,RoleValue)

Department(DeparId,DeparName,Master,HigherDepar)

Permission(PermNo,PermName,ModelNo)

Model_Permission(ModelNo,ModelName)

User_Role(UserId,RoleId)

User_Department(UserId,DeparId,PermSet)

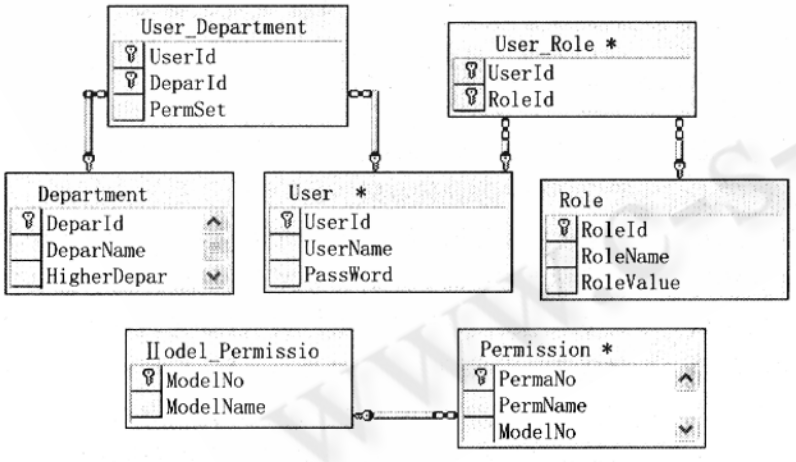

图 2　RBAC 访问控制数据库关系图

说明:有下划线的属性为主键;在转换为关系模式时,我们并没有将 Role 与 Permission 单独转换为一个表,而是把角色表的 RoleValue 字段类型定义为类似二进制数的0、1代码,并将权限表的 PermNo 的值与角色表的 RoleValue 字段值的位置相对应,如果 RoleValue 字段某位置的值为0,则表示该角色不具有该位置所对应的权限表里的权限,如果值为1则相反。这种方法简化了数据表的设计,方便了系统的实现。

3.4 数据库设计

数据库设计是整个访问控制设计的核心。在投放应用前,它将存储所有部门及功能模块的信息;在投放应用以后,它将存储所有系统用户定义的新用户、角色、用户所属部门、用户所属角色、角色所具有的权限等信息。

根据逻辑结构设计阶段的设计,我们在数据库中设计用户表、部门表、权限模块表、权限表、角色表、用户_部门表、用户_角色表等七个表,关系图如图2所示。

4 结束语

本文介绍了 RBAC 模型在教务管理系统权限管理中的应用,重点分析和介绍了数据库的设计过程,系统地分析了数据库表和关系图的设计。本文还对角色增加了用户权限部门限制,弥补了单纯基于角色无法实现的权限管理。

通过对角色在整个权限管理系统中应用的分析,不难看出基于角色的访问控制模型比传统的自主访问控制和强制访问控制更优越,同时也提供了更高的灵活性和扩展性。

参考文献

1 桂艳峰、林作铨,一个基于角色的 WEB 安全访问控制系统,计算机研究与发展,2002,24-45.

2 张正球、张志明、余敏,基于 Web 的信息系统中 RBAC 的实现,计算机与现代化,2005(1),70-73.

3 郑涛,一种应用于管理信息系统的访问控制策略,海空航空工程学院学报,2005(1),167-169.

4 郭力兵、徐东平,基于角色的数据库安全访问控制方法的设计与实现,交通与计算机,2005(1),120-122.