RBAC在MIS中的应用

Application of Role-based Access Control in MIS

摘要：基于角色访问控制具有减少授权管理复杂性，降低管理开销，增强系统的安全性管理功能，本文以某经营公司管理系统的开发为例，提出了RBAC在管理信息系统中的实现，不同的用户拥有不同的操作权限，使得整个系统的管理得到了优化。

关键词：基于角色的访问控制(RBAC)　管理信息系统(MIS)　权限管理

陈芳　陈朝

(武汉中国地质大学　430074)

1 引言

对数据库的访问控制实质上是对资源使用的限制，决定主体是否被授权而对客体执行某种操作。目前对于数据主要有三种不同的访问控制策略：

(1) 强制访问控制（Mandatory Access Control, MAC）

MAC是指系统强制主体服从事先制订的访问控制政策。它主要用于多层次安全级别的军事应用中，预先定义用户的可信任级别及信息的敏感程度（安全级别），当用户提出访问请求时，系统对两者进行比较以确定访问是否合法。其缺点在于主体访问级别和客体访问级别的划分与现实要求无法一致，在同级别间缺乏控制机制，灵活性较差。

(2) 自主访问控制（Discretionary Access Control, DAC）

DAC是在确认主体身份及所属的组的基础上，对访问进行限定的一种控制策略，访问控制策略保存在一个矩阵中，行为主体，列为客体。为了提高效率，系统不保存整个矩阵，在具体实现时是基于矩阵的行或列来实现访问控制策略。目前以基于列（客体）的访问控制表ACL采用得最多，其缺点在于ACL中含有大量的表单，当组织内的人员发生变化（升迁、换岗、招聘、离职）、工作职能发生变化（新增业务）时，ACL的修改异常困难。因此，DAC十分灵活，但安全性不强，授权管理复杂（用户，权限，存取对象）。

(3) 基于角色的访问控制（Role-based Access Control, RBAC）

在RBAC中，由系统管理员将一组权限付给角色，角色分配给用户。一个用户可拥有多个角色，一个角色可授权给多个用户；一个角色可包含多个权限，一个权限可被多个角色包含。用户通过角色享有权限，它不直接与权限相关联；权限对存取对象的操作是通过活跃角色实现的。用户与角色，角色与权限，角色与存取对象之间的关系均为多对多关系。

近年来，基于角色的访问控制得到广泛的应用，该技术主要研究将用户划分成与其在组织结构体系相一致的角色，以减少授权管理的复杂性，降低管理开销和为管理员提供一个比较好的实现复杂安全政策的环境而著称。目前已提出RBAC模型的有美国国家标准与技术局的NIST RBAC模型和George Mason大学的RBAC96模型两种。本文提出了基于角色访问控制的新型安全管理机制在MIS中的应用，有效的实现了前后台安全机制的统一，增强了系统的安全管理性能。

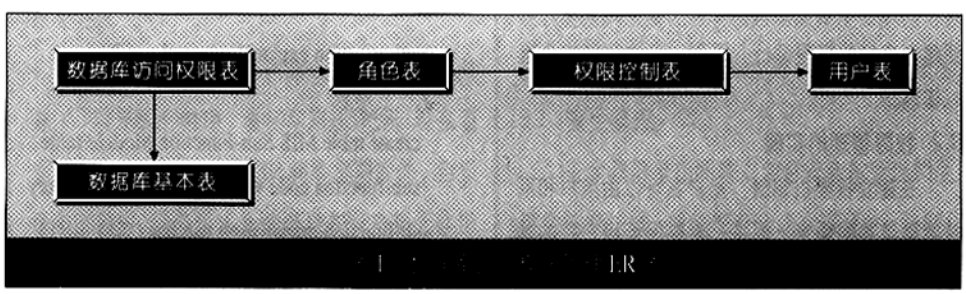
2 MIS中基于角色权限管理的模型设计

RBAC的基本思想是：授权给用户的访问权限通常由用户在一个组织中担当的角色来确定。ACL直接将主体和受控客体相联系，而RBAC在中间加入了角色，通过角色沟通主体和客体，分层的优点是当主体发生变化时，只需修改主体和角色之间的关联而不必修改角色于客体的关系。

MIS的权限管理从功能模型和信息模型的角度可分为两个层次，即功能层的访问权限和数据库访问层的访问权限。对MIS的功能层和数据库访问层进行权限管理的信息模型包含以下实体：用户，角色，数据库访问权限表，数据库基本表，MIS子功能模块表，功能模块访问权限，模块菜单。对于MIS功能层管理到"窗体"的菜单层，对于数据库访问层管理到基本表的操作权限。

数据库访问权限的管理包含用户表，角色表，权限控制表，数据库访问权限表，数据库基本表。用户表用于存储所有用户，是系统的个体用户集，随用户的添加与删除动态的变化。角色表用于存储所有的角色，是系统的角色集，由系统员进行定义。数据库基本表用于存储MIS中所有数据库的基本表，角色和数据库基本表之间是多对多的关系，为了消除这种多对多的非确定的关系，引入"数据库访问权限表"这一中间实体，给某角色分配了权限后，就将这个角色分配给多个用户，角色表和用户表之间也是"多对多"的关系，又引入权限控制表一实体来进行限制。实现的ER图如图1所示：

功能层的访问权限管理包含用户表，角色表，权限控制表，MIS子功能模块表，模块菜单表，功能模块访问权限表。其中用户表和角色表分别存储系统的用户信息和角色设定信息；MIS子功能模块表存储应用系统的所有子模块的信息，模块菜单表存储相应模块的菜单信息，通常包括对该模块的"增加、删除、修改、查询"几个基本操作。同样引入功能模块访问权限表来限制角色和MIS子系



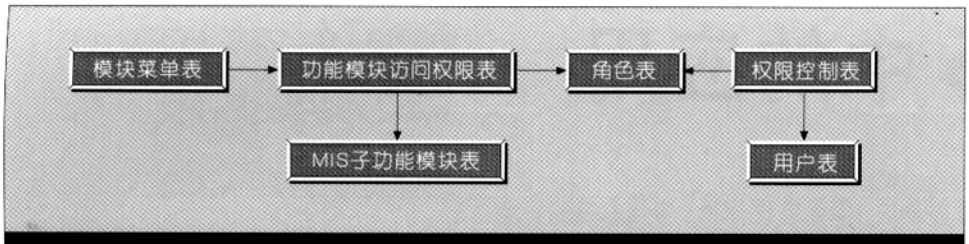


图2 功能层访问权限管理ER图

统间的多对多的关系，并将模块菜单名称、角色名、子功能模块的代码合起来做主键，以满足“三范式”。实现的ER图如图2所示：

3 MIS中权限管理的实现

3.1 权限设计的安全问题

在实际中，该经营公司的管理信息系统是一个多用户的网络信息系统，主要是完成一系列的公文创建及其流转问题，其程序和模块通常是以诸多功能项的形式（菜单、窗口、按钮、编辑区等）向用户提供各种操作，但并不是每个用户都有权使用系统的每项功能，需要根据用户的不同职责对其操作权限加以限定。它具有以下特性：

(1) 公文流转是一种协同工作，它由一系列的人员分阶段共同完成，并且每一个人只能在指定的阶段进行处理。

(2) 每一个参与者的角色（岗位）不同，对于公文的处理方式也不同。

(3) 不同角色拥有不同的权限，只能在指定的数据区域进行察看和编辑。

考虑到网络的安全性，公文在内部网上进行流转时，应对公文的不同执行者进行权限控制，以防出现下列问题：

① 非授权人员篡改基本数据区的数据。

② 非审批人员在评阅区填写意见。

③ 非承办人员在文档已被评阅批示后，再修改基本数据区，使承办人员或批示人员承担责任。

④ 承办人员在评阅完成后，再修改批示意见，让领导承担责任。

因此为了有效防止上述可能事件的发生，必须建立良好的权限设计，否则，公文流转将会难以实现。

3.2 权限管理的实现

在建立权限机制时，用户名可以由用户和系统管理员共同确定，但是一定要保证用户名的唯一，且用户拥有自己的密码。角色名要按照实际情况决定，这里的标准是：尽量使工作细化到一个人完成的粒度。

经过以上的说明建立以下几个基本表：用户表（Users）、角色表(Roles)、权限控制表(Authorize Control)、数据库访问权限表(Database Access Authorization)、数据库基本表(Database base-table)、MIS子功能模块表(Sub-Function Module table)、功能模块访问权限表(Function Module Access Authorization Table)、模块菜单表(Module MenuTable)。

```
create table Users
(users name varchar (10) primarykey
password  varchar (8)
)
create table Roles
( roles name varchar ( 20 ) primarykey
roles explanation varchar ( 20 )
)
create table Authorize Control
( roles name varchar ( 20 ) primarykey
 users name varchar ( 10 ) primarykey
)
create table Database basic table
( basic table name varchar (15 ) primarykey
basic table explanation varchar (30)
)
create table Database Access Authorization
( operate authorization varchar (8)    primarykey
roles name  varchar ( 20 ) primarykey
basic table name  varchar ( 15 ) primarykey
 authorization explanation varchar ( 20 )
)
 create table MIS Sub-Function Module table
(sub-function code varchar ( 20 )    primarykey
sub-function explanation varchar ( 30 )
)
 create table Module Menu Table
( module menu name varchar ( 15 )    primarykey
sub-function code varchar ( 20 )
function explanation varchar ( 60 )
)
 create table Function Module Access Authorization Table
( sub-function code varchar ( 20 )    primarykey
 roles name  varchar ( 20 ) primarykey
 module menu name varchar ( 15 )    primarykey
 function explanation varchar (60)
)
```

以上各表联合起来就形成了系统的授权机制。其运行机制是：当用户通过系统登陆界面后，系统首先利用权限控制表找出用户所对应的角色，然后利用数据库访问权限表和功能模块访问权限表找出该角色所能进行的操作（即基本表单名和子功能项代码），并在角色表中查看该角色的状态，如果角色处于被激活的状态，该用户就可使用该系统，如果该角色处于被禁止的状态，即使该用户拥有该权限仍不能执行。这样，系统管理员在进行维护时，若更改功能角色，则只需修改数据库访问权限表，功能模块访问权限表和权限控制的表中的记录即可，而不需修改表的结构，这样大大方便了管理员的维护工作。

4 结束语

本文将基于角色的访问控制应用到了MIS，结合某公司的MIS系统给出了具体的实现方法，基于角色的多层应用系统安全控制有效地将前端的“菜单层的权限管理”和后台的“数据库基本表及相应操作权限”统一起来，在实践中有一定的通用性。

参考文献

1 朱红，冯玉才，MIS系统的授权管理[J]，计算机工程与应用，1999，35(3)：72-74。

2 施景超等，基于角色的存取控制及其实现[J]，计算机应用研究，2000，(6)。

3 Sandhu R.Role-based Access Control, Laboratory for information Security Technology, ISSE, Department, MS 4A4, George Mason University 1997.