Improvements and implementation of the permission system based on RBAC model

Zhenrong Deng*, Xingxing Tang, Chuan Zhang, Xi Zhang, Wenming Huang

Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

Received 14 May 2014, www.tsi.lv

Abstract

Role-based access control as a traditional access control (discretionary access, mandatory access) is a promising place to receive widespread attention. Systematic researches on RBAC models, on one hand, this paper combined with the characteristics of Electronic government affair information management system and added regional filter function to the core RBAC model, besides, the research developed by J2EE framework and this paper presents a high availability and extensibility of RL-RABC competence management system.

Key words: RBAC, J2EE, permission system

1 Introduction

Along with the rapid development of modern computer and the Internet in recent years, information technology to infiltrate all walks of life, the electronic government affairs information management system obtained high speed development. Although the information management system in e-government brought convenience to people's life and work, the information security problems should be considered. And it not only endanger an individual interests, but also, at the higher level, involves the government and the national security. Access control as an important function of information system security component, its main purpose is to combat the threat of unauthorized operations involving computer or communication system, these threats can be subdivided into the unauthorized use, disclosure, modification, destruction and denial of service, etc. [1,2]. Role-based Access Control (Role -based Access Control, RBAC) model can effectively implement organization security policy, RABC model greatly alleviate the resource of permissions management problems [3], with the Role of interventional make authority management is more flexible and convenient.

In recent years, people put forward a lot of to improve the model of RBAC model [4-8], but the electronic government affairs information management system is usually diverse from territory. For example, we regard municipal level staff as a user of this system, another county level staff is also a role of this system, they should have the same operation permissions based on the electronic government affairs system module, meanwhile, under the jurisdiction of the municipal agency for is for the whole city area personnel management, and only at the county level staff for his county personnel management, which requires our rights management system can carry on the limits to the regional data access. However, these models can't solve the problem, this paper focuses on the characteristics of the electronic government affairs information management system, in this paper, on the basis of the classical RABC model, join the regional limit, put forward a general permission system adapted to the electronic government affairs system model (RL - RBAC).

Current RBAC research mainly focuses on the theoretical research, but lack of the concrete implementation. It restricts the use of research results in engineering practice. In view of the above problems, this paper put forward theoretical model at the same time, the model is given in the current popular open source framework of J2EE implementation method, and implements an easy expansion, easy to use and versatility of adapting to permissions in J2EE system.

2 RBAC model and improved

2.1 THE BASIC MODEL OF RBAC

Role based access control model is put forward by Ferraiolo et al. [9], through continued efforts, RBAC community members in RBAC in February 2004 by the United States standard committee (ANSI, the American National Standards) and IT (INCITS) international Standards committee accepted as ANSI INCITS359-2204 [10]. Basic RBAC model as shown in Figure 1:

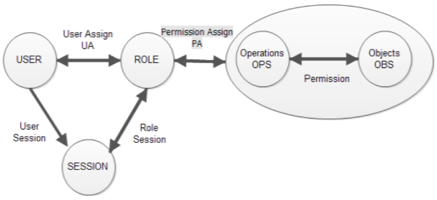

FIGURE 1 Core RBAC figure

* ***Corresponding author*** e-mail: zhrdeng@guet.edu.cn

RBAC's basic theory is: gives the user role, and its function will be authorized permissions to roles rather than users. By the user role authorization, a user can be given multiple roles, a role can have multiple permissions, with permissions can be given multiple roles, is also a many-to-many relationship between roles and permissions, access permissions and roles are linked together, roles associated with the user again, achieve the logical separation of user and the access, great convenient for rights management.

Basic RBAC model includes the following parts:

1) The basic objects include *USER*, *ROLE*, *OPS*, *OBS*, *SESSION*, (users, roles, action, object, session)

2) $UA \subseteq UDER \times ROLE$, *UA* is the user role assignment, is to be assigned to users to the roles of the many-to-many relationship, a role can be assigned to multiple users, a user can have multiple roles.

3) $PA \subseteq PRMS \times ROLE$, shows that a many-to-many relationship between the authority and role, a role can have multiple permissions, a permission can be assigned to multiple roles.

4) $USER \rightarrow 2^{SESSION}$, *USER* contacts user session and the session a one-to-many relationship, a user can have multiple sessions, and a session only allow one user participation.

5) $SESSION \rightarrow 2^{ROLE}$, shows that the session and the character of a one-to-many relationship, a session can have multiple roles, the same user of a character can only corresponding to a session.

6) $2^{OPS \times OBS}$, shows the set of permissions PRMS, also can represent the operation and the object of a many-to-many relationship.

2.2 RL-RBAC MODEL

Although RABC model reduced the workload and the complexity of authorization management, due to the classic RBAC model based on role as the medium of access control, this way in the electronic government affairs information management system has significant limitations, if classic RABC model is used to implement regional limit is difficult. This paper aimed at the characteristics of the electronic government affairs information management system is put forward based on the regional limit role access control (Region Limited RBAC, RL - RBAC) rights management system, the structure of the model as shown in Figure 2:

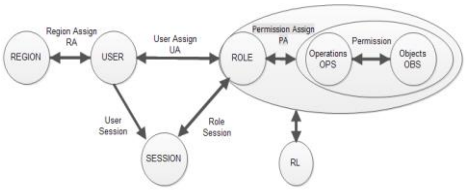

FIGURE 2 RL - RBAC model

Increases the area on the RL - RBAC in the original model defined concept, formal definition is as follows: REGION on behalf of the REGION, said users and regional many-to-many relationship, a user can be assigned to multiple areas, an area can be assigned to multiple users. After the user login system, according to the users' area, users can only to perform operations on data in the area, a user can belong to multiple regions at the same time, the user is multiple regional data and operating range, RL is the regional limit parameters.

3 System analysis and implementation

3.1 THE IMPLEMENTATION ENVIRONMENT OF RL-RBAC MODEL

During the development process of information management system, J2EE becomes most developers' first choice as its characteristics such as independence, portability, security, and more users. The paper implements a RL – RBAC model permissions system based on the J2EE architecture as the development framework as it has been greatly used. If other framework developers want to apply it in their own system, they just need to modify the permissions system slightly.

3.2 THE DATABASE DESIGN OF RL - RBAC MODEL

The entity relationship model of RL-RBAC is shown in Figure 3:

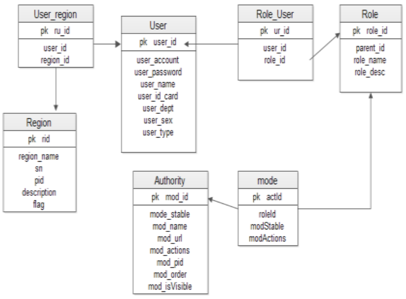

FIGURE 3 The entity relationship model of RL-RBAC

(1) The user table, mainly to describe the users' information. It is the carrier of access control, corresponding to the users of the model.

(2) Area table, used to define the limited area, corresponding to the area in the system.

(3) Area user table, mainly to define the relationship between users and area. It used to limit area for users. The field userid is used to specify the user while regionid used to specify the user area, corresponding to the area assigned in the model.

(4) Permissions module table, mainly used to define the system function entity and its permissions. It corresponding to the operation and the object of the model as it were set by the system's administrator.

(5) Role table, it mainly used to define roles, including its name, description and the superior role, corresponding to the model role.

(6) Role module table, mainly to define the relationship

between roles and system modules, as well as access to the module operation. It used to authorization for roles, corresponding to the permissions assigned in the model.

(7) Role user table, used to define the relationship between roles and users in the system. The relationship is stored in the table, corresponding to the user assignment in the model.

3.3 THE AUTHORIZATION PROCESS

According to the requirements of the electronic government information management system, the authorization process mainly includes three steps as follows.

1) Limit the user area.

Limit user area means distribution area to users: choose user area, then write the incidence relationship between users and area into the area user table.

2) Role's authorization.

Role's authorization is one of the core module of rights management system. Login the system as an administrator, choose the role which need authorize, choose the authorize menu in the system menu tree, granted to the role permissions such as select, delete, modify and so on according to requirement. Then click save menu to write the corresponding information of role and menu into the role menu table. The operation interface are shown in Figure 4, the code is as follows:

```
long roleId=Long.parseLong(req.getParameter("roleId"));
String rmActions = req.getParameter("rmActions");
boolean flag = this.roleActionsService.setting(roleId,
rmActions);
```

3) Enter into the associated modules between roles and users, choose roles assigned to users. This process is used to write the associated relationship between roles and users into the role user table.

3.4 ACCESS FILTER IMPLEMENTATION

After the user login to the system, the system first verify the legitimacy of the user by user name, password, and other ways. For legitimate users, system according to the RL - RBAC model to obtain all of the user's permissions, RL - permissions in RBAC model filter is divided into three parts to processing, to query the data of regional limit, menu display and operating limits.

When user login system, first reads the user role, read all the permissions of the user and writing session, based on the user has permissions, when users use a module during operation, the area of limited information in the form of parameters to the corresponding query statements, realize area is limited. Permissions filtration processes are shown in Figure 5:

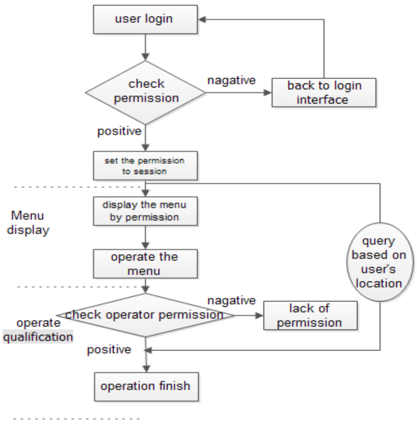

FIGURE 5 permissions filtering process

The following described the concrete implementation of access filter:

3.4.1 User authentication

This function is to validate the user's effective user authentication, is the basis of the authority system, for legitimate users, first by user ID to get all the roles assigned to the user, and then through the character ID table lookup roles module, get all the permissions of the role, the specific code implementation is as follows:

```
List<PermissionRole>                rolelist               =
this.userLoginService.getRolesByLoginUser(u.getUserId());
  /* Get to the user by user ID assigned roles */
List<PermissionRoleActions>    perRoleAlist    =    new
ArrayList();
  /* Store user permissions */
For（int i=0；i< rolelist.size;i++）
{
/* get the role */
   PermissionRole role = rolelist.get(i);
         long roleId = role.getRoleId();
/* get the user's permissions through the role ID */
List<PermissionRoleActions>                alist               =
this.userLoginService.getActionsByLoginUser(roleId);
/* added to the access list */
perRoleAlist.addall(alist);
}
/* Permissions write to the user session */
request.getSession().setAttribute(Constants.USER_INFO_A
CTION , perRoleAlist);
```

PermissionRoleActions is role of Hibernate persistence of module table, table a PermissionRole is part of the model, rolelist used to store PermissionRole types of objects, are all the logged in user's role, traverse rolelist, role module via the character ID table, insert all permissions perRoleAlist, all permissions through perRoleAlist into the session.

3.4.2 The menu display

Through access control function menu shows, is one of the main work of the authority system, function module in the information system are the form of a tree list, the main idea of this article is, first of all find out all function modules and deposited in the function module in the system list, and then traverse function module in the list of each object, compared with all of the user's permission to list, if the object in the permissions list, there are marked is proved that the user has permissions to the function module, the function returns the front desk page, according to the specific implementation code is as follows:

```
List<PermissionRoleActions> perRoleAlist =
BaseUtil.getUserAction(req); /* get permissions list from
the session user */
for(Iterator<PermissionModule> it = mlist.iterator();
it.hasNext();){
  PermissionModule m = it.next();
  boolean open = false;
for(PermissionRoleActions    ra : perRoleAlist){
if(m.getModStable().equals(
ra.getModStable())){
        open = true;
        break; }}
        if( !open ){
              it.remove();
        }
}
return this.buildTreeMenu(mlist);
}
```

List mlist deposit all the function modules of the system, first of all, get all the permissions through user session, all of the user's permissions are deposited in the permissions list module named perRoleAlist, check the permission of a function module PermissionModule by traverse the perRoleAlist, getModStable () function gets the function module a unique identifier, traverse perRoleAlist is looking for and need to check whether an object function module identifier function module identifier equal to, and it proved that the user has permissions to the function modules.

3.4.3 Operating limit

In this paper, the system is used for the user's basic operation mainly includes input, delete, modify, check, check, import, export, etc., different role requires different permissions, operating limit belongs to the fine-grained access control, the user first needs to be a qualified operation code to display the code, when the user click on the module, through the added to the front desk in advance display code, call the corresponding access control module of backend server, won the users to have all the operation of the module, if the user has the permission operation, will be the user, the operating limit access control on the server side code is given below:

```
/* Returns the menu of the logged in user's authority
information */
public static String
getUserActionByStable(HttpServletRequest req, String
stable){
List<PermissionRoleActions> permissionRoleAlist = null;
permissionRoleAlist =
(List<PermissionRoleActions>)req.getSession().getAttribut
e(Constants.USER_INFO_ACTION)
for(PermissionRoleActions ra : permissionRoleAlist){
  if(ra.getModStable().equals(stable)){
        return ra.getModActions();
}
return null;
}
```

getUserActionByStable（HttpServletRequest req, String stable） function of the parameter stable represent user access module, current from the sission permissionRoleAlist stored all of the user's permissions, by iterating through permissionRoleAlist find stable corresponding module, and then returned to the front desk page processing.

3.4.4 Area limit

Area limited is a very important part of the authority system, the user can operate the data is in the user area within the scope of qualified, first through the user ID assigned to the user of all regions. Data query area will be added to the statement, and have more than one area of the users search multiple regions and sets. The specific code implementation is as follows:

```
List<PermissionRegion> perRegionlist =
this.userLoginService.getRegionByLoginUser(
u.getUserId());
List<Region>    regionlist =new ArrayList();
Map<String,Object> searchmap=new HashMap();
for（int i=0；i< perRegionlist.size;i++）{
Region region = perRegionlist.get(i);
long regionId= region.getId();
region = this.userLoginService.getRegionById(regionId);
regionlist.add（region）;}
search.put(“regionList”, regionlist);
IGridModel gmodel =
houseFamilyService.pageQuery(gModel,map);
```

Query users by user ID set into the area perRegionlist, by iterating through perRegionlist query user area set, to which the user belong the regionlist all areas is the user's collection, the last area collection list regionlist deposited in the storage container map query conditions, in the query is executed, the area is limited to join query.

4 Conclusion

Area limit is very important part of the authority system, this paper combined with the characteristics of the electronic government affairs information management system, to improve the classic RBAC model, is proposed based on region (RL – RBAC) limit of RBAC model, classic RBAC model by solving problems on the regional limit, and improve the model at the same time, this article also discusses in detail the implementation steps of the model based on J2EE architecture, the model has good generality and now this model has been applied in civil affairs medical rescue system real-time [11], the urban planning information system and urban and rural low-income family economic conditions of water use and water verification system, the practice shows that this model can solve the problems of the e-government system of access control well.

Acknowledgment

This work is supported by Guangxi key Laboratory of Trusted Software (No: kx201317), by the Postgraduate's Innovation Project of Guilin University of Electronic Technology under (No: XJY2012017), and also supported by the Research Projects of Education Department of Guangxi Province (No: 2009MS1195).

References

[1] Smeureanu I，Diosteanu A 2010 Knowledge Dynamics in Semantic Web Service Composition for Supply Chain Management Applications *Journal of Applied Quantitative Methods* **5** (1) 1-13

[2] Shravani D，Suresh P V，Padrnaja B R 2010 The Web Services Security Architectures Composition and Contract Design Using RBAC *International Journal on Computer Science and Engineering* **8**(2) 2609-15

[3] Sandhu R S 1996 Role-Based Access Control Models *IEEE Computer* **29**(2) 38-47

[4] Zhang L H, Ahn G-J, Chu B-T 2001 A rule-based framework for role-baseddelegation *Proceedings of the 6th ACM Symposium on Access Control Models and Technologies* 2001 153-62

[5] Wainer J, Kumar A 2005 A fine-grained, controllable, user-to-user delegation method in RBAC *Proceedings of the 10th ACM Symposiumon Access Control Models and Technologies* 59-66

[6] Si W, Zeng G, Cheng Q 2006 The fine-grained expansion and application of RBAC model *Computer science* **33**(4) 227-80

[7] Zhai Z D 2006 Quantified-role based controllable delegation model *Journal of Computers* 2006 **29**(8) 1401-7

[8] Li H, Guan K 2011 The information terminal kernel model based on RBAC *Computer science* 2011 **38**(11) 100-3

[9] Ferraiolo D F, Sandhu R S, Gavrila S 2001 Proposed NIST standard for role-based access control *ACM Transactions on Information and System Security* **4**(3) 224-74

[10] ANSI INCITS 359-2004: American National Standard for Information Technology–Rolebased Access Control 2004

[11] Deng Z 2010 A real-time medical assistance billing system *International Conference on Intelligent Computing and Integrated System (ICISS 2010)* 757-60 October 2010 Guilin China

Authors

Zhenrong Deng, born on July 2, 1977, Guilin, China

Current position: Associate professor of computer science and engineering.
University studies: M.S. degree in computer science and technology in Guangxi University, China, in 2005.
Scientific interest: Grid computing, data mining, trustworthy software.
Publications: 28.

Xingxing Tang, born on February 26, 1988, Beijing, China

Current position: machine learning researcher in qunar.com.
University studies: computer science master in Guiling university of electronic technology.
Scientific interest: machine learning, reinforcement learning and computer vision.
Publications: 2 papers.
Experience: machine learning in qunar.com.

Chuan Zhang, born on July 25, 1988, Guilin, China

Current position: M.S candidate in Guilin University of electronic technology.
University studies: the technology of computer in Guilin University of electronic technology.
Scientific interest: cloud computing, data mining, software engineering.
Publications: 2 papers.
Experience: worked at the largest Chinese search engine company Baidu Inc.

Xi Zhang, born on August 4, 1989, Guilin, China

Current position: M.S candidate in Guilin University of electronic technology.
University studies: the technology of computer in Guilin University of electronic technology.
Scientific interest: recommender systems and data mining.

Wenming Huang, born on July 8, 1963, Suzhou, China

Current position: professor of computer science and engineering, director of department of software engineering.
Scientific interest: grid computing, image processing, software engineering.
Publications: 50.