A Novel OA System Access Control Method Based on Improved RBAC Model

Xinze Guo\(^1\)*, Chuanchang Liu\(^1\) and Zhiyuan Su\(^2\)

\(^1\)State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
\(^2\)School of Automation, Beijing University of Posts and Telecommunications

*Corresponding author e-mail: guoxinzejava@163.com

Abstract. Access control is an indispensable part of the OA system. Reasonable access control can fundamentally guarantee data security. With the expansion of enterprise business scale and the adjustment of organizational structure, there will be more and more system users. Due to the sensitivity and privacy of business data, it is necessary to manage the authority of users in this system. In order to solve the problem of large granularity of permission management and strong coupling between access control code and business code, this paper compares the implementation scheme of existing access control system, realizes a fine-grained access control method based on improved RBAC model and achieves good results in practical system application.

1. Introduction

Intense market competition requires that enterprise offices not only be automated, but also capable of delivering information at high speeds across regions, and the vigorous development of computer network technologies and databases provides support [1]. The OA system is a complex human-computer interaction system. Every specific link may be threatened by vicious attack. It is very important to build strong access control to ensure the security of information. In the OA application system, the navigation menu serves as an interface for the user to perform system operations. In addition to the basic features of simplicity, flexibility, and convenience, it is also the first barrier to system security and should be able to identify the user's identity as well as prevent the intrusion of illegal users. Therefore, how to provide users with a safe operating environment is a key issue that designers have to solve.

Since the 1970s, research on secure access control has received widespread attention. The RBAC model proposed by George Mason University in 1996 is widely accepted and applied due to its flexibility and relative ease of maintenance [2].

Based on the RBAC model theory and combined with the actual enterprise requirements, this paper designs and implements a set of fine-grained access control schemes, which realizes the control of page buttons and data filtering. Through the AOP technology of SpringMVC framework, this paper concentrates the logic of access control in the local code of application system, realizes decoupling from business code, reduces duplication of development, and improves system robustness and maintainability.
2. **Comparison of existing access control models**

At present, there are mainly three access control models: Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-Based Access Control (RBAC). Each has advantages and disadvantages [3].

2.1. **DAC**

For this model, the same user has different permissions for different resource objects (the smallest unit of access control). Different users have different permissions to the same resource object. Users can grant other users the permissions they have without any restrictions. The advantage of this model is that the assignment of permissions is very flexible. The disadvantages are lack of security and complex permission management.

2.2. **MAC**

For this model, each resource object is marked with a corresponding level of confidentiality, and each user is also granted a level of access license. Only users with licenses can access the corresponding secret objects. The user license and resource object secret levels are managed by the system administrator. The advantage of this model is that the permissions relationship has a good hierarchical structure, the access control is strict, and the disadvantage is that the flexibility is poor.

2.3. **RBAC**

The concept of role is introduced in the RBAC model, which realizes the decoupling of users and permissions and improves the reusability of permissions. The model mainly defines three entities: user, role, and authority, as well as the concept of session and restriction and the relationship between them [4]. In this model, permissions are assigned to roles and roles are assigned to users. A user can have multiple roles, one role can be granted to multiple users, one role can contain multiple permissions, and one permission can be included by multiple roles. A user is not directly associated with a permission but through a role. The permission to operate on a resource object is achieved through an active role.

For the RBAC model, the role is a relatively stable concept. There is a many-to-many relationship between user and role, and the relationship can change rapidly, that is, the role that the user plays can change frequently. There is also a many-to-many relationship between role and authority, but the change in this relationship is relatively slow, that is, the role has a relatively stable authority. Using this model, it is possible to achieve perfect access control through two steps: user authorization (assigning a role to the user) and role authorization (allocating operation authorities to the role), which is both flexible and easy to maintain, so it is widely used [5]. When the user changes more and the role changes relatively less, the role is relatively stable and easy to maintain. However, in large enterprises, there are many roles, and the inheritance relationship between roles is also complicated. This model makes the algorithm of access control implementation more complicated.

![Figure 1. RBAC model.](image-url)
3. Features of OA system access control

Compared with other application systems, the OA system has the following features in terms of access control: the number of users is large, including the personnel of various departments of the enterprise; the employees are transferred in and out frequently; the job position and job duty change rapidly; the amount of data that the system has to process is large, mostly sensitive data, and for security considerations different data need to be distributed to different users for processing. At present, the access control of OA systems often cannot meet the needs of enterprises. Although some OA system access control adopts the management idea of RBAC in part, it still has the following disadvantages [6, 7].

- It is impossible to describe complex security policies. The system structure design is not flexible enough from the beginning, and the authorization granularity is large, which cannot meet the requirements of fine-grained access control.
- It is difficult to manage a large number of users effectively. Due to the large number of users, the access requirements to the system are different, resulting in increased management difficulty, and errors such as permission conflicts and unauthorized access. Therefore, in order to fundamentally solve the problem of access control, the entire system must have a more appropriate and fine-grained authorization granularity, and the access control must have high reliability and flexibility.

4. Design description

Based on the analysis of the advantages and disadvantages of the comprehensive access control model, combined with the features of the OA system access control, it is decided to implement a access control model based on RBAC with clear hierarchy, flexible authorization, granular refinement and easy maintenance. The difficulty of access control is that the design of it needs to meet the fine-grained authorization requirements, on the other hand, it needs to have certain flexibility and easy maintenance to cope with the changing enterprise environment. First of all, the meaning of permission is defined. The permissions refer to resources within the web application, including URLs, operations, page elements, database instances, and so on. The permission is a tree-like structure, so it is designed to be one-to-many relationship.

For fine-grained access control, the concept of operation set and scope set is introduced. The action set is a collection of user-to-page operations, including view, edit, and edit individual. The view represents that users have permission to view the details of the data that can be seen. The edit represents that users have permission to edit the data that can be seen. The edit individual represents that users only have permission to edit the data created by themselves. The scope set is the collection of user’s data permissions, which is divided into head office, branch, department, and individual. The scope set and the operation set are used together for filtering of user data list. More specifically, the scope set is a collection of departments to which the creator of the data belongs that the user can see. Given the flexibility of permission design, users can see data from multiple departments. For example, if the department assigned by the user is department A and department B, and the operation is edit individual, the user can view all data of departments A and B, but can only edit the data created by himself.

The development process of the OA system access control can be seen as the process of establishing the system page resource tree. The OA system can generally be divided into several functional modules, each of them is essentially a collection of page files with specific functions, so user's permissions are directly related to page [8]. The logic control in the program is when the user logs in, first verify the password. Then get its authorized pages and corresponding operation set and scope set information on each page by their account. Finally display the user a navigation tree that can be manipulated, which contains links to pages. When the user accesses the page, the HTTP request is sent to corresponding interface, and the backend system uniformly intercepts the request for authentication.

5. Design

According to the design idea of the access control method, an instance is developed and successfully applied in the actual project.
5.1. Functional module design
Design of access control includes an authorization module and an authentication module. The authorization module includes user management and role management. The user management includes create, retrieve, update, delete, and role assignment of user. The role management includes create, retrieve, update, delete, and permission assignment of role. The authentication module uses the interceptor of the SpringMVC framework, which can intercept HTTP requests by URL relative to the filter of Servlet and have more flexibility.

5.2. Database design
There are mainly four types of database tables involved in access control in the system.
- User table, which is used to save user related information.
- Role table, which is used to save role related information.
- Authority table, which is used save authorities of the system.
- Department table, which is used to save department information.
- User-role table, role-authority table, role-authority-department table, which is used to save relations between tables.

5.3. Entity design
Hibernate is used as an ORM framework in development. Mapping relationships between database tables and entity classes are managed by JPA annotations. There are five entities involved in access control as following: User, Role, Authority, Department, RoleAuthority. User entity and Role entity respectively implement user management and role management. RoleAuthority entity implements the assignment of permissions for a role. Figure 2 shows their relationship.

![Figure 2. Static class diagram of access control model.](image-url)
6. Implementation

6.1. Authorization
Administrator assigns permissions to roles through role management page as well as create accounts and assigns roles to users through the user management page.

6.2. Login
In order to prevent users from accessing the system by inputting URL of internal page in the address bar of browser, all requests to the system are uniformly intercepted by the interceptor. User's validity is verified by querying the user database table. If account does not exist or the password is incorrect, redirect request to the login page. For legitimate user, user account is saved in the session attribute, and cookie attribute of the browser is set.

6.3. Navigation tree generation
After the user logs in successfully, permissions is read from the authority database table, and organized into a navigation tree recursively. The tree contains links of the pages that the user can operate and the operation permissions of the page. At the same time, because access control is refined to the button, users can only see buttons they have been authorized. Different interfaces are displayed to different users, and administrators can see all the pages. Others can only see the pages they are authorized to.

6.4. Authentication
When the user accesses the page, the HTTP request is sent to the background system. If the URL satisfies the intercepted URL rule configured in the SpringMVC configuration file, the request will first be processed by SpringMVC's HandlerInterceptor. If the user is administrator, all requests are released. Otherwise, the request URL is first obtained, and then the permission set of the user is obtained through querying cache. If request URL belongs to the permission set, the request is allowed to access the corresponding background interface with operation permission and department permissions. Otherwise return no permission prompt to the user. In addition, in order to prevent the user from illegally accessing the background interface by splicing the URL, the interface is identified by the @RequestMapping annotation of SpringMVC. Interfaces that can be used to modify resources can only be accessed by HTTP POST request, others are accessed by HTTP GET request. If the user is assigned view operation under a certain page, then only HTTP requests with a request type of GET are allowed in this page range.

6.5. Data screening
The user can access background interface after authentication, but the fine-grained access control requires the user not to see all the data. So data needs to be filtered before acquired, and only data that the user has permission to see is retrieved. Firstly, the department permission set to which the user is assigned is obtained, and then use it as a query condition to retrieve data. Only if creator of a record belongs to one of the department in department permission set, then the record is retrieved.

7. Conclusion
For OA system, designing a flexible, reliable access control is of great significance. Under the guidance of RBAC model, combined with the AOP technology, this paper designs and implements a OA system with high security, fine granularity and convenient maintenance. It effectively prevents the illegal access to resources and realizes access to resources refined to button level and data level. The method proposed by this system has been applied to the development of a project and has achieved good operational results.
Acknowledgments
The work was supported by the National Natural Science Foundation of China (Grant No.U1536112) and Special Funds for Public Industry Research Projects of State Administration of Grain of China (Grant No.201513002).

References
[1] Sandhu R S, Samarati P. Access control: Principles and practice [J]. IEEE Communications, 1994, 32(9):40-48.
[2] Mario Storga. XML-based Web Service for Collaborative Product Data Management [EB/OL]. http://www.ice2002.org, 2002.
[3] Rick Kuhn. Role Based Access Control [EB/OL]. http://csrc.nist.gov/rbac/rbac-std-ncits.pdf, 2003.
[4] Demin Feng, Xiaoming Wang, Zongtao Zhao. An Extension Role Retrieve Control Model [J]. Computer Engineering and Applications, 2003; 39(3):87-89.
[5] Xianhu Meng, Junyi Shen. WEB Database Application Security Implementation [J]. Computer Engineering and Applications, 2002; 38(9): 199-200.
[6] Zhang Guangting, Lu Wei, Li Guizhi. Role-based OA System Access Control Model and Its Implementation [J]. Small Computer Systems, 2002, 23(8): 985-988.
[7] Cao Tianjie, Zhang Yongping. Role-based Access Control in Management Information System [J]. Computer Applications, 2001, 21(8): 21-23.
[8] Ouyang Xingming, Zhang Huazhe. Role-based Authority Management in Large-scale Network MIS System [J]. Computer Engineering and Applications, 2000, 4: 138~140.