A Role-Based Access Control Model that Supports Authorization with The Combination of Static and Dynamic

Liangyu Li[1, a],Yuanning Liu[1, b], Xiaodong Zhu[1, c,*], Biao Huang[1, d] and Youwei Wang[1, e]

[1] College of Computer Science and Technology, Jilin University, Jilin Changchun 130012, China

[a]641491833@qq.com, [b]lyn@jlu.edu.cn, [c]zhuxd@jlu.edu.cn, [d]476738323@qq.com, [e]1036569449@qq.com

Keywords: RBAC, Coarse-granularity, Fine-granularity, The combination of static and dynamic.

Abstract. The traditional Role-Based Access Control(RBAC) model[1] is widely used in enterprises, but it's control granularity is too single and it's permissions configuration is too complex. This paper introduces the concepts of coarse-granularity and fine-granularity[2], and puts forward a RBAC model that supports authorization with the combination of static and dynamic. In this paper, we first show the overall structure of the model, then divide the model into two parts of permissions configuration and permissions control to describe. In each part, we introduce the related basic elements and design the algorithms. Finally, this paper shows the application based on this model. And tests shows that the model is a good solution to these problems and has good feasibility and effectiveness.

1 Introduction

When we were conducting research on the needs of enterprise, such a problem was found, that an user can act as different roles, so he can operate different components or receipts within a windows form. Obviously, the amount of the components and receipts of an information system is huge. The permissions configuration will be a very heavy work and be unreliable if the system administrator configures permissions separately for each component or receipt, so this grained permissions configuration is required to be simple and accurate. In addition, the operation on a fine-granularity control unit not only needs the fine-granularity permissions, but also the related coarse-granularity permissions.

In summary, there are some shortcomings of the traditional RBAC models' permissions configuration, on the one hand, the assignment of roles is set completely by the administrator, so it is difficult to ensure the correctness[3], on the other hand, it can only do access control in the single granularity. The proposed model in the article[4] can do access control in different granularities. The article[5] proposes a trust-based model, and implements a dynamic assignment of user's roles. But they ignore two important tissues, that the simplicity and accuracy of access control and the convergence between multi-granularity. The proposed model in this paper is based on the idea of RBAC, and uses the authorization mode with the combination of static and dynamic, not only avoids the tedious work of permission configuration, but also is flexible enough to deal with the special circumstances.

2 The overall design of the model

In Fig.1, the route with 'a' shows that fine-granularity access control will inevitably involve coarse-granularity access control. This is the reason why we pay attention to the convergence between different control granularity.

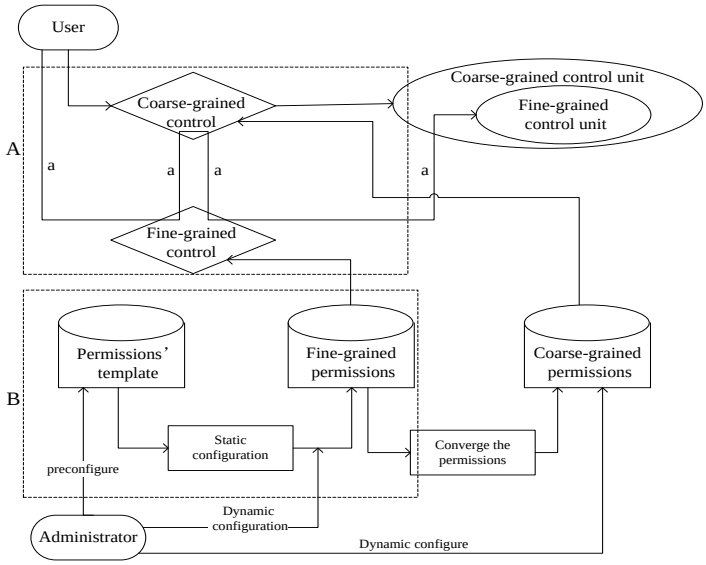

Fig.1 the model's structure

The portion A drawn by the dotted line, is the main part of the convergence between the different control granularity and shows that the model will access the coarse-granularity permissions when it generates a fine-granularity permissions template. The portion B drawn by the dotted line, shows that the model will selectively uses a template to achieve the purpose of authorization with the combination of static and dynamic according to user's operation.

2.1 The algorithm that achieve the convergence between the coarse-granularity and fine-granularity access control

Before giving the algorithm, we first introduce some concepts:

- *U:* In the RBAC model, we define $U=W_U \times P_U=\{ u_1,u_2,\cdots,u_i,\cdots,u_n \}$ as the set of users. We use $w_u \in W_U$ to represent the user's worker ID and use $p_u \in P_U$ to represent the user's other identity information, so that we can use a two-tuples (w_u,p_u) to represent uniquely this user.
- *M:* We define $M=\{ m_1,m_2,\cdots,m_i,\cdots,m_n \}$ as the set of coarse-grained permissions. The form of $m_i \in M$ is (mID,u), $u \in U$, and the *mID* identifies a coarse-grained control unit.
- *D:* We define $D=\{ d_1,d_2,\cdots,d_i,\cdots,d_n \}$ as the set of fine-grained permissions. The form of $d_i \in D$ is (dID,dOp,mID,u), and the *dID* identifies a fine-grained control unit, and the *dOp* represents the operation type information of the control unit.

The algorithm as follows:

1) Get the current user U_c and the current fine-grained control unit ID D_c.

2) For each two-tuples (mID,u) in the set M, get these which meet the condition that $u=U_c$, and record as M_1.

3) For each (dID,dOp,mID,u) in the set D, get all two-tuples (mID,u) which meet the condition that $u=U_c$, and record as D_1.

4) So the set of coarse-grained permissions P is the union of M_1 and D_1, that is $P=M_1 \cup D_1$.

5) For each (dID,dOp,mID,u) in the set D, get these which meet the condition that $(u = U_c)\&\&(dID = D_c)$, and record as D_2.

6) So the set of fine-grained permissions is D_2.

As we can see, some coarse-granularity permissions derive from the related fine-granularity permissions, which can ensure that once the user have the permission to access the fine-granularity control unit, he will gain the permission of related coarse-granularity control unit, so we can access the fine-granularity control unit smoothly, which solves the problem that the incoherence between the coarse-granularity access control and the fine-granularity access control.

2.2 The idea and algorithm of authorization with the combination of static and dynamic

The model's permissions configuration is based on the idea of authorization with the combination of static and dynamic.

Usually, the dependence between a coarse-grained control unit and a fine-granularity control unit is always fixed, but the operator and operation type of a fine-granularity control unit are not fixed. It's fixity prompts us to use the permissions templates. We use the following formula to represent the process of authorization with the combination of static and dynamic. At first, we define an operation * to achieve the following, that a new equal-dimension vector will be obtained, if we multiply the two elements *a* and *b* that come from the same position of two equal-dimension vectors, the formula as follows:

$$\vec{w} = \vec{m} * \vec{n} \tag{1}$$

if $\vec{m} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_k \end{bmatrix}$, $\vec{n} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_k \end{bmatrix}$, then $\vec{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix} = \begin{bmatrix} m_1 \times n_1 \\ m_2 \times n_2 \\ \vdots \\ m_k \times n_k \end{bmatrix}$, so we can use the following vector to formalize the idea of the combined authorization of static and dynamic ,

$$\vec{P} = \vec{a} * \vec{P}_T + b \times \vec{P}_U \tag{2}$$

The $\vec{P}$, $\vec{P}_T$ and $\vec{P}_U$ are equal-dimension vectors, and the same position's elements of this three vectors represent a same kind of permissions. The vector $\vec{P}$ represents the permissions which will take effect, and the vector $\vec{P}_T$ represents the static permissions which are generated by using the permissions template, and the vector $\vec{P}_U$ represents the dynamic permissions which are generated by users' configuration. The dimension of the vector $\vec{a}$ is the same with the vector $\vec{P}$ and its elements can only be 0 or 1. The *b* is a variable and its value can only be 0 or 1.

When $\vec{a} \neq \vec{0}$ and *b=0*, then $\vec{P} = \vec{a} * \vec{P}_T$, it indicates that using the permissions template merely, we call it static configuration mode;

When $\vec{a} = \vec{0}$ and *b=1*, then $\vec{P} = \vec{P}_U$, it indicates that configuring permissions without using template at all, we call it dynamic configuration mode;

When $\vec{a} \neq \vec{0}$ and *b=1*, then $\vec{P} = \vec{a} * \vec{P}_T + \vec{P}_U$, we call it the mode with the combination of static and dynamic. The users can adjust the vector $\vec{a}$ and the vector $\vec{P}_U$ to generate vector $\vec{P}$ flexibly.

Before we show the algorithm which is used to read the template's permissions, we first introduce some concepts:

- *R:* We define $R=\{ r_1,r_2,\cdots,r_i,\cdots,r_n \}$ as the set of roles. Each $r_i \in R$ represents a role of the user belongs to. And we define $RPU=\{ rpu_1,rpu_2,\cdots,rpu_i,\cdots,rpu_n \}$ to represent the relationships between users and roles, the form of $rpu_i \in RPU$ is (r,p_u), r ,$p_u \in P_U$.
- *T:* The set of permissions template is a pre-configured table of the relationships between roles and permissions. We define $T=\{ t_1,t_2,\cdots,t_i,\cdots,t_n \}$ as the set of permissions template. The form of $t_i \in T$ is *(dOp,mID,r)*.

A fine-granularity control unit is dependent on the presence of a coarse-granularity control unit. Each operation on the fine-granularity control unit must involve a group of parameters (dID_i,dop_i,mID_i) and a related coarse-granularity control unit.

The algorithm as follows:

1) For each *(dOp,mID,r)* in the set *T*, get all *r* which meet the condition that $(dOp=dop_i)\&\&(mID=mID_i)$, and return the result R_k, that $R_k=\{ r_1,r_2,\cdots,r_i,\cdots,r_k \}$.

2) For each (r,p_u) in the set *RPU*, get all pu_i which meet the condition that $r=r_j$, $r_j \in R_k$, $j=(1,2,3\cdots k)$ and return the result PU_y, that $PU_y = \{ pu_1, pu_2, \cdots, pu_i, \cdots, pu_y \}$.

3) For each (wu_i,pu_j) in the set *U*, get all *u* which meet the conditions that $pu=pu_j$, $pu_j \in PU_y$, $j=(1,2,3\cdots y)$ and return the result U_Z, that $U_Z = \{ u_1, u_2, \cdots, u_i, \cdots, u_z \}$.

4) Insert each (dID_i,mID_i,dop_i,u_x) into the set *D*, $u_x \in U_z$, $x=(1,2,3\cdots z)$.

The use of permissions template not only simplifies the permissions configuration, but also is flexible enough to modify the result of static permissions.

3 Application and Conclusion

The invoicing system of company X is an implement of this model, the access control of the system depends on the two parts, that permissions configuration and permissions control. The permissions configuration use the combined mode of dynamic and static, and the permissions control makes the control between forms and receipts more coherent. We can show the improvement in the following table:

Model Version	Accuracy	Configuration Simplicity	Support multi-granularity
Traditional RBAC model	low	complex	yes
This paper's model	high	simple	no

4 Summary

On the basis of the traditional RBAC model, the model proposed in this paper is based on the subordinate relationship between the coarse-grained control unit and the fine-grained control unit and uses the permissions template, finally realizes the combined authorization of static and dynamic. And this model successful solves the problems, that the complex configuration and the incoherence between the different grained control unit. In future, our study will focus on the following points:

1. Continue to optimize the dynamic configuring of permissions and reduce the workload of the users' configuration.
2. Make the control granularity refine to a part of a component, such as a component including multiple columns of data, we request some users can't see all the columns of data.

References

[1]Ravi S Sandhu,et al.Role-Based Access Control Models[J].IEEE Computer,1996,29(2):38-47.

[2]Zhao Xia, Huai Jin-peng. XML-Based Multi-Granularity Access Control System[J].Computer Engineering and Applications ,2002,21:155-159.

[3]FU Xiang-ping,WU Zhen-qiang,YANG Bei. Trust-degree and attributes based RBAC authorization model[J]. Application Research of Computers,2011,02:742-745.

[4]YE Chun-xiao, FU Yun-qing, WU Zhong-fu, LI Yun. Access Control Based on Multi-granularity Privileges[J].Application Research of Computers,2004,10:87-89.

[5] CHAKRABORTY S,RAY I.TrustBAC: integrating trust relationships into the RBAC model for access control in open systems［C］//Proc of the 11th ACM Symposium on Access Control Models And Technolo-gies.New York: ACM Press,2006: 49-58.