



## Layer Model and Algorithm of Organization Authorization Based on Position Network

Feng Ding (Corresponding author)

Institute of Information and Decision Technology

Dalian University of Technology

Dalian 116023, China

E-mail: dingff@gmail.com

Yanzhang Wang

Institute of Information and Decision Technology

Dalian University of Technology

Dalian 116023, China

E-mail: yzwang@dlut.edu.cn

Xin Ye

Institute of Information and Decision Technology

Dalian University of Technology

Dalian 116023, China

E-mail: yejstar@163.com

*This work was partially supported by the National Graduate Student Program of Building World-Class Universities (Grant No.[2007]3020)*

### Abstract

After the RBAC model formalized in 1992 by David Ferraiolo and Rick Kuhn, the models of organization authorization have been developing fast. Among these models, the one based on position network will be used widely and has a prospective future. Algorithms for single-layer model and multi-layer model based on position network were cited in this article. Meanwhile an algorithm example was given at the end of this article.

**Keywords:** Position, Network, Organization, Authorization, Layer, Model, Algorithm

### 1. Preface

In modern complex government systems, there are too many roles to be setup and provided to the users. If the duties or privileges of users are changed, or the objects in system are changed, it will take lots of time to adjust the relationship of User-Role and Role-Object. The integration of different government application systems has been affected by this bottleneck. For improving the efficiency and security and reducing the workload of system maintenances, "Position" was added between the "User" and "Role". This makes it more close to the reality of Chinese government.

After turning this into reality, we found that a straight and effective algorithm was needed to evaluate the rationality of the authorization process, make sure the strictness, and find out the problems. So the single-layer model algorithm and multi-layer model algorithm for organization were put forward.

### 2. Algorithm for Organization Authorization System

We can easily connect the users, positions, roles and objects in different systems with network model. Show as figure 1.

How to look at the connection clearly and check out whether it is right? We need to setup a kind of model system to evaluate the feasibility of the connection.

According to the relationship of the “user” ontology, the “users” in the e-government systems have the character of social network. Meanwhile the “position” ontology also has the same character. For making the description and expression of the character clearly, we describe the algorithm as follows.

### 2.1 Single-layer Model Algorithm for organization authorization system

Because of the complexity of the government, the mapping of user ontology and position ontology is not one-to-one. It is multi-mapping. Show as figure 2.

Define a set of user ontology:

$\{user_1, user_2, \dots, user_m\}$ , recorded as vector  $U = [user_1, user_2, \dots, user_m]^T$ ;

Define a set of position ontology:

$\{pos_1, pos_2, \dots, pos_n\}$ , record as vector  $P = [pos_1, pos_2, \dots, pos_n]^T$ ;

We record the relationship  $user_m \rightarrow pos_n$  of  $user_m$  and  $pos_n$  as  $u_{mn}$ , and describe it with the form  $U \cdot P^T$ :

$$\begin{aligned}
 U \cdot P^T &= \begin{bmatrix} user_1 \\ user_2 \\ \dots \\ user_m \end{bmatrix} \cdot [pos_1, pos_2, \dots, pos_n] \\
 &= \begin{bmatrix} user_1 pos_1 & user_1 pos_2 & \dots & user_1 pos_n \\ user_2 pos_1 & user_2 pos_2 & \dots & user_2 pos_n \\ \dots & \dots & \dots & \dots \\ user_m pos_1 & user_m pos_2 & \dots & user_m pos_n \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ u_{m1} & u_{m2} & \dots & u_{mn} \end{bmatrix}
 \end{aligned}$$

Shorten the  $U \cdot P^T$  as  $U_p$ , then we get:  $U_p = [u_{mn}]_{m \times n}$ .

Especially, under this structure, if the only aim is to show the connection of  $user_m$  and  $pos_n$ , we can use 0 and 1 to realize it.

If  $user_m$  has no right to access  $pos_n$ ,  $user_m \rightarrow pos_n = 0$ , record as  $u_{mn} = 0$ ;

If  $user_m$  has the right to access  $pos_n$ ,  $user_m \rightarrow pos_n = 1$ , record as  $u_{mn} = 1$ .

If so, we can get a matrix like  $\begin{bmatrix} 1 & 0 & \dots & 1 \\ 1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$ , which is formed only with 0 and 1.

For keeping consistence with traditional *RBAC* model, the roles in the organization authorization system still have the definitions and attributes as in *RBAC* model. And also the roles can be formed as layer structure. It is multi-mapping between the positions and roles. Show as figure 3.

Define a set of role ontology:

$\{rol_1, rol_2, \dots, rol_i\}$ , record as vector  $R = [rol_1, rol_2, \dots, rol_i]^T$ ;

We record the mapping  $pos_n \rightarrow rol_i$  between  $pos_n$  and  $rol_i$  as  $p_{ni}$ . We describe the mapping with the form of  $P \cdot R^T$ :

$$\begin{aligned}
 P \cdot R^T &= \begin{bmatrix} pos_1 \\ pos_2 \\ \dots \\ pos_n \end{bmatrix} \cdot [rol_1, rol_2, \dots, rol_i] \\
 &= \begin{bmatrix} pos_1 rol_1 & pos_1 rol_2 & \dots & pos_1 rol_i \\ pos_2 rol_1 & pos_2 rol_2 & \dots & pos_2 rol_i \\ \dots & \dots & \dots & \dots \\ pos_n rol_1 & pos_n rol_2 & \dots & pos_n rol_i \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1i} \\ p_{21} & p_{22} & \dots & p_{2i} \\ \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{ni} \end{bmatrix}
 \end{aligned}$$

Shorten  $P \cdot R^T$  for  $P_R$ , then:  $P_R = [p_{ni}]_{n \times i}$ .

We still use the descriptions for roles and operations in *RBAC* model, and setup multi-mapping for them. By allowing this, the workload for distributing the system function is reduced a lot. Show as figure 4.

We define a set of operation:

$\{oper_1, oper_2, \dots, oper_j\}$ , record as vector  $O = [oper_1, oper_2, \dots, oper_j]^T$ ;

We record the mapping  $rol_i \rightarrow oper_j$  of  $rol_i$  and  $oper_j$  as  $r_{ij}$ , and record it as the form of  $R \cdot O^T$ :

$$\begin{aligned} R \cdot O^T &= \begin{bmatrix} rol_1 \\ rol_2 \\ \dots \\ rol_i \end{bmatrix} \cdot [oper_1, oper_2, \dots, oper_j] \\ &= \begin{bmatrix} rol_1 oper_1 & rol_1 oper_2 & \dots & rol_1 oper_j \\ rol_2 oper_1 & rol_2 oper_2 & \dots & rol_2 oper_j \\ \dots & \dots & \dots & \dots \\ rol_i oper_1 & rol_i oper_2 & \dots & rol_i oper_j \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1j} \\ r_{21} & r_{22} & \dots & r_{2j} \\ \dots & \dots & \dots & \dots \\ r_{i1} & r_{i2} & \dots & r_{ij} \end{bmatrix} \end{aligned}$$

Shorten  $R \cdot O^T$  for  $R_O$ , then:  $R_O = [r_{ij}]_{i \times j}$ .

By the matrix of the mapping of “User-Position”, “Position-Role” and “Role-Operation”, we can get the  $m \times j$  matrix  $T$  for single layer organization authorization. With  $T$ , the relationship of different parts in the system can be found out easily.

$$\begin{aligned} T &= U_P \cdot P_R \cdot R_O \\ &= [u_{mn}]_{m \times n} \cdot [p_{ni}]_{n \times i} \cdot [r_{ij}]_{i \times j} \\ &= \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ u_{m1} & u_{m2} & \dots & u_{mn} \end{bmatrix} \cdot \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1i} \\ p_{21} & p_{22} & \dots & p_{2i} \\ \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{ni} \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1j} \\ r_{21} & r_{22} & \dots & r_{2j} \\ \dots & \dots & \dots & \dots \\ r_{i1} & r_{i2} & \dots & r_{ij} \end{bmatrix} \end{aligned}$$

We record  $t_{mj}$  as the elements in  $T$ , and  $t_{mj}$  is the mapping of users and operations. So we get *Single-layer Model Algorithm for Organization Authorization System*:

$$T = [u_{mn}]_{m \times n} \cdot [p_{ni}]_{n \times i} \cdot [r_{ij}]_{i \times j} = [t_{mj}]_{m \times j}$$

If  $t_{mj} \neq 0$ , it shows that the user  $m$  has the right to get the operation  $j$ ; if  $t_{mj} = 0$ , it shows that the user  $m$  has no right to get the operation  $j$ . From this matrix  $T$ , all the operations that each user can get are shown clearly. This can help the system administrator check the rationality of each user, and enhance the reliability of the whole system.

From the above process,  $\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ u_{m1} & u_{m2} & \dots & u_{mn} \end{bmatrix} \cdot \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1i} \\ p_{21} & p_{22} & \dots & p_{2i} \\ \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{ni} \end{bmatrix}$  shows the relationship between users

and positions, recorded as:  $U_R = [u_{mn}]_{m \times n} \cdot [p_{ni}]_{n \times i}$ ;

And  $\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1i} \\ p_{21} & p_{22} & \cdots & p_{2i} \\ \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & \cdots & p_{ni} \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1j} \\ r_{21} & r_{22} & \cdots & r_{2j} \\ \cdots & \cdots & \cdots & \cdots \\ r_{i1} & r_{i2} & \cdots & r_{ij} \end{bmatrix}$  shows the relationship between positions and roles, recorded

as:  $P_O = [p_{ni}]_{n \times i} \cdot [r_{ij}]_{i \times j}$ .

$U_R$  and  $P_O$  are very helpful in the application of checking the middle layer mapping in organization authorization system.

## 2.2 Multi-layer Model Algorithm for organization authorization system

In the practice of government, a structure like pyramid has been formed between different layers and sections. And this caused that the same structure has been used in the former constructed systems. And also there are lots of “pyramid” structures in positions.

Meanwhile for the easy configuration of organization authorization system, the “pyramid” structures were formed between roles in many systems based on *RBAC*.

Because of the multi-layer structures, we need to expand the single-layer model for further detection to verify the authorization process.

For the multi-layer, we add a column vector between each two layers. Each column vector consists of the mapping of the elements of the two layers.

As to the multi-layer structure of positions, if the number of positions is  $n$ , and they have been divided into  $t$  layers, we describe each position as  $pos$ , and show the abstract description as figure 5.

We describe the whole position set as  $\begin{bmatrix} pos_1 & pos_1 & \cdots & pos_1 \\ pos_2 & pos_2 & \cdots & pos_2 \\ \cdots & \cdots & \cdots & \cdots \\ pos_n & pos_n & \cdots & pos_n \end{bmatrix}_{n \times t}$ , and describe the layer  $l$  as

$p_l = \begin{bmatrix} pos_1 \\ pos_2 \\ \cdots \\ pos_n \end{bmatrix}, l \in [1, t]$ ; if there are  $t$  layers, we can get  $t$  vectors.

We get the relationship of layer  $l$  and layer  $l+1$  with the multiplication of column vector  $l$  and the transpose of column vector  $l+1$ , and record it as  $\hat{p}_{l(l+1)}$ :

$\hat{p}_{l,l+1} = \begin{bmatrix} pos_1 \\ pos_2 \\ \cdots \\ pos_n \end{bmatrix} \cdot [pos_1, pos_2, \dots, pos_n] = \begin{bmatrix} pos_1 pos_1 & pos_1 pos_2 & \cdots & pos_1 pos_n \\ pos_2 pos_1 & pos_2 pos_2 & \cdots & pos_2 pos_n \\ \cdots & \cdots & \cdots & \cdots \\ pos_n pos_1 & pos_n pos_2 & \cdots & pos_n pos_n \end{bmatrix}_{n \times n}$  When  $pos_i$  has

some kind of relationship with  $pos_k$ , we record the value of  $pos_i pos_k, i, k \in [1, n]$  as 1. If there is no relationship, we record it as 0. So we can get a  $n \times n$  matrix which only consists of 0 and 1. Suppose there are  $t$  layers in position structure, we can get  $t-1$   $n \times n$  matrices in total. Record them as  $\hat{p}_{12}, \hat{p}_{23}, \dots, \hat{p}_{(t-1)t}$ .

Multiply these  $t-1$   $n \times n$  matrices one by one, we can get the whole position structure matrix  $\hat{P}$ :

$\hat{P} = \prod_{l=1}^{t-1} \hat{p}_{l(l+1)}, (l, n \in N, t: \text{total layers of positions})$

For the roles with abstract layer structure, figure 6 gives an example.

We use  $\hat{r}_{k(k+1)}$  to describe the mapping matrix between the two nearby role layers, then we can get the whole role

structure matrix  $\hat{R}$  :

$$\hat{R} = \prod_{k=1}^{h-1} \hat{r}_{k(k+1)}, (k, h \in N, h : \text{total layers of roles})$$

The organization authorization system structure with multi-layer positions and roles shows in figure 7.

When getting the position structure matrix and role structure matrix, we can get the *Multi-layer Model Algorithm for Organization Authorization System*:

$$T = U_P \cdot \hat{P} \cdot P_R \cdot \hat{R} \cdot R_O = [u_{mn}]_{m \times n} \cdot \prod_{l=1}^{t-1} \hat{p}_{l(l+1)} \cdot [p_{ni}]_{n \times i} \cdot \prod_{k=1}^{h-1} \hat{r}_{k(k+1)} \cdot [r_{ij}]_{i \times j}$$

( $l, t, h \in N, t$  : total layers of positions,  $h$  : total layers of roles)

### 3. Example for Single-layer Model Algorithm

For a given existing organization authorization system, we analyze its internal relationship between users, positions, roles and operations.

Figure 8 shows the mapping between users and positions.

We get:

$$U_P = [u_{mn}]_{m \times n} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} \\ u_{21} & u_{22} & u_{23} & u_{24} & u_{25} \\ u_{31} & u_{32} & u_{33} & u_{34} & u_{35} \\ u_{41} & u_{42} & u_{43} & u_{44} & u_{45} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix};$$

Figure 9 shows the mapping between positions and roles:

We get:

$$P_R = [p_{ni}]_{n \times i} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \\ p_{41} & p_{42} & p_{43} \\ p_{51} & p_{52} & p_{53} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix};$$

Figure 10 shows the mapping between roles and operations:

We get:

$$R_O = [r_{ij}]_{i \times j} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} & r_{15} \\ r_{21} & r_{22} & r_{23} & r_{24} & r_{25} \\ r_{31} & r_{32} & r_{33} & r_{34} & r_{35} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix};$$

The whole system mapping is shown in figure 11.

From  $U_R = [u_{mn}]_{m \times n} \cdot [p_{ni}]_{n \times i}$  we get:

$$U_R = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix};$$

From  $P_O = [p_{ni}]_{n \times i} \cdot [r_{ij}]_{i \times j}$  we get:

$$P_O = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 1 & 2 & 2 & 1 & 1 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix};$$

From  $T = [u_{mn}]_{m \times n} \cdot [p_{ni}]_{n \times i} \cdot [r_{ij}]_{i \times j}$  we get:

$$T = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 5 & 3 & 1 & 1 \\ 2 & 5 & 5 & 2 & 2 \\ 1 & 3 & 5 & 3 & 3 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix};$$

Now we can get the conclusion from the above calculation results:

From the mapping  $U_R$ , the relationship of users and roles, we can see that there are 3 paths for “user 1” to access “role 1”, 3 paths for “user 2” to access “role 2”, 3 paths for “user 3” to access “role 3”, and only 1 path for “user 4” to access “role 4”. Among these users, user 1, 2, 3 get full the rights to access all the roles, but user 4 does not.

From the mapping  $P_O$ , we can see that “position 1” can only access two operations, but “position 3” can get the right to access all the operations in the system.

From the mapping  $T$ , there are 5 paths for user 1, user 2 and user 3 to get operation 2 and operation 3. Obviously the authorization process was duplicated, and threatened to the safety of the system. If some user has more than 1 path to get an operation, when someday the system administrator changes the configuration of the system and bans one path of the user, the user can still access the operation through other paths. This can cause some uncertainty of the system.

## References

- Adams, S Farrell. (1999). Internet X. 509 Public Key Infrastructure: Certificate Management Protocols[EB/OL] .RFC 2510. March 1999.
- Anders Kristensen. (1999). Formsheets and the XML Forms Language .Computer NetWorks, 1999, 31 :1189 .
- Arpinar I B et al. (1999). Formalization of workflows and correctness issues in the presence of concurrency .Distributed and Parallel Databases, 1999, 7(2):199~248.
- Charlie Kaufman, Radia Perlman, Mike Speciner. (2002). Network Security: Private Communication in a Public World [M] .America: Prentice Hall, 2002.
- Christensen S R, Petrucci L. (2000). Modular analysis of Petri nets .The Computer Journal, 2000, 43(3):224~242 .
- Christopher Steel, Ramesh Nagappan, Ray Lai. (2006). Core Security Patterns Best Practices and Strategies for J2EE, Web Services and Identity Management[M] .2006.
- Dave Crane, Eric Pascarello, Darren James. (2005). Ajax in Action [M]. America: Manning, 2005.
- Eva Maler, Prateek Mishra. (2003). Assertions and Protocol for the OASIS Security Assertion Markup Language(SAML)v1.1[EB/OL] .http://www.oasisopen.org/committee/security/. 2003-09.
- Ferraiolo David F,Sandhu Ravi,Gavrila Serban,et al. (2001). Pro2 posed NIST standard for role2based access control[J] .ACM Transac-tions on Information and Systems Security, 2001, 4 (3) :224-274 .
- Holloway L E, Krogh B H, Giua A. (1999). A survey of Petri net methods for controlled discrete event systems .Discrete Event Dynamic Systems: Theory and Applications, 1999, 7(2):151~190 .
- Jesse James Garrett. (2005). Ajax: A New Approach to Web Applications [EB/OL] .http://www.adaptivepath.com/publications/essays/archives/000385.php. 2005-2.
- Liberty Alliance Project. Liberty Architecture View v1.1[EB/OL] .http://www.Projectliberty.org. 2003-01.
- Liu K C, Ong T N. (1999). A modeling approach for handling business rules and exception .The Computer Journal, 1999, 42(3):206~223 .
- Luo Z W, Sheth A, Kochut K, Miller J. (2000). Exception handling in workflow system .Applied Intelligence, 2000, 13(2):125~147 .
- Mazzeo A, Mazzeocca N, Russo S, Vittorin V. (1997). A systematic approach to the Petri net based specification of

concurrent systems .Real-Time Systems, 1997, 13(3): 219~236 .

OASIS eXtensible Access Control Markup Language(XACML) version 2.0[EB/OL].  
<http://docs.oasis-open.org/xacml/2.0/access-control-xacml-2.0-core-spec-OS.pdf>. 2005—2-01.

Perry D E, Romanovsky A, Tripathi A. (2000). Current trands in exception handling .IEEE Transaction on Software Engineering, 2000, 26(9) :817~819 .

R. Fielding, J. Gettys, J. Mogul, et al. (1999). Hypertext Transfer Protocol-HTTP/1.1 [EB/OL] .RFC2616.  
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>: October, 1999.

Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, et al. (1996). Role-Based Access Control Models .IEEE Computer, 1996, 29(2):38-47 .

Ryan Asleson, (2005). Nathaniel T Schutta. Foundations of Ajax [M] .America: Apress, 2005.

SANDHU R S, FERRAILOLO D, KUHN R. (2000). The NIST model for role-based access control: Towards a unified standard .Proceedings of the Fifth ACM Workshop on Role Based Access Control [C]. Berlin, Germany:ACM, 2000, :47-63 .

Volchkov A. (2001). Revisiting Single Sign-On A Pragmatic Approach in a New Context .IT Professional, IEEE, 2001, :39-45 .

W M P van der Aalst. (2001). A reference model for team-enabled workflow management systems .Data and Knowledge Engineering, 2001, 38(3):335~263 .

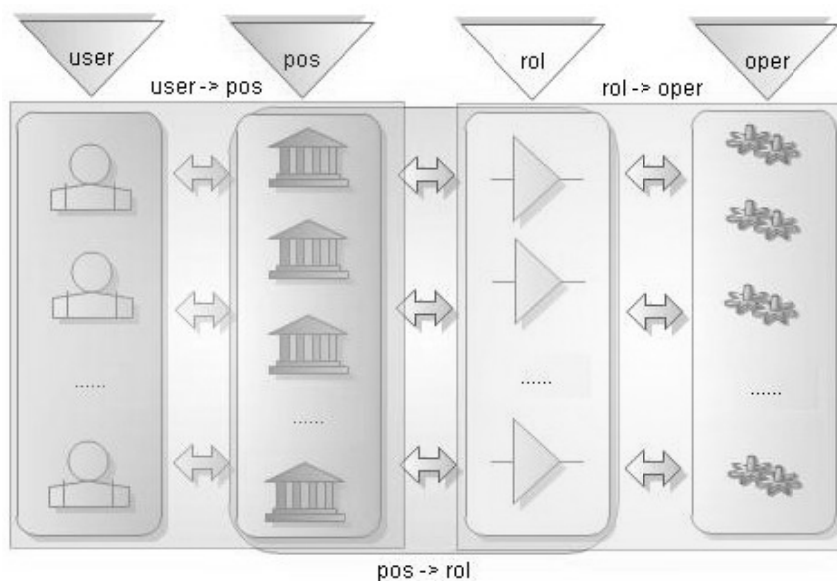


Figure 1. User-Position-Role-Operation Model

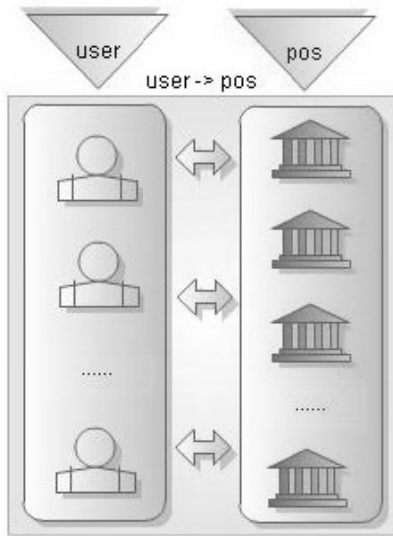


Figure 2. User-Position Mapping

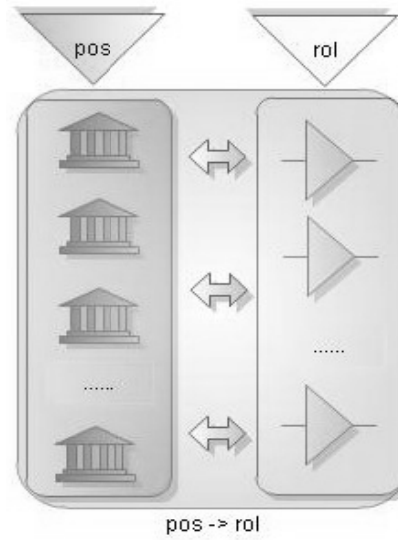


Figure 3. Position-Role Mapping

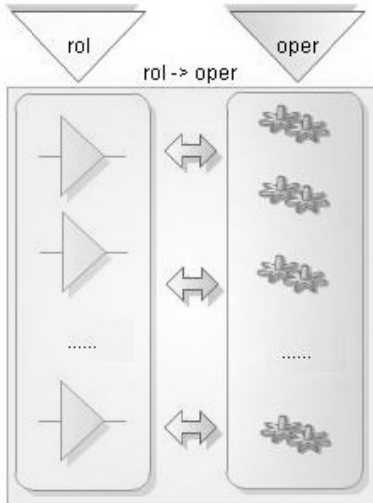


Figure 4. Role-Operation Mapping

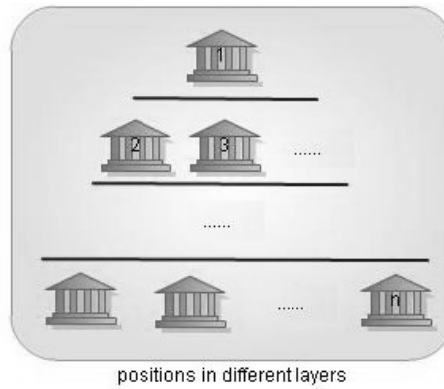


Figure 5. Positions Layers

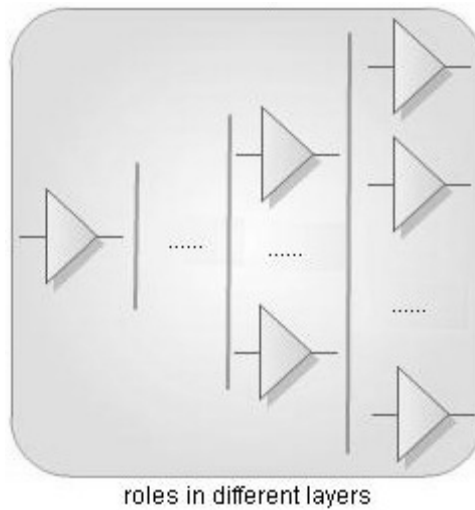


Figure 6. Roles Layers



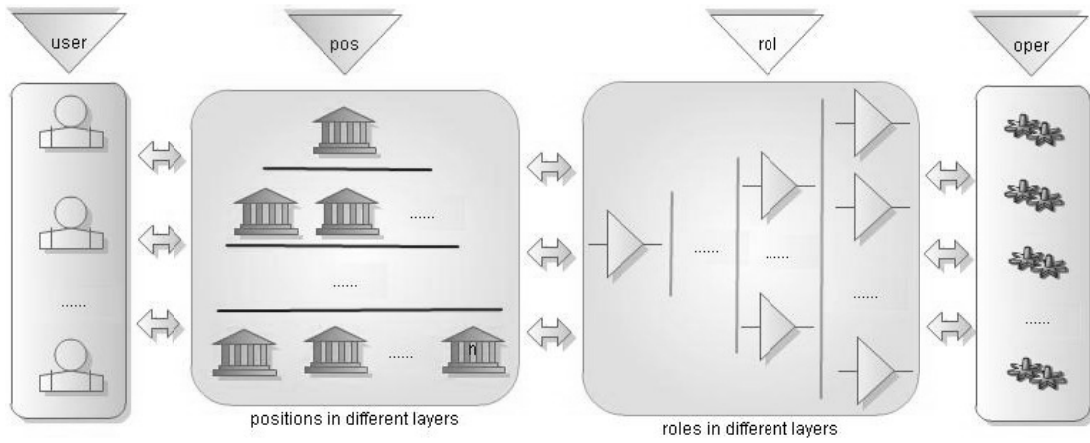


Figure 7. Complex Organization Authorization System Model Based on Positions and Roles

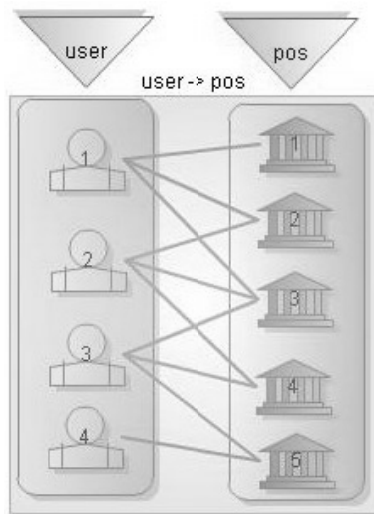


Figure 8. User-Position Mapping

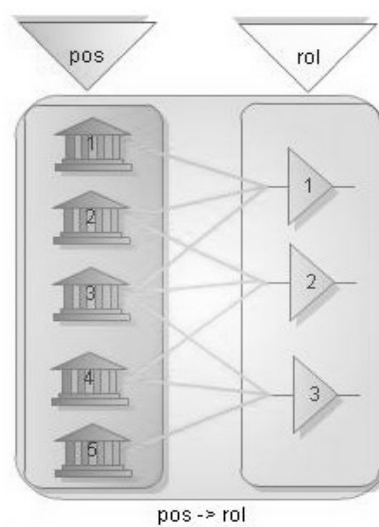


Figure 9. Position-Role Mapping

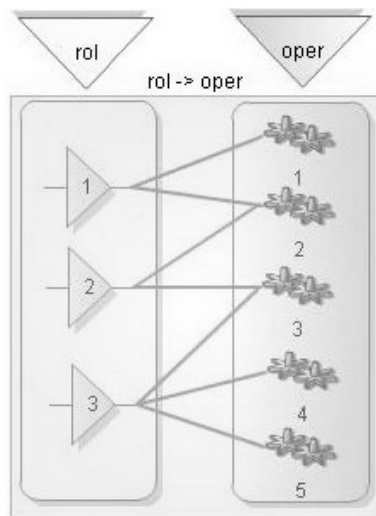


Figure 10. Role-Operation Mapping

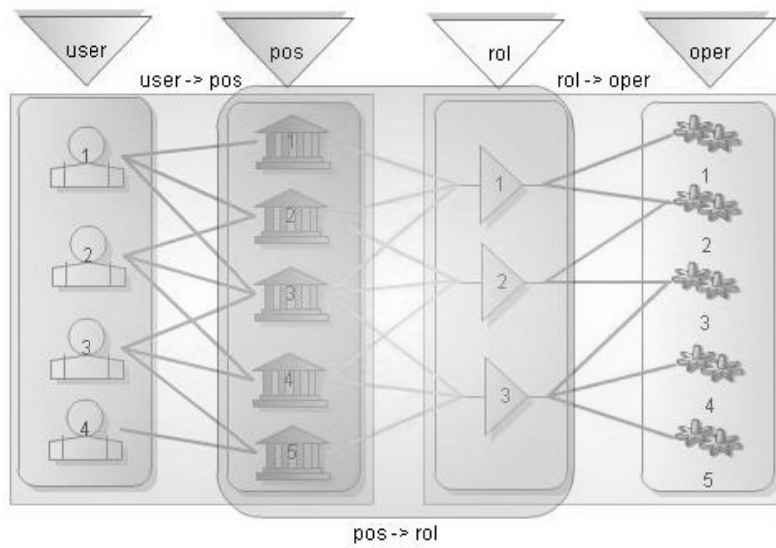


Figure 11. Relationship in Organization Authorization Process