Enterprise Model as a Basis of Administration on Role-Based Access Control

Sejong Oh , Seog Park
Dept. of Computer Science, Sogang University,
121-742, Seoul, Korea
{sejong,spark}@dblab.sogang.ac.kr

Abstract

Access control is one of important security issues for large enterprise organizations. Role-based access control (RBAC) model is well known and recognized as a good security model for enterprise environment. Though RBAC is a good model, administration of RBAC including building and maintaining access control information remains a difficult problem in large companies. RBAC model itself does not tell the solution. Little research was done on practical ways to find the information that fills RBAC components such as role, role hierarchy, permission-role assignment, user-role assignment, and so on from the real world.

In this paper we suggest model-based administration of RBAC in an enterprise environment. Model-based administration methods allows security administrator to manage access control by GUI that supports graphical enterprise model. If security administrator creates or changes some of components of graphical enterprise model, then it is translated to RBAC schema information by administration tool. We focus on a practical way of deriving access control information from real world. It is a core of model-based administration. Here we show the derivation method and implementation experiences

1. Introduction

Since many companies have recognized computer as an essential medium to increase competitive power, they have competitively built their computer systems. Hence growth of companies, volumes of information, and related personnel have increased, as a result security problems have become increasingly difficult. Access control is an important security issue in the enterprise environment. Access means the ability to perform work such as reading, writing, and the execution of the system resources. Access control is the way to control ability to perform the work [1]. The huge number information objects and users in a large company make expression of the access right relationship between the users and information objects a difficult issue.

Role-based access control model (RBAC) [5][6] is known to be proper access control model for enterprise environment. The central notion of RBAC is to prevent users from accessing company information by discretion. Instead, access rights are associated with roles in which users are assigned to appropriate roles. The notion of role is an enterprise or organizational concept. As such, RBAC allows us to model security from an enterprise perspective since we can align security modeling to the roles and responsibilities in the company.

Even though RBAC research was developed increasingly, little research was done on the practical way towards administration access control information from the enterprise world [4]. RBAC administration includes building and updating RBAC information. Finding the role, constructing role hierarchy (RH), user-role assignment (URA), and permission-role assignment (PRA) are responsibilities of developers or security administrators. A large enterprise-wide system has number of roles, users, and information objects. Therefore, managing these roles and users, and their interrelationships is a formidable task for security administrators. Administrative RBAC (ARBAC) [7] is an alternative solution. Though ARBAC relaxes the complexity of administration, it cannot solve the fundamental problems of security administration.

In this paper, we have suggested the framework for model-based RBAC administration method. In the general RBAC administration, security administrators deal with raw access control information such as role, RH, URA, and PRA. In the model-based RBAC administration, security administrators deal with graphical enterprise model, and the change of enterprise model reflects in a semi-automatic way in the raw access control information. Therefore security administrators can manage access control information easily. The security administration is very intuitive and

0-7695-1128-7/01 $10.00 © 2001 IEEE

real world friendly in our method.

In the previous paper, we suggested task-role-based access control (T-RBAC) model [2][3]. It is an improved RBAC model that solves the problems of general RBAC model such as role hierarchy [10]. In T-RBAC model, permissions are assigned to tasks, and tasks are assigned to roles. Task is the unit of job function or business activity. In this paper we suppose T-RBAC as the access control model rather than general RBAC model.

The rest of this paper is organized as follows. In section 2 we show our motivation and basic idea of model-based RBAC administration method. Section 3 has a brief description of T-RBAC model. Section 4 describes process of deriving access control (T-RBAC) information from enterprise model. Discussion about derivation process are in section 5. In section 6 we introduces implementation of model-based RBAC administration method. Section 7 presents conclusion and proposes further work.

2. Motivation

As we said before, building & managing RBAC schema information are important issues in an enterprise environment. Let's see a simple management problem as an example. We suppose a situation as follows:

There is a role 'sales_manager'. Now security administrator should revoke all 'write' privileges that are belong to task of processing sales order.

In general RBAC model, security administrator follows the following two steps to do the work. First, he/she should know which information objects belong to the task. Second, he/she updates PRA (permission-role assignment) information to revoke target 'write' privileges from the role 'sales_manager'. In the general RBAC model, permissions are directly assigned the roles as shown in Fig.1. It is difficult for security administrator to know which information objects belong to specific tasks. He/she needs background knowledge about tasks and related information objects. But it is a very heavy work in a large company. If an object belongs to many tasks, changing privilege of the object may bring about an undesirable result. Second step also has problems. Security administrator deal with raw data in PRA table to update privilege, and it is not convenient. PRA table shows simple information. System cannot show any extra information – for example, the effect of updating – about the behavior of updating PRA table. Building RBAC schema information has similar difficulties in the above said management problems.

Role_name	Permission
sales_manager	File1[r,w]
sales_manager	File2[r,w]
sales_manager	File3[r]
sales_manager	File4[r,w]

Fig 1. An example of PRA table

Fundamental problem of RBAC administration including building and managing RBAC schema information is **lack of interrelation information between change of real world and change of RBAC schema information**. Security administrator needs efficient user interface that support real world style access control. This is our motivation. To solve the problem, we adopt improved RBAC model, task-role-based access control (T-RBAC) model. Permissions are assigned to related tasks, and tasks are assigned to related roles in T-RBAC. Therefore PRA table in RBAC is separated to PTA (permission task assignment) and TRA (task role assignment) tables in T-RBAC. Fig. 2 shows PTA and TRA tables as an equivalent PRA table in Fig. 1. Users may easily know which information objects are related to the specific task in T-RBAC model.

Role_name	Task
sales_manager	sales_order
sales_manager	sales_account

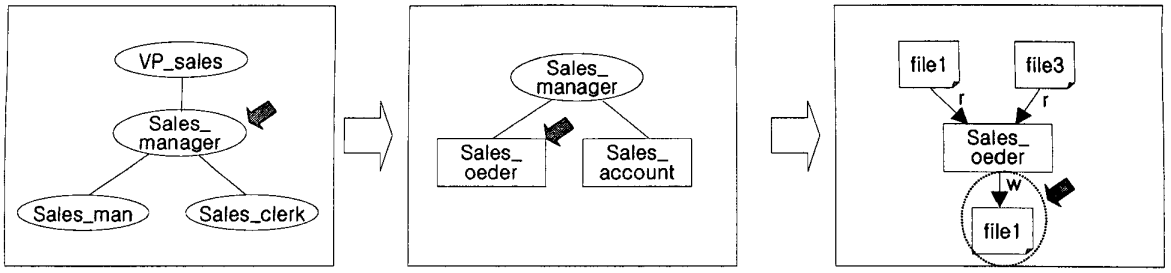
(a) TRA table

Task	Permission
sales_order	File1[r,w]
sales_order	File3[r]
sales_account	File2[r,w]
sales_account	File4[r,w]

(b) PTA table

Fig.2 TRA & PTA table in T-RBAC

Fig.3 shows a model scenario of permission update in the model-based security administration. Security administrator uses GUI instead of dealing with PTA table directly. The GUI supports some of business model such as organization diagram and task diagram.



(a) Click "sales_manager" role in the organization diagram

(b) Click "sales_order" task in the role-task diagram

(c) Delete file1[w] in the task diagram

Task	Permission
sales_order	file1[r]
sales_order	file3[r]
sales_account	file2[r,w]
sales_account	file4[r,w]

(d) T-RBAC system updates PTA table automatically.

Fig.3 Permission update example in the model-based security administration

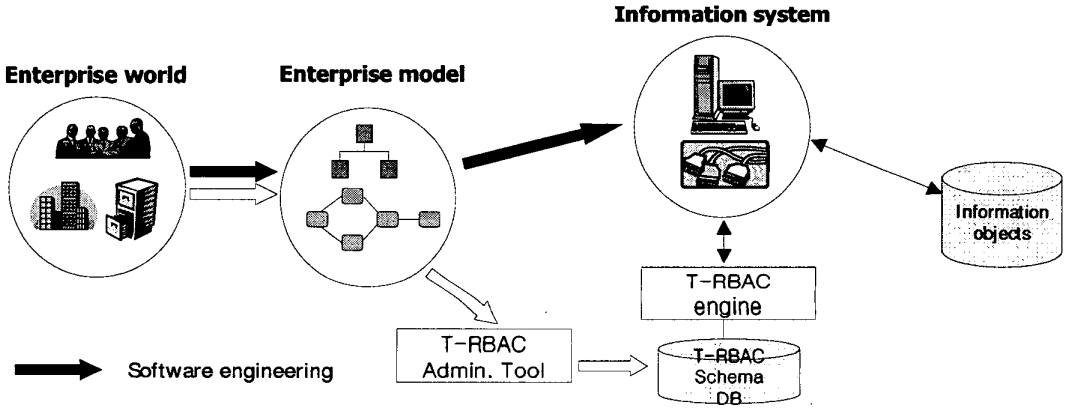


Fig.4 Basic concept of model-based RBAC administration

Security administrator clicks or changes related components of the diagrams. Then the result automatically reflects in corresponding records at the PTA table. If we can implement such an administration GUI, security administration may become an easy task. It is the final goal for us. In many business software projects such as ERP, it builds a conceptual enterprise model before programming information system; and the enterprise model contains access control information. Therefore we can build and manage RBAC schema information by using some enterprise model diagrams. Fig.4 shows the basic concept of model-based RBAC administration. The core of model-based RBAC administration is a derivation process. It describes how to derive RBAC schema information from an enterprise model. We predominantly describe the derivation process.

3. A Brief Description of T-RBAC Model

Before we describe the derivation process of RBAC schema information, we shall introduce T-RBAC model. T-RBAC is an integrated model of role-based access control and activity-based access control models based on task classification. There are four classes that have different access control characteristics in the companies. If a user U_i has tasks that belong to class-S, their related access rights are inherited to user Un who has a higher job position than U_i in the organization structure. Tasks that belong to class-W, which is related

with workflow and show the characteristics of an ABAC model. Tasks that belong to class-P are private ones. class-W and class-P do not have inheritance characteristics. class-A gathers activity approval tasks and it has characteristics of both inheritance and ABAC.

Fig. 5 shows a brief of T-RBAC. The major difference between T-RBAC and RBAC is that the access rights are assigned to task in T-RBAC, but access rights are assigned to role in RBAC. In the real world access rights are needed for the user to perform tasks. So assigning access rights to task is reasonable. Another difference is the role hierarchy. We use supervision role hierarchy (S-RH) instead of general role hierarchy. In the S-RH, higher role does not inherit all access rights of the lower role in the role hierarchy. Only access rights of class-S or class-A are inherited from lower role to the higher role. Tasks in the class-W or class-A are used to compose workflow. Workflow creates the workflow instances that are set of task instances. Access rights are assigned to tasks in the class-W statically. But the access rights are bound and activated during the execution of task instance.

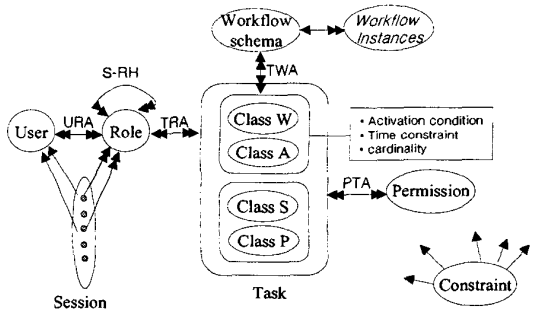


Fig. 5 T-RBAC model

4. Derivation Process of T-RBAC Schema Information

4.1. Basic Philosophy and Concept

The fact that access control model has a profound relationship with real world encourages us to research derivation process of access control information from enterprise environment. We think that enterprise environment implies T-RBAC aspects, and there exists a methodology to derive the T-RBAC schema information. Our Derivation process is based on the following observation:

- In general, users in the company belong to the organization structure.
- Users perform their assigned tasks (job functions) according to their job positions. Task is a meaningful unit of business work. From the point of information system, it is a set of both information object and operation.
- Access rights are required only for executing assigned tasks. If we can know a user's assigned tasks, we can decide which access rights are assigned to the user.
- Business role or job position can be defined as a set of tasks.
- Least privilege includes of 'Need-to-know' and 'Need-to-do' is a basic principle of access control in the enterprise environment.

Paper [4] shows that there are 5 classes of roles in the business world. Based on paper [4], we define 3 types of roles such as

- *Organizational role* (ex. Sales_Dept, Finance_ Dept): this is a basic role for all users who belong to the organization. So the permissions are inherited to users who belong to the organization.
- *Job position role* (ex. Vice_President_Sales, Finance_manager)
- *Functional role* (ex. Developer, Programmer)

Our basic strategy is shown in Fig.6. We assumed that enterprise model is built in the process of software engineering. From the enterprise model, we abstract related components of T-RBAC and reform them to our predefined business model diagrams (reduced enterprise model). And then we derive T-RBAC information from reduced enterprise model in a semi-automatic way. So reduced enterprise model is a realistic material of derivation process. This reduced enterprise model includes 4 diagrams in Fig.8 ~ Fig.11. These diagrams are minimal set of enterprise model components that imply T-RBAC aspects. Fig.7 shows notations in the 4 diagrams.

Fig.8 is an organization diagram. Tetragon box expresses division name and parallelogram box expresses job position name that belongs to the division. Organization diagram shows division hierarchy of organization and job position hierarchy. Fig.9 is an information object diagram. For simplicity we assume that information object is a table in the database. Fig.9 can be translated from E-R diagram of the Enterprise model. Fig.10 is a task diagram. It shows a work unit and its input/output data (tables in Fig. 9). Also it shows the subjects (executors) that execute the task. Fig.11 is a Business process diagram. It shows the

workflow and tasks in relation with 'separation of duty'. D_SOD means dynamic separation of duty and S_SOD means static separation of duty.

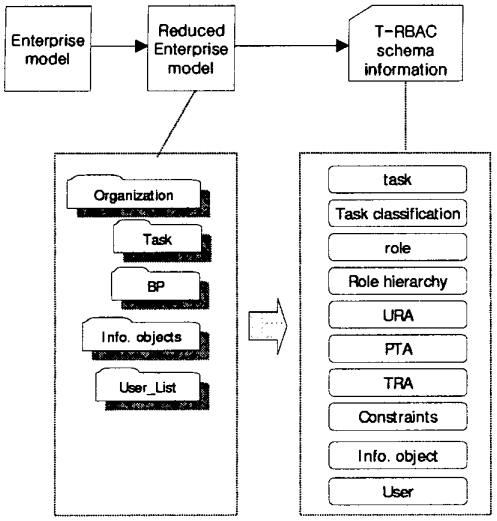


Fig.6 Basic strategy of derivation

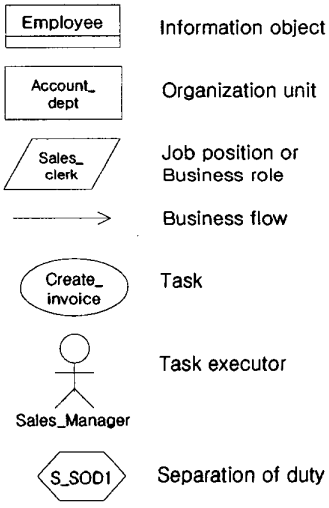


Fig.7 Notations of reduced enterprise model

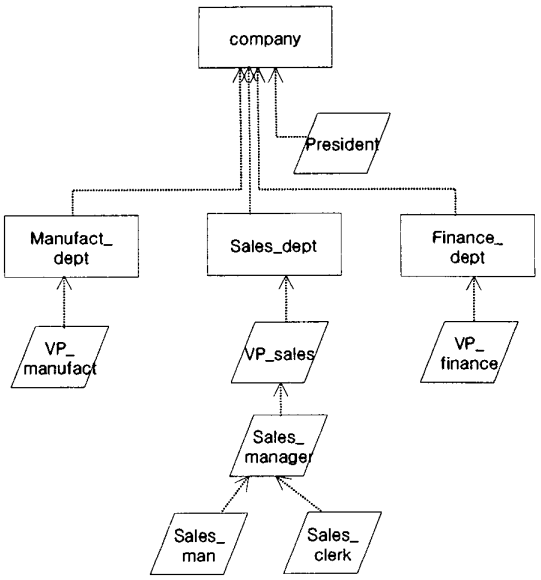


Fig. 8 Organization Diagram

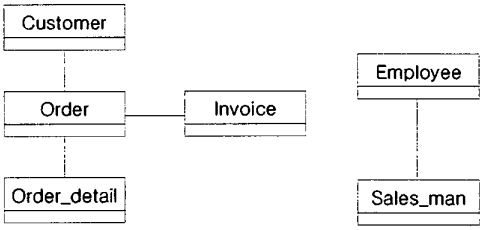


Fig. 9 Information Object Diagram

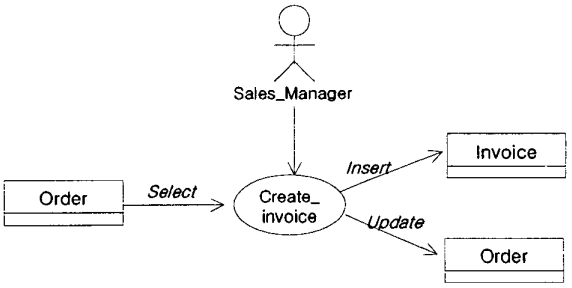


Fig. 10 Task diagram

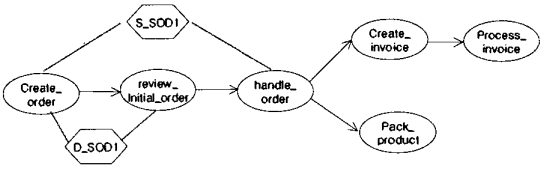


Fig. 11 Business Process Diagram

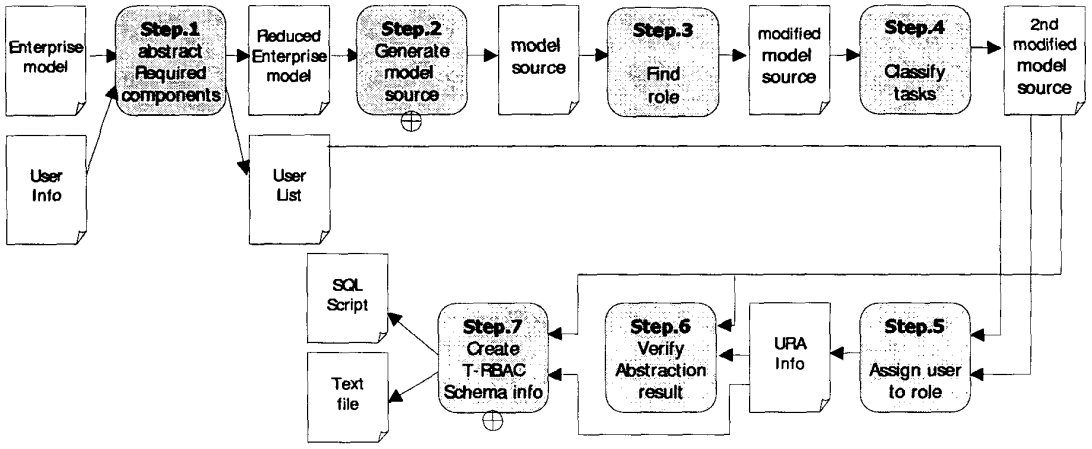


Fig. 12 Derivation process of T-RBAC schema information

4.2. The Steps of Derivation Process

The complete steps of derivation process are expressed in Fig.12. Our basic strategy is to build a reduced enterprise model from a pre-constructed large business model (Step.1). And to derive access control information from reduced enterprise model (Step.2~Step.7). Initial input of the process is pre-constructed enterprise model and user information. Final output of the process is SQL script that makes T-RBAC data into database or text files that contains T-RBAC schema information. The sign '⊕' means that the pointed step can be processed automatically apart from human intervention.

Now we describe each step of derivation process of T-RBAC aspects. Heading of each step shows its input, output, and related reduced enterprise diagram. 'Derived T-RBAC components' means the T-RBAC components that can be derived after the process.

Step.1 Abstracts Required components.

Input	: enterprise model User Information
Output	: Reduced enterprise model User list
Derived T-RBAC Components	: User

Main work of step.1 is to abstract T-RBAC related components from enterprise model to reduced enterprise model. Enterprise model can be constructed by various tools or methodology. If input enterprise model were constructed following reduced enterprise model diagrams, step.1 would be an easy work. Further if the enterprise model includes diagrams from Fig.8~Fig.11, step.1 can be processed automatically. In otherwise case, human experts – in general the developers of the enterprise model – should do step.1 manually. Generally user information is not included in the business model. Therefore user information is additionally needed. User list includes 'user id', 'organization name', and 'job position'.

Step.2 Generate model source

Input	: UML model
Output	: Model source
Derived T-RBAC Components	: Information objects, Task, Workflow, PTA

Most of the enterprise modeling tools supports the function of translating graphical model to text source file. So by tool, step.2 can be processed automatically by tool. Below box shows a part of the example source file. It is translated from information object diagram (Fig.4). From the model source file we can derive T-RBAC information such as information object, task, workflow, permission-task assignment (PTA).

```
logical_models      (list unit_reference_list
(object Class "Customer"
        quid        "39C72121010E"
        language    "Java")
(object Class "Order"
        quid        "39C7221201C2"
        language    "Java")
(object Class "P_Invoice"
        quid        "39C72FD100C8"
        language    "Java")
(object Class "Order_detail"
        quid        "39C7308600B4"
        language    "Java")
(object Class "Employee"
```

```
        quid        "39C730CD0352"
        language    "Java")
```

Step.3 Find role

Input : Model source
Output : Modified model source(1)
Derived T-RBAC Components :
Role, RH, TRA

Step.3 use information of the task diagram and organization diagram. There are 5 sub-steps in step.3.

3.1 Create roles from task diagram.
Task diagram contains executor component. We can choose executor names as role names. Four types of roles are created according to the characteristics of executors.

3.2 Assign tasks to roles (TRA) from task diagram.
In the task diagram executor (role) is related to task. An executor can be related to many tasks and A task can be related to many executors.

3.3 Create initial role hierarchy from the organization diagram.
Organization diagram implies 'organizational role' and 'job position role'. So initial role hierarchy contains two types of roles. For example, the organization structure in Fig.8 translates to role hierarchy in Fig.13.

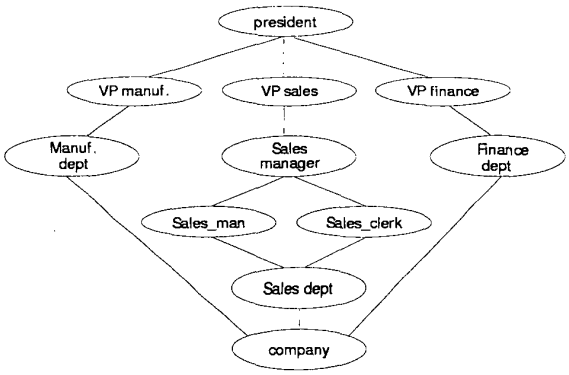


Fig.13 Initial role hierarchy example

3.4 Unify roles that contain the same tasks
If a role A and role B have same tasks, then we can assume that role A and role B are same role. These roles should be unified.

3.5 Adding some roles (functional role, basic roles) to role hierarchy
Initial role hierarchy doesn't contain functional role and basic roles. Because they are not expressed in the organization diagram and there is no information to add them in the role hierarchy. So a human expert should process step3.5.

Step.4 Classify tasks

Input : Model source(1)
Output : Model source(2)
Derived T-RBAC Components :
classified tasks

In T-RBAC model, tasks are classified into 3 classes. And different access control is applied to tasks according to their class. There are three classified rules.

Rule 1. Tasks in business process diagram and assigned to functional roles are belong to class-W

Rule 2. All tasks executed by organization roles are belong to class-S (because they have a inheritance characteristic)

Rule 3. Tasks belong to business process diagram and assigned to job position roles or organizational roles are class-A

Rule 4. Others are belong to class-P

It is possible that some tasks that are classified to class P that should be classified to class-S. Therefore if necessary; a human expert executes modification.

Step.5 Assign user to role

Input : User list
Modified moel source(2)
Output : URA information
Derived T-RBAC Components : URA

According to the job position information in the user list, job position roles can be assigned to appropriate users. Organizational roles are assigned to users who belong to the organization. A human expert assigns functional roles.

Step.6 Verify abstraction

Input : User list
Modified moel source(2)
Output : Verified model
Derived T-RBAC Components : -

After the prior steps are completed, verification is required for the result. Tools can support some part of verification.

Step.7 Create T-RBAC SQL script or T-RBAC schema files

Input : User list
Modified moel source(2)
Output : T-RBAC SQL script
T-RBAC schema files
Derived T-RBAC Components : -

Final step is to create a SQL script for generating T-RBAC information in the database. T-RBAC schema files that have same information can be created in exchange of SQL script. Sample script file and schema files are as follows

```
CREATE USER 'S001' IDENTIFYED BY 'qwee' ;
CREATE USER 'S002' IDENTIFYED BY 'aaas1' ;
...
CREATE ROLE 'SALES_CLERK' ;
...
GRANT SALES_CLERK ON VP_SALES TO S001 ;
...
INSERT INTO tra_tbl (role_name, task_name)
        VALUES
        ('SALES_CLERK', 'ISSUE_INVOICE');
```

File name: URA.txt

User_id	assigned_role
S001	SALES_CLERK
S002	SALES_MANAGER

5. Discussion

We suggest a practical solution to derive T-RBAC information from enterprise environment. It has some discussion points.

Good enterprise model produces good T-RBAC information. Enterprise model is a realistic input source of our methodology. And derivation process is executed semi-automatically. So the quality of T-RBAC information depends on the quality of enterprise model.

Good description of task is the most important thing in our derivation process. As you can see task is the central concept of access control design. Fig.14 shows it. In general, Data flow diagram in the enterprise model implies task information. Main issue is that task can be described at various levels. For example, two tasks 'oeder_update' and 'order_cancel' can be described as a single task 'order_manage'. So determining appropriate task level is important. In depth research is needed in this issue.

Total automation of derivation process is very difficult. Ideal goal of our derivation process is total automation. But it requires enormous material information. Automation degree and the cost of maintenance needed material information are in trade-off.

Supporting tool is necessary. Our derivation process deals with many related components. So manual work is difficult and requires long duration. Supporting tools is an essential component of our methodology. In the next section, we show the implementation result of the supporting tool.

Reduced enterprise model can be used for managing (updating) T-RBAC schema information following change in the real world. Reduced enterprise model is necessary to build T-RBAC information, the first time. After the building process, reduced enterprise model would not be useless. Maintenance of T-RBAC information is needed following the change in the real world. In this case, security administrator use reduced enterprise model. If he/she modifies the reduced enterprise model following the change, then T-RBAC can be update automatically. Management module will be added to our supporting tool in the next section.

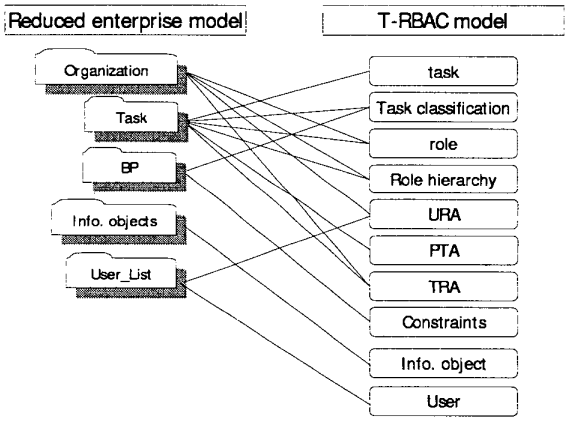


Fig.14 Relationship reduced enterprise diagram & T-RBAC components.

6. Implementation

We have a two-step implementation plan. First step is implementing the building process of T-RBAC schema information. Second step is the implementation of maintenance module including enterprise-modeling function. Now the first step was processed, and we have implement prototype system. Fig.15 shows work of first step.

We use an ARIS [8][9] tool for designing reduced enterprise model. ARIS tool produces text source files from inserted enterprise model diagrams. Visual C++ language is used to program translation from text

source files to temporal database. Deriving tool, which is created by Power Builder, produces final T-RBAC schema information from temporal database.

Fig.16 shows a future of second step. In this program, if a security manager modifies business model (task diagram), then related information of access control is automatically updated. We are researching about step 2.

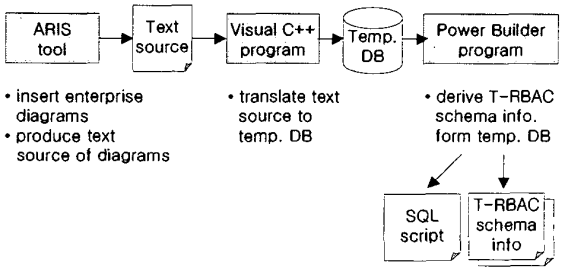


Fig.15 Implementation environment of the first step

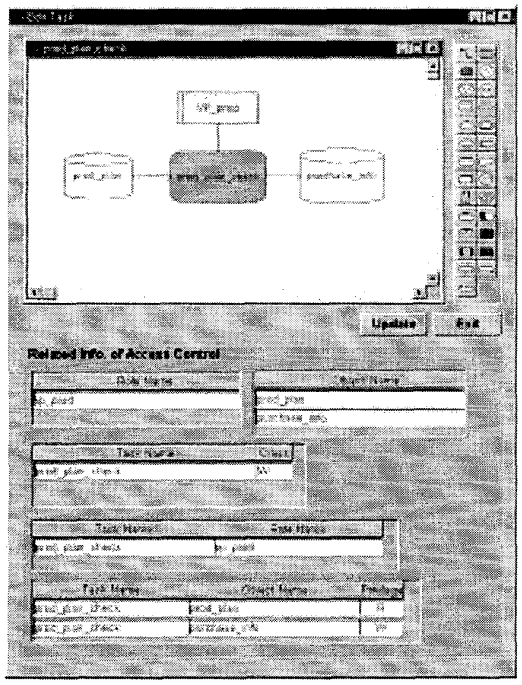


Fig.16 A future of maintenance module

7. Conclusion & Further Work

Access control mechanism in the enterprise environment has deep relationship with the real world. Task is the central concept of design access control. Task is defined as a unit of meaningful business work. It determines the contents of permission and role. Role is a set of tasks. T-RBAC model is based on the concept task and role.

In this paper, we have suggested model-based administration of RBAC – T-RBAC accurately. Deriving access control information from enterprise environment is core of model-based administration. It supports semi-automatic derivation mechanism. Our basic strategy is to build a reduced enterprise model from pre-constructed large business model. And to derive access control information from reduced enterprise model in a semi-automatic way.

Developing model-based administration tool is our final goal. We implemented derivation module, and management module will be added. In the long run, software engineering and access control engineering should be combined as pointed in the paper [4].

References

[1] C.P.Pfleeger, *Security in Computing*, second edition, Prentice-Hall International Inc.,1997.

[2] S.Oh and S.Park, "Task-Role Based Access Control (T-RBAC): An Improved Access Control Method for Enterprise Environment", Lecture Note in Computer Science 1873, Database and Expert Systems Applications, Proc. Of 11th International Conference, DEXA 2000. Sep. 2000.

[3] S.Oh and S.Park, "An Integration Model of Role-Based Access Control and Activity-Based Access Control Using Task", Proc. of 14th Annual IFIP WG 11.3 Working Conference on Database Security, Aug. 2000.

[4] H.Roeckle, G.Schimpf, and R.Weidinger, "Process-Oriented Approach for Role-Finding to Implement Role-Based Security Administration in a Large Industrial Organization", Proc. 5th ACM Workshop on Role-Based Access Control, 2000.

[5] R.S.Sandhu, E.J.Coyne, H.L.Feinstein, and C.E.Youman, "Role-Based Access Control Method", IEEE Computer, vol.29, Feb. 1996.

[6] Ferraio, J.Cugini, and R.Kuhn, "Role-based Access Control (RBAC): Features and motivations", Proc. of 11th Annual Computer Security Application Conference, 1995.12.

[7] R.S.Sandhu, V.Bhamidipati, E.Coyne, S.Ganta, and C.Youman, "The ARBAC97 Model for Role-Based Administration of Roles: Preliminary Description and Outline", Proc. of second ACM Workshop on Role-Based Access Control, 1997.

[8] IDS Scheer, *ARIS Easy Design Guide*, http://www.ids-scheer.com

[9] IDS Scheer,, *ARIS Modeling Concept*, http://www.ids-scheer.com

[10] R.S.Sandhu, "Role Activation Hierarchies", Proc. of 3rd ACM Workshop on Role-Based Access Control, 1998.