

An Improved Administration Method on Role-Based Access Control in the Enterprise Environment

SEJONG OH AND SEOG PARK*

Department of Computer Science

Sogang University

Seoul 121-742, Korea

Email: {sejong,spark}@dmlab.sogang.ac.kr

Access control is a difficult security issue for 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 the possibility of 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

Keywords: RBAC, access control, security, enterprise environment, business model

1. INTRODUCTION

Since many companies have recognized the computer as an essential tool to increase competitiveness, they have competitively built their computer systems. Hence growth of companies, volumes of information, and related personnel have increased, and 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 the ability to perform the work [1]. The huge number of information objects and users in a large company make the right of access relationship between the users and information objects a difficult issue.

Role-based access control model (RBAC) [5, 6] is known to be a 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 associ-

Received January 30, 2001; accepted July 10, 2001.
Communicated by Chi Sung Laih.

ated 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 with the roles and responsibilities in the company.

Even though RBAC research was increasingly developed, little research was done on the practical way towards administration access control information from the enterprise world. 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 a 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 administration.

In this paper, we have suggested the framework for the 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 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, 11]. 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 is in section 5. In section 6 we introduce implementation of model-based RBAC administration method. Section 7 presents conclusion and proposes further work. (*Note. For simplicity, we use some abbreviations such as: URA/ user-role assignment, TRA/ task-role assignment, PTA/ permission-task assignment, (S)-RH/ (supervision)-role hierarchy*).

2. MOTIVATION

As we said before, building and 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 the security administrator should revoke all 'write' privileges that belong to the task of processing sales orders.

In the general RBAC model, the security administrator follows the following two steps. 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 to the roles as shown in Fig. 1. It is difficult for the 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 difficult task in a large company. If an object belongs to many tasks, changing privilege of the object may bring about an undesirable result. The second step also has problems. The security administrator deal with raw data in PRA table to update privilege, and it is not convenient. PRA table shows simple information. The system cannot show any extra information – for example, the effects 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.

The fundamental problem of RBAC administration, including building and managing RBAC schema information, is **lack of information of interrelation between change of real world and change of RBAC schema information**. Security administrator needs an efficient user interface that supports real world style access control. This is our motivation. To solve the problem, we adopt an 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.

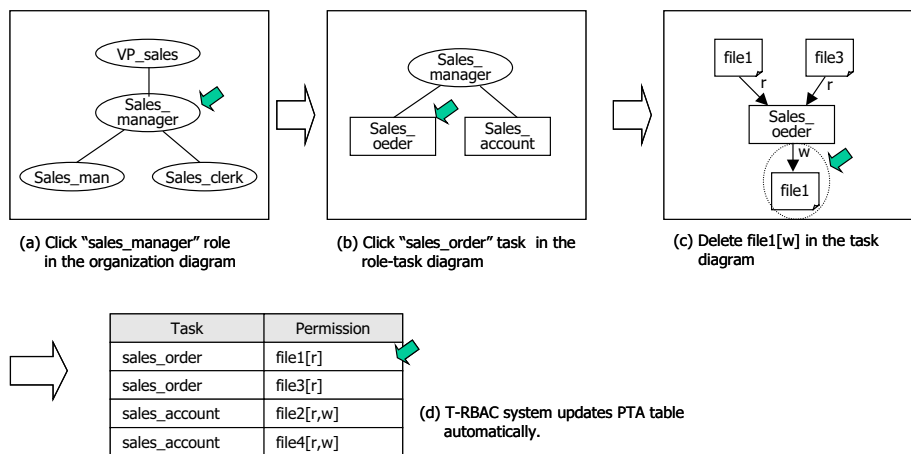


Fig. 3. Permission update example in the model-based security administration.

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. Security administrator clicks or changes related components of the diagrams. Then the result automatically reflects in corresponding records at the TPA 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.

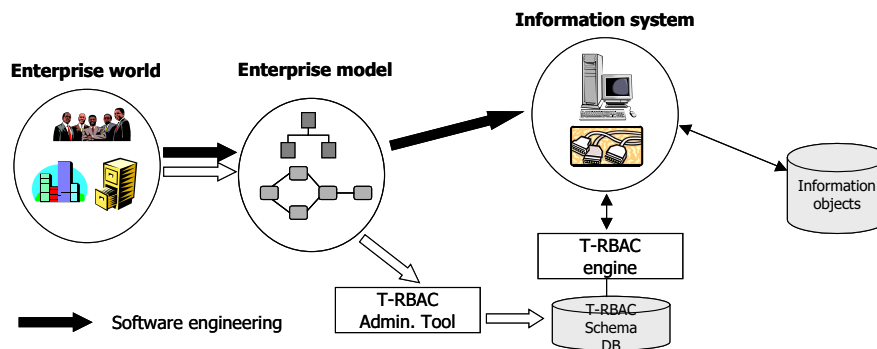


Fig. 4. Basic concept of model-based RBAC administration.

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 4 classes that have different access control characteristics in the companies. If a user U_1 has tasks that belong to class S, their related access rights are inherited to user U_n who has a higher job position than U_1 in the organization structure. Tasks that belong to class W, which is related with workflow and show the characteristics of an active access control model. Tasks that belong to class P are private ones, they do not have inheritance characteristic and related with workflow. Class A has characteristics of class S and class W. Class W and class P do not have inheritance characteristics.

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 and class A are inherited from lower role to the higher role. Tasks in the class W and 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 and class A statically. But the access rights are bound and activated during the execution of task instance.

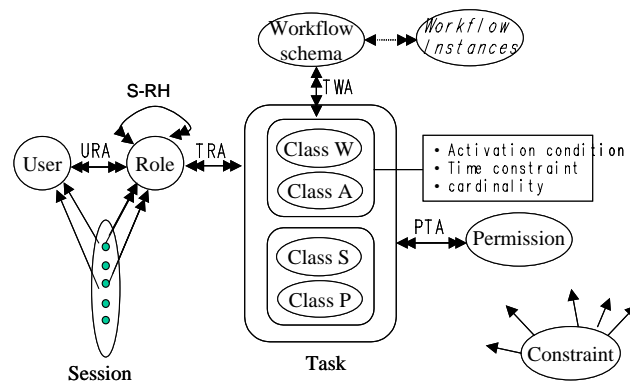


Fig. 5. T-RBAC model.

4. DERIVATION PROCESS OF T-RBAC SCHEMA INFORMATION

4.1 Basic Philosophy and Assumption

The fact that access control model has a profound relationship with real world encourages us to research derivation process of access control information form 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 assumptions:

- There exist domain experts who understand the business model and know the T-RBAC concepts. They create a reduced enterprise model from the original enterprise model according to the proposed notation.
- There exists user information in the target organization. (In the proposed process, we use existing user information rather than creating new user information).
- All the concepts of derivation are based on Chapter 2 and Section 4.1.
- We consider three types of roles:
 - Organizational role (ex. Sales_Dept, Finance_Dept): This is a basic role for all users who belong to the organization. Therefore, the permissions are inherited to users who belong to the organization.
 - Job position role (ex. Vice_President_Sales, Finance_Manager)
 - Business role (ex. Developer, Programmer, Employer, Customer)

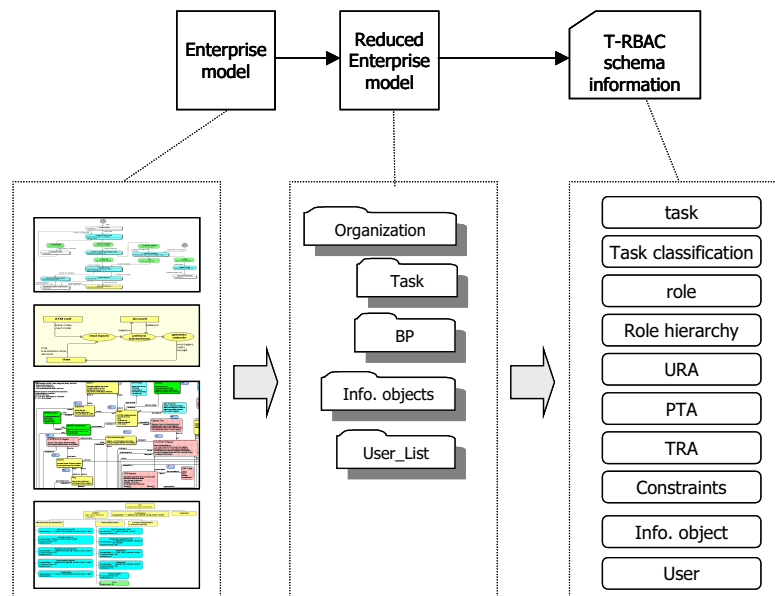


Fig. 6. Basic strategy of derivation.

Our basic strategy is shown in Fig. 6. We assume that the enterprise model is built within the software engineering process. From the enterprise model, we can derive related components of T-RBAC and reform them to our predefined business model diagrams (reduced enterprise model). Then we derive T-RBAC information from the reduced enterprise model in a semi-automatic fashion, so the reduced enterprise model

provides realistic material for the derivation process. To find reasonable reduced enterprise model diagrams, we analyzed well-known business modeling tools and their supported diagrams. Then we chose model components that contain T-RBAC access control information as shown in Fig. 7.

The reduced enterprise model includes four diagrams, as shown in Figs. 8-11. These diagrams are a minimal set of enterprise model diagrams that imply T-RBAC aspects.

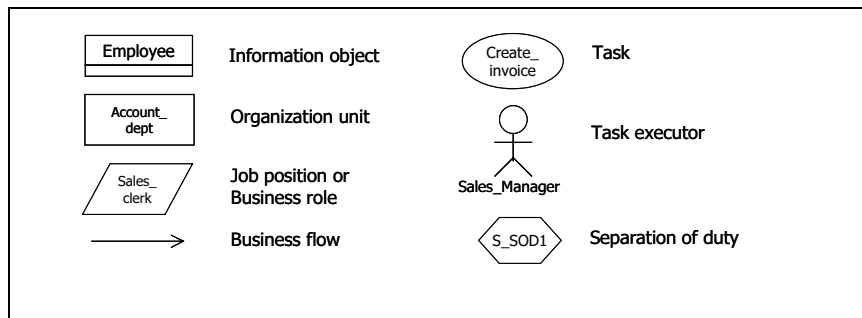


Fig. 7. Notations of reduced enterprise model.

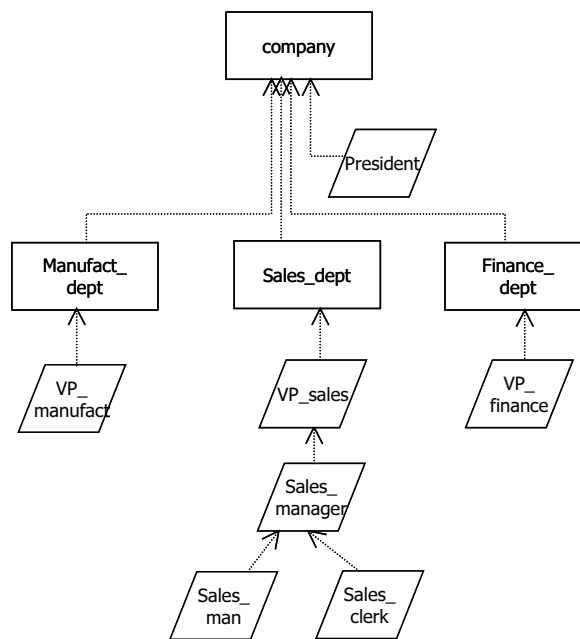


Fig. 8. Organization diagram.

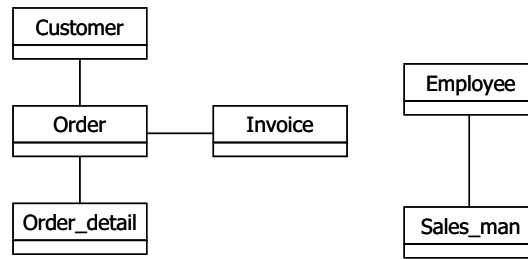


Fig. 9. Information object diagram.

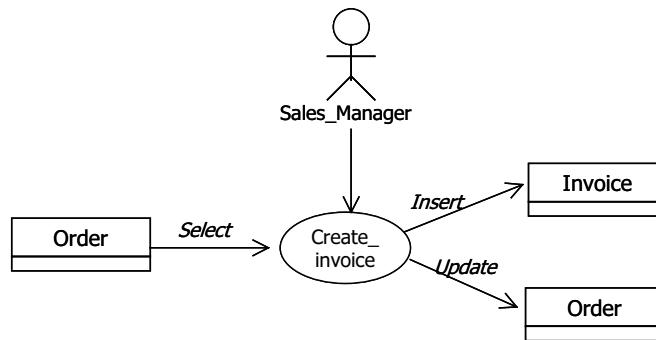


Fig. 10. Task diagram.

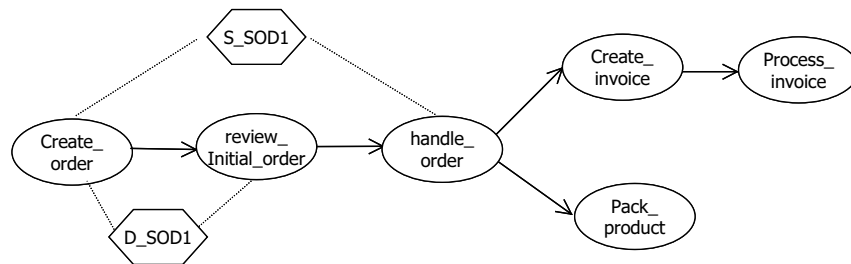


Fig. 11. Business process diagram.

Fig. 8 is an organization diagram where rectangles are divisions, and parallelograms are job positions that belong to the division. The organization diagram shows the division hierarchy of organization, and job position hierarchies. Fig. 9 is an information object diagram. For simplicity we assume that an information object is a file or table in the database. Fig. 9 can be translated from the E-R diagram of the Enterprise model. Fig. 10 is a task diagram that shows a work unit and its input/output data. It also shows the subjects (executors) that execute the task. Fig. 11 is a business process diagram that shows the workflow and tasks in relation to 'separation of duty'. Here D_SOD stands for dynamic separation of duty and S_SOD stands for static separation of duty.

4.2 The Steps of Derivation Process

The complete steps of the derivation process are shown in Fig. 12. The 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 the reduced enterprise model (Steps. 2-7). Initial input of the process is the pre-constructed enterprise model and user information. Final output of the process is SQL script that makes T-RBAC data into the database or text files that contain T-RBAC schema information. The sign '⊕' means that the step can be processed automatically without human intervention.

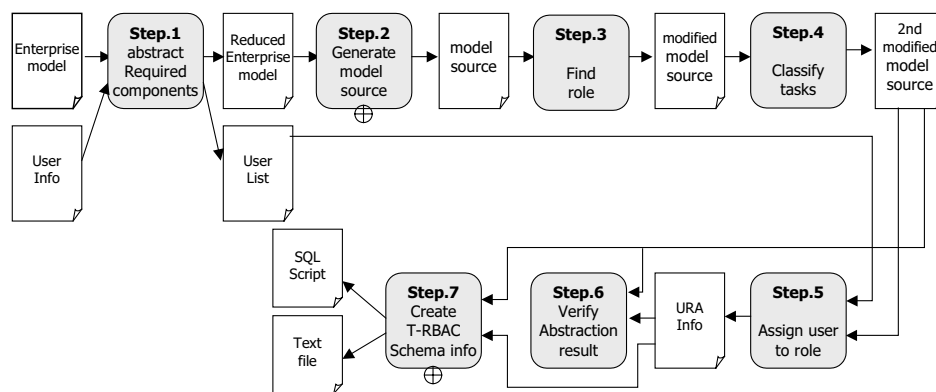


Fig. 12. Derivation process of T-RBAC schema information.

Now we describe each step of the derivation process of the T-RBAC aspects. The 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 Abstract required components

- **Input:** Enterprise model, User Information
- **Output:** Reduced enterprise model, User list
- **Related Diagram:** -
- **Derived T-RBAC Components:** User

The main work of step. 1 is to derive T-RBAC-related components from the enterprise model reduction to the reduced enterprise model. The enterprise model can be constructed using various tools, or methodologies. If the reduced enterprise model diagrams have been used in the construction of enterprise model, step.1 would be simple. Further, if the enterprise model includes diagrams from Figs. 8-11, step. 1 can even be processed automatically. Otherwise, human experts, in general the developers of the enterprise model, should perform step. 1 manually. Generally, user information is not included in

the business model. Thus, user information needs to be added. The user list in the output includes 'user id', 'organization name', and 'job position'.

Step. 2 Generate model source

- **Input:** Reduced enterprise model
 - **Output:** Model source
 - **Related Diagram:** All four Reduced enterprise model diagrams
 - **Derived T-RBAC Components:** Information objects, Task, Workflow, PTA
-

Most of the enterprise modeling tools support the function of translating the graphical model to a text source file. With this tool, step. 2 can be processed automatically. The box below shows a part of the example source file created by the Rational Rose Tool. It is translated from the information object diagram (Fig. 4). From the model source file we can derive T-RBAC information such as information object, task, workflow, and 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)
 - **Related Diagram:** Organization diagram, Task diagram
 - **Derived T-RBAC Components:** Role, RH, TRA
-

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

Step 3.1 Create roles from the task diagram.

The task diagram contains executor components. We can choose the executor names as role names. Four types of role are created according to the characteristics of the executors.

Step 3.2 Assign tasks to roles (TRA) from the task diagram.

In the task diagram the executor (role) is related to the task, an executor can be related to many tasks, and a task can be related to many executors.

Step 3.3 Create the initial role hierarchy from the organization diagram.

The organization diagram implies 'organizational role' and 'job position role'. Therefore, the initial role hierarchy contains two types of roles. For example, the organization structure in Fig. 8 is translated to the role hierarchy in Fig. 13.

Step 3.4 Unify roles that contain the same tasks.

If role A and role B have the same tasks, we can assume that role A and role B are the same role, and therefore these roles should be unified.

Step 3.5 Adding some roles to role hierarchy.

The initial role hierarchy does not contain some business roles or some executors in the task diagrams. Because they are not shown in the organization diagram and there is no information to add them in the role hierarchy, human experts should perform step 3.5.

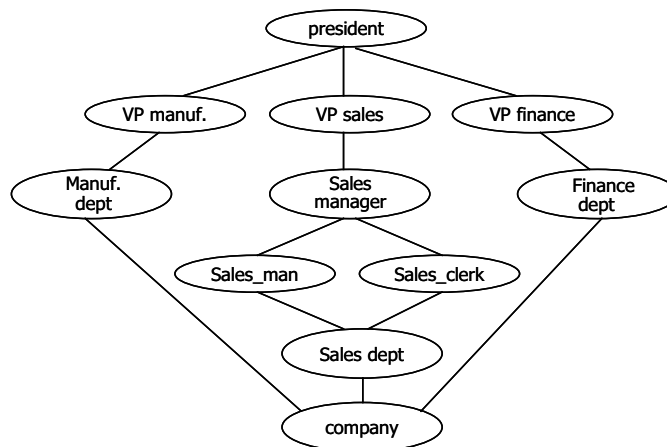


Fig. 13. Initial role hierarchy example.

Step. 4 Classify tasks

-
- **Input:** Modified model source (1)
 - **Output:** Modified model source (2)
 - **Related Diagram:** Organization diagram, Organization diagram
 - **Derived T-RBAC Components:** (classified) Task
-

In the T-RBAC model, tasks are classified into three classes. Different access controls are applied to tasks according to their class. There are four classification rules.

- Rule 1. Tasks belonging to business process and assigned to 'business role' roles are class W
- Rule 2. All tasks executed by organization roles belong to class S (because they have the inheritance characteristic)
- Rule 3. Tasks belonging to business process and assigned to 'job position' roles or 'organizational' roles are class A
- Rule 4. Others belong to class P

The above algorithmic classification is not complete. Therefore, if necessary, a human expert executes modification.

Step. 5 Assign user to role

- **Input:** User list, Modified model source (2)
 - **Output:** Modified model source (2)
 - **Related Diagram:** User list, Organization diagram
 - **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 business roles.

Step. 6 Verify abstraction Result

- **Input:** User list, Modified model source (2)
 - **Output:** Verified model
 - **Related Diagram:** All four diagrams
 - **Derived T-RBAC Components:** -
-

After completing step.5, verification is required for the result. We can use the 'completeness state checking rules' of Appendix C for verifying the abstraction result.

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

- **Input:** User list, Modified model source (2)
 - **Output:** T-RBAC SQL script, T-RBAC schema files
 - **Related Diagram:** -
 - **Derived T-RBAC Components:** -
-

The final step is to create an SQL script for generating T-RBAC information in the database. T-RBAC schema files, which have the same information with SQL script, can be created in exchange of SQL script. Sample script file and schema files are as follows

```

CREATE USER 'S001' IDENTIFIED BY 'qwee';
CREATE USER 'S002' IDENTIFIED 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

Here we suggest a practical solution to derive the T-RBAC information from the enterprise environment; an example of this method can be found in Appendix A. During our research, we recognized some points as follows.

A good enterprise model produces good T-RBAC information. The enterprise model is a realistic input source for our methodology, and the derivation process is executed semi-automatically. Therefore, the quality of T-RBAC information depends on the quality of the enterprise model.

A good description of the task is the most important thing in our derivation process. As can be seen in Fig. 14, the task is the central concept of access control design. In general, the data flow diagram in the enterprise model implies task information. The main issue is that task can be described at various levels. For example, two tasks 'order_update' and 'order_cancel' can be described as a single task 'order_manage', so determining the appropriate task level is important. This issue needs more research.

Total automation of the derivation process is very difficult. The ideal goal of our derivation process is total automation, but this requires enormous material information. There is a trade-off between the degree of automation and the cost of maintenance that requires material information.

Supporting tools are necessary. Our derivation process deals with many related components, and therefore manual work is difficult and time consuming. Supporting tools are an essential component of our methodology. In the next section, we show the implementation result of the supporting tools.

The reduced enterprise model can be used for managing (updating) the T-RBAC schema information following a change in the real world. A reduced enterprise model is necessary the first time the T-RBAC information is built. Even after the

building process, the reduced enterprise model will be used for other purposes. Maintenance of T-RBAC information will still be required to account for the changes in the real world. In this case, when the security administrator modifies the reduced enterprise model reflecting the changes in the real world, the T-RBAC schema information will be updated automatically. A management module will be added to our supporting tools in future research.

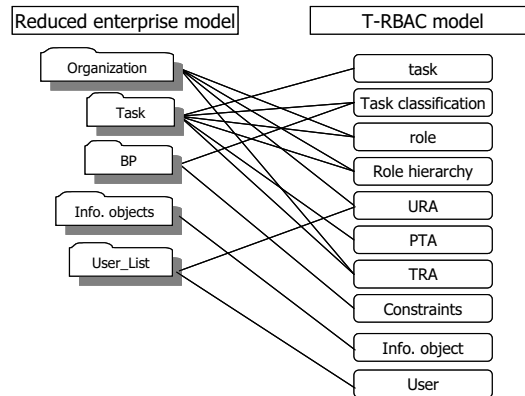


Fig. 14. Relationship of reduced enterprise diagram & T-RBAC components.

The derivation method of this section is only the basic skeleton. If the user wishes to apply our derivation method to the real world, the user will need to elaborate our theory in some areas, such as role finding, task classification, and user-role assignment.

6. IMPLEMENTATION

Fig. 15 shows a more detailed derivation process of the T-RBAC schema information. The ARIS tool produces text source files from the inserted enterprise model diagrams. The Visual C++ language is used to program translation from text source files to the temporal database. The deriving tool, which is created by Power Builder, produces the final T-RBAC schema information from the temporal database. We show a case of derivation process in Appendix A.

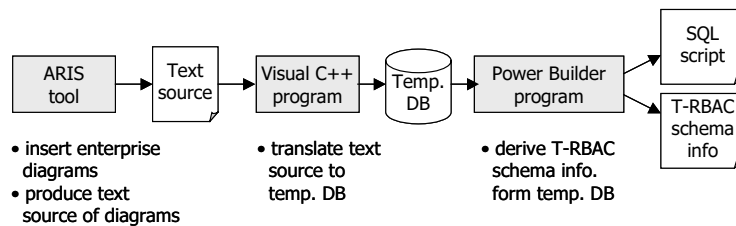


Fig. 15. Derivation process of T-RBAC schema information.

The ultimate objective of our proposed method is to combine security engineering and software engineering, the intermediate concept being a business model. The proposed method may be used as a basis for business-model-based access control. In the model-based access control, security administrators deal with the business model instead of dealing with access control data as shown in Fig. 16. In this case, updating the business model leads to changes in the access control data.

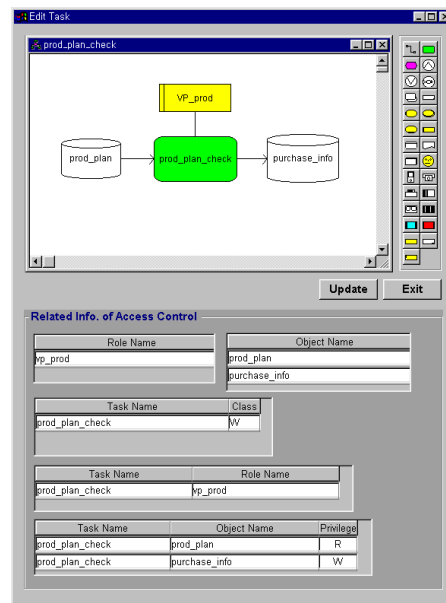


Fig. 16. Example of model-based access control.

7. CONCLUSIONS AND 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 activity. 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 suggest possibility of model-based administration of (T-)RBAC. 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.

Our approach can reduce the burden of security administration in the large enterprise organizations. Security officers deal with abstract business model rather than raw security information. Also our approach has strong adaptability for changing of business processes in the real world.

Developing model-based administration tool is our final goal. We implemented deri-

vation 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].

ACKNOWLEDGEMENT

This work was supported by grant No. 2000-1-303-001-3 from Basic Research Program of the Korea Science and Engineering Foundation.

REFERENCES

1. C. P. Pfleeger, *Security in Computing*, second edition, Prentice-Hall International Inc., 1997, pp. 244-250.
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, Proceedings of 11th International Conference, DEXA 2000*, 2000, pp. 264-273.
3. S. Oh and S. Park, "An integration model of role-based access control and activity-based access control using task," in *Proceedings 14th Annual IFIP WG 11.3 Working Conference on Database Security*, Aug. 2000, pp. 557-569.
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," in *Proceedings of 5th ACM Workshop on Role-Based Access Control*, 2000, pp. 103-110.
5. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control method," *Magazine of IEEE Computer*, Vol. 29, No. 2, 1996, pp. 38-47.
6. Ferrario, J. Cugini, and R. Kuhn, "Role-based access control (RBAC): Features and motivations," in *Proceedings of 11th Annual Computer Security Application Conference*, 1995.
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," in *Proceedings 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," in *Proceedings of 3rd ACM Workshop on Role-Based Access Control*, 1998, pp. 33-40.
11. J. D. Moffett, "Control principles and role hierarchies," in *Proceedings of 3rd ACM Workshop on Role-Based Access Control*, 1998, pp. 153-160.

APPENDIX A

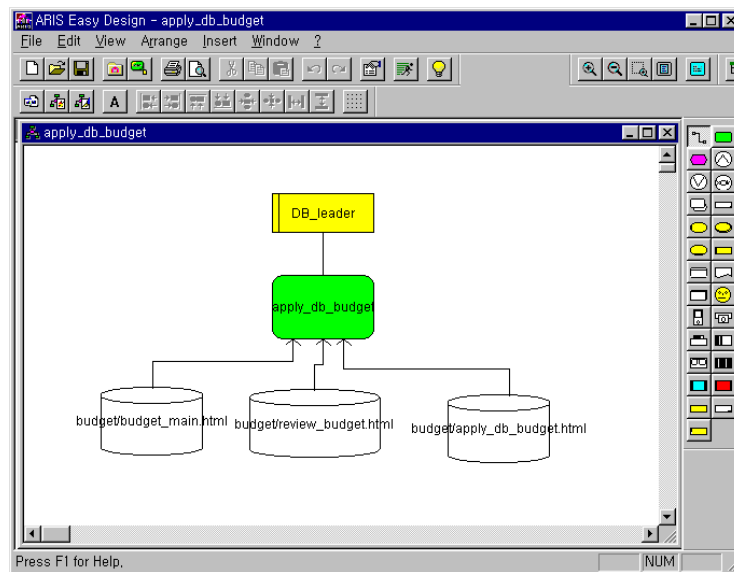
As a target real world for derivation process, we choose BK21(Brain Korea 21) web site for Sogang university. The URL is http://dmlab.sogang.ac.kr/trbac/BK21/rbac_bottom.htm that is a mirroring site of <http://dmlab.sogang.ac.kr/BK21/main.html>. Three laboratories take part in the BK21 project. The target objects of access control are html or asp documents. A simple statistics of BK21 web site is as follows.

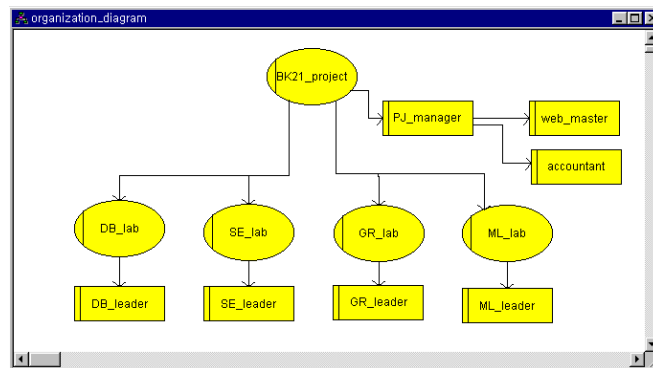
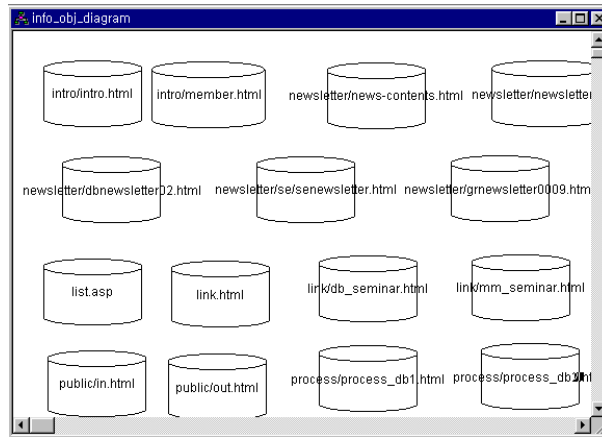
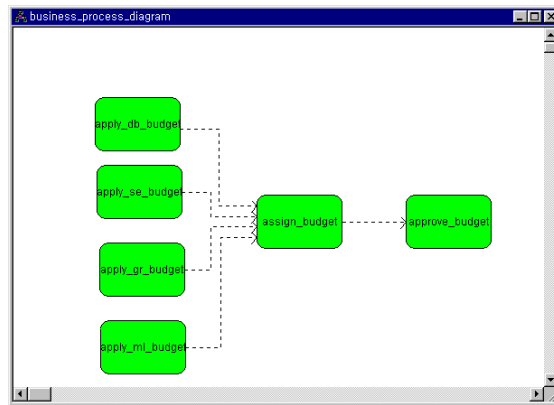
Number of users	21 (professors + graduate students)
Number of documents	120 (*.html, *.asp)
Number of roles	12
Number of tasks	18

We modeled management of BK21 project and derived access control information by our derivation tool. We will show each step of derivation process based on derivation tool.

(1) Create a reduced enterprise model (designed by ARIS Easy Design)

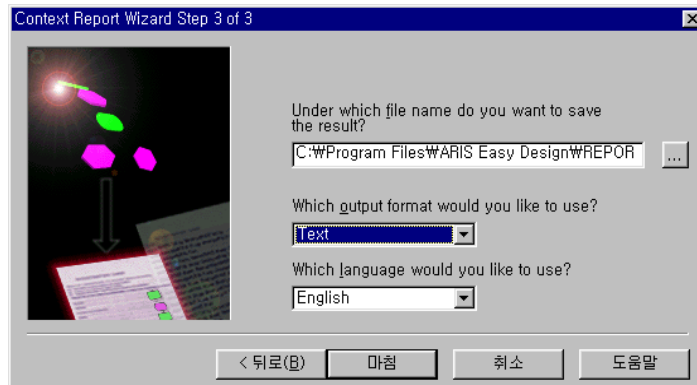
We inserted reduced enterprise model by 'ARIS Easy Design' that is a modeling tool. Below diagrams are example of task diagram, business process diagram, information object diagram, and organization diagram.





(2) Generate model source

We use report generation function, which is supported by ARIS tool.

**** Example of model source file**

```

Demoversion Information Nr.0
ARIS Report
Created:
Date: 01-05-26
Specific time: PM 3:31:08
Demoversion Information Nr.1
Database info:
Server: LOCAL
Database: Bk21
Users: system

apply_gr_budget eEPC
Attributes:
Objects:
Demoversion Information Nr.2
apply_gr_budget Function
Relationships
is executed by
GR_leader Position
gets input from
budget/review_budget.html Information carrier
gets input from
budget/budget_main.html Information carrier
gets input from
Demoversion Information Nr.3
budget/apply_gr_budget.html Information carrier
budget/apply_gr_budget.html Information carrier
Relationships
provides input for
apply_gr_budget Function

budget/budget_main.html Information carrier
Relationships
provides input for
Demoversion Information Nr.4
apply_gr_budget Function

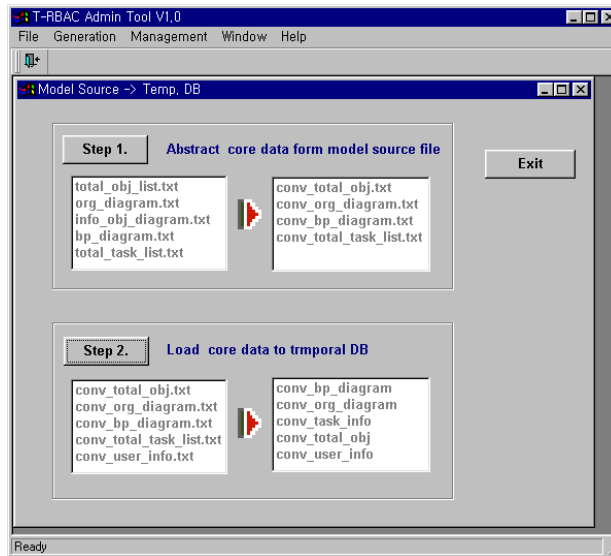
budget/review_budget.html Information carrier
Relationships
provides input for
apply_gr_budget Function

GR_leader Position
Relationships
Executes
Demoversion Information Nr.5

```

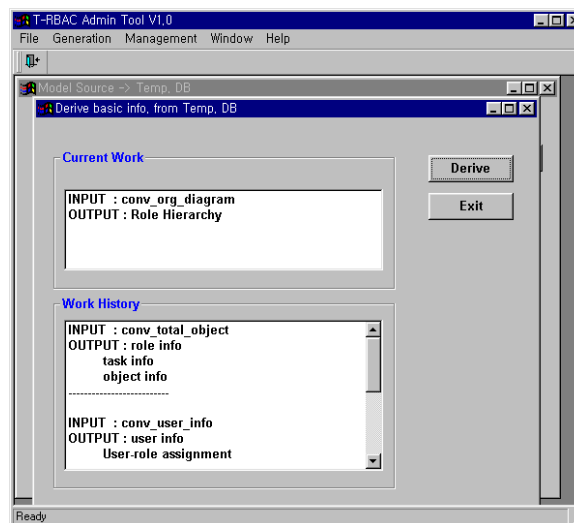
(3) Abstract core data and load it to temporal DB

Our derivation tool reduces useless data from model sources files and loads useful data to temporal database of derivation tool.



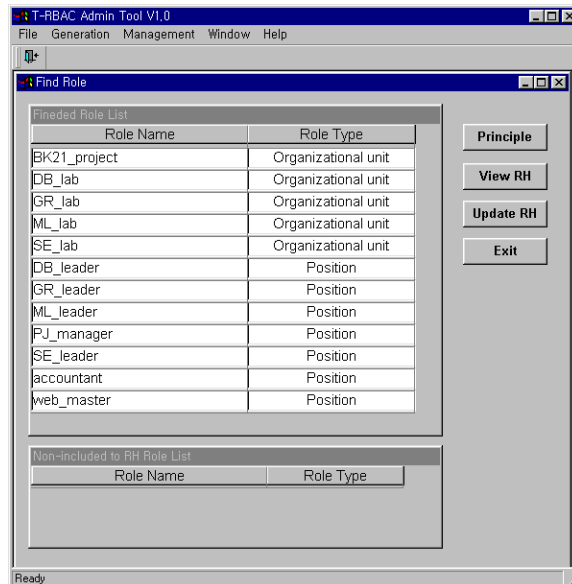
(4) Derive basic information from temporal DB

Temporal DB implies basic information such as users, roles, tasks, and objects. These are derived very easy way.

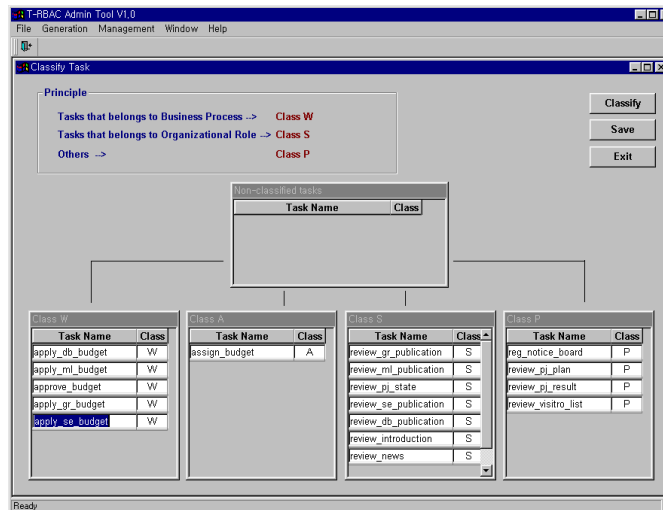


(5) Find role

Derivation tool shows found roles and supports inserting unfound roles.

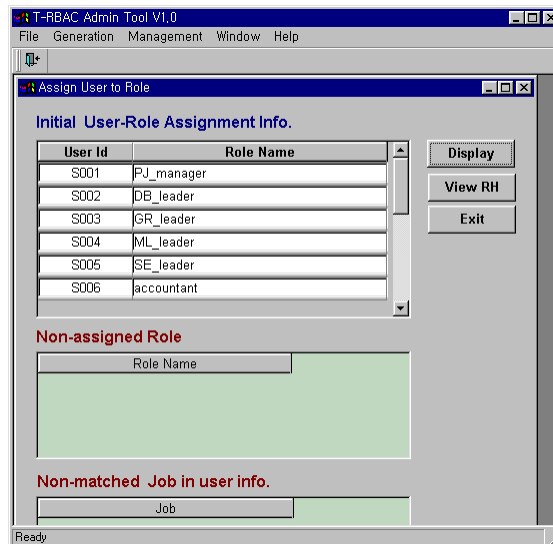
**(6) Classify task**

Derivation tool can classify tasks according to classification rules. Then security administrator adjusts the classification result.



(7) Assign user to role

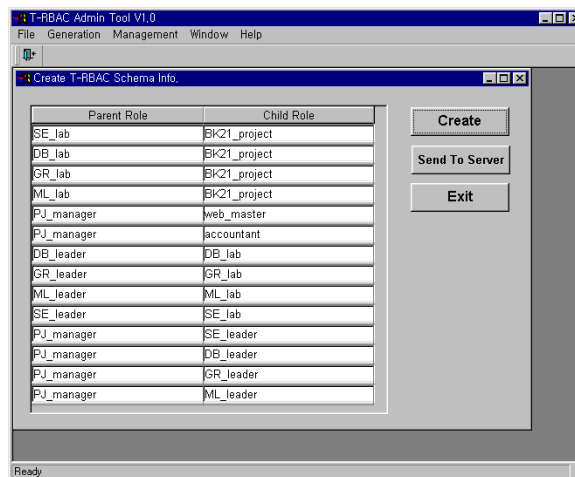
Derivation tool compares user information and task/role information. As a result, user-role assignment (URA) information is created. Security administrator can add new URA information by adjusting unmatched information.

**(8) Verify abstracting result**

We do not implement this step yet.

(9) Create T-RBAC schema information

Deriving tool creates T-RBAC schema information and sends it to web server which has a T-RBAC access control engine.



**** Example of T-RBAC schema file**

file name : TROLE.TXT

BK21_project,2,2,ORGANIZATION
 DB_lab,2,2,ORGANIZATION
 GR_lab,2,2,ORGANIZATION
 ML_lab,2,2,ORGANIZATION
 SE_lab,2,2,ORGANIZATION
 DB_leader,2,2,POSITION
 GR_leader,2,2,POSITION
 ML_leader,2,2,POSITION
 PJ_manager,2,2,POSITION
 SE_leader,2,2,POSITION
 accountant,2,2,POSITION
 web_master,2,2,POSITION

file name : TTRA.TXT // Task-Role Assignment

accountant,assign_budget
 BK21_project,review_news
 BK21_project,review_publication_list
 BK21_project,review_pj_state
 BK21_project,review_introduction
 DB_lab,review_db_publication
 DB_leader,apply_db_budget
 DB_leader,review_pj_result
 GR_lab,review_gr_publication
 GR_leader,review_pj_result
 GR_leader,apply_gr_budget
 ML_lab,review_ml_publication
 ML_leader,apply_ml_budget
 ML_leader,review_pj_result
 PJ_manager,approve_budget
 PJ_manager,review_pj_plan
 SE_lab,review_se_publication
 SE_leader,review_pj_result
 SE_leader,apply_se_budget
 web_master,reg_notice_board



Sejong Oh earned his Ph.D. in the Department of Computer Science from Sogang University in 2001. He is currently a Post Doctor researcher of the School of Information Technology and Engineering, George Mason University. His main research interests include access control for enterprise and distributed systems, ERP, secure DBMS, and internet security.



Seog Park is a Professor of Computer Science at Sogang University. He received the B.S degree in Computer Science from Seoul National University in 1978, the M.S. and the Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and Technology(KAIST) in 1980 and 1983, respectively. Since 1983, he has been working in the Department of Computer Science of the College of Engineering, Sogang University. His major research areas are database security, real-time systems, data warehouse, digital li-

brary, multimedia database systems, role-based access control and Web database. Dr. Park is a member of the IEEE Computer Society, ACM and the Korea Information Science Society. Also, he has been a member of Database Systems for Advanced Application (DASFAA) steering committee since 1999.