SACMAT 2001: 6th ACM Symposium on Access Control Models and Technologies, Chantilly, VA, USA, ACM.

The Role-Based Access Control System of a European Bank: A Case Study and Discussion

Andreas Schaad
Department of Computer Science
University of York
York, YO10 5DD, UK

andreas@cs.york.ac.uk

Jonathan Moffett
Department of Computer Science
University of York
York, YO10 5DD, UK

jdm@cs.york.ac.uk

Jeremy Jacob
Department of Computer Science
University of York
York, YO10 5DD, UK

jeremy@cs.york.ac.uk

ABSTRACT

Research in the area of role-based access control has made fast progress over the last few years. However, little has been done to identify and describe existing role-based access control systems within large organisations. This paper describes the access control system of a major European Bank. An overview of the system's structure, its administration and existing control principles constraining the administration is given. In addition, we provide an answer to a key question – the ratio of the number of roles to the system user population – which was raised in the recent RBAC2000 Workshop. Having described certain weaknesses of the Bank's system, the case study is extended to a comparison between the system and the RBAC96 models. In particular the issues of inheritance and grouping are addressed.

General Terms

Security

Keywords

role-based access control, control principles, separation of duties, dual control, number of roles, role administration, inheritance, least privilege

1. INTRODUCTION

Role-based access control is a well-defined research area and there is an on-going effort in the definition of a role-based access control standard [1]. Broadly accepted models exist [2], [3], [4]. Research and commercial tools and applications have been developed to help with the engineering [5], [6] and management of roles [7]. However, often research tools work with minimal testing datasets as no real figures for the number of users, roles and permissions in commercial systems have been published. Companies that have successfully deployed access control systems are often unwilling to provide descriptions and figures for their role-based systems for security reasons. So researchers use toy examples that fail to reflect the complexity of large industrial organisations.

This leads to a lack of credibility from the user side. Researchers cannot provide evidence that their tool works as well in a real environment as under laboratory conditions. Providing a concrete example and realistic figures can help us to mitigate that situation. On the basis of these numbers more realistic datasets and examples could be generated that serve as a test bed for tools. In contrast to other descriptions of role-based access control implementations such as in [8] or [9], we stress the fact that the system we describe in this paper is not specific to a single application or operating system, but provides access control services on an enterprise-wide level to a variety of systems and applications.

The rest of this paper is structured as follows. In section 2 we provide a case study of a real-world access control system as it is used in a major European bank. Sections 2.1 – 2.3 describe the background and structure, and give an application example. The number of users, roles and permissions and their relationship is given and evaluated in section 2.4. In section 2.5 we describe the administration of the system and how it is controlled by constraints such as Separation of Duties, Dual Control and Least Privilege. Goals for further development of the system, to address weaknesses in the current system, are discussed in section 2.6. Section 3 compares and contrasts this system with Sandhu's RBAC96 access control models, addressing the issues of inheritance of access rights in section 3.1 and the grouping of users in section 3.2. We finish the case study with a summary and conclusion in section 4.

2. THE FUB ACCESS CONTROL SYSTEM

2.1 Background

The particular case study that we present in this paper was carried out in co-operation with Dresdner Bank, a major European bank with 50,659 employees and 1,459 branches world-wide. The main business, with about 6.5 million private customers and 1,000 branches, is situated in Germany, the rest in Europe and overseas. The Bank uses a variety of different computing applications to support its business, many of which have their origin in the

SACMAT '01, May 3-4, 2001, Chantilly, VA.

mainframe world, but also more recently deployed client-server based systems.

Before 1990, most of the host-based applications used the local access control file administered at the relevant host for the determination of access rights. For each employee, the access rights had to be administered manually at the individual application level. This caused enormous administrative overheads as a result of the growing number of people working with these applications. Additionally the maintenance of several application-level security files for each user was an error-prone process and could not be justified within the general security policy framework. In order to improve this situation a system, called the FUB (= Funktionale Berechtigung), was developed by the Bank as no suitable commercial solutions were available at that time. In this system access rights are given to the individual user according to a combination of his job function and official position within the organisation. The direct assignment of access rights to the individual user at the application level was discontinued.

The FUB is an example of an enterprise-wide role-based access control system. Applications cannot make access control decisions on their own. They grant access to data on the basis of a centrally provided security profile. Over 60 applications within the bank make use of this system. These cover a wide area of organisational functions such as private customer instruments at the local branch, credit data checks, automated signature approval and the administration of Unix accounts. An application is launched by a user who first identifies and authenticates himself to it. Initially the application has no knowledge of any relevant access permissions the user might possess. It queries the FUB about the security profile of the current user in order to obtain this information.

Since 1990 (and thus much earlier than most of the published role-based access control discussion), the FUB has hosted roles and delivered access rights for usage within other applications that run under various environments such as UNIX derivatives (SINIX/AIX) or WINDOWS NT.

On average 42,000 security profiles per day are distributed by the FUB. The time needed by the system to determine one individual security profile is approximately 85 ms. The system's availability rate is 99% per year.

2.2 The Basic System Structure

The FUB is a role-based access control system. Roles are defined as a combination of the

- official position and
- job function

Typical official positions could be that of the ordinary Clerk, Group Manager or Regional Manager. Functions represent their daily duties such as being a financial analyst, share technician or internal software engineer. Additionally the organisational unit to which a user belongs is used as an access control criterion for certain applications. All these data are defined and maintained in the human resources database. A batch job runs between the human resources system and the FUB every night. Thus, the access control system has a very accurate image of the current organisational status and existing roles. Within the FUB the data delivered by the human resources database are linked to applications. Access rights are assigned to applications. When a user starts an application the FUB delivers the security profile that tells the application which individual access rights the user possesses. Figure 1 shows the basic architecture of the system and its interfaces to the human resources database and individual applications.

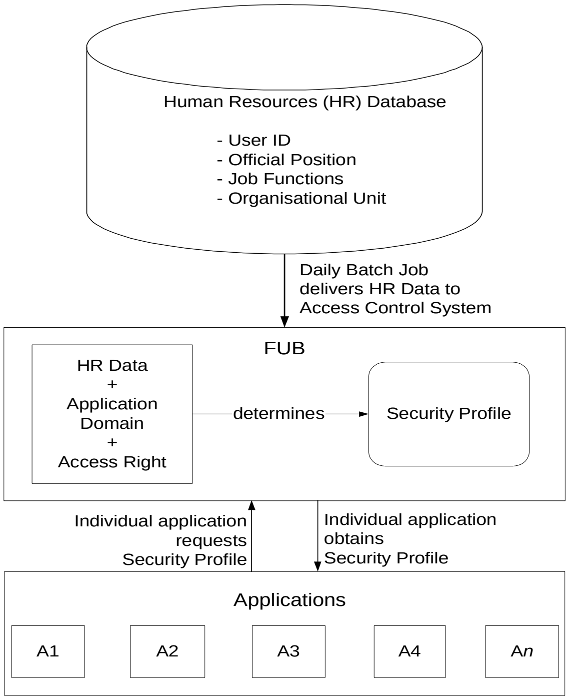

Human Resources (HR) Database
- User ID
- Official Position
- Job Functions
- Organisational Unit
Daily Batch Job delivers HR Data to Access Control System
FUB
HR Data + Application Domain + Access Right
determines
Security Profile
Individual application requests Security Profile
Individual application obtains Security Profile
Applications
A1
A2
A3
A4
An

Figure 1 : The basic structure of the FUB

Employees belong to organisational units. Ideally each employee is only assigned to one role. However, in special circumstances an employee can be given up to four roles (e.g. in the case of illness of a colleague). Several applications can be accessed through a role. Each application has a set of access rights assigned to it. A simplified underlying data model is shown in Figure 2.

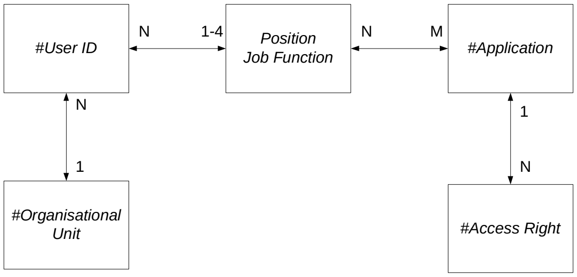


Figure 2 : The FUB data model

2.3 An Application Example

To make the above technical description more concrete we provide a scenario that reflects the daily business at a high-street branch of Dresdner Bank (Figure 3). An existing bank client wishes to discuss his personal savings situation with the branch's financial advisor. The advisor and the client go to a meeting room which contains a Personal Computer. The advisor identifies and authenticates himself to the machine using a smartcard and his password. He launches an application that allows him to enter the records of his client which are stored on a central server.

When the application is launched it issues a request to the host, querying which rights the advisor has within the application domain. The application request contains the personnel number, which was obtained during the identification and authentication process. Also the application identifier is submitted to obtain the relevant authorisation profile for the application. Once the FUB has used these data to deliver the security profile, the application knows which access rights are assigned to the role of the user and allows him to execute his access rights accordingly. In this particular case information about the relevant organisational unit to which the advisor belongs (here, the branch) will prohibit him from accessing account data outside his branch. His access rights are confined within the organisational domain of the branch. However, other applications can be used from access points all over the bank, as the access rights which are granted for them do not depend on any local information.

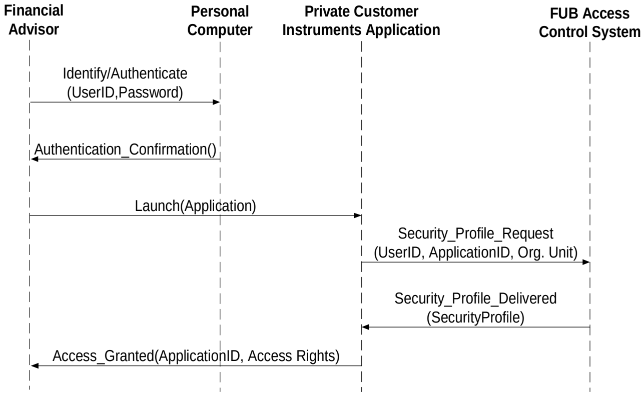

Financial Advisor
Personal Computer
Private Customer Instruments Application
FUB Access Control System
Identify/Authenticate (UserID,Password)
Authentication_Confirmation()
Launch(Application)
Security_Profile_Request (UserID, ApplicationID, Org. Unit)
Security_Profile_Delivered (SecurityProfile)
Access_Granted(ApplicationID, Access Rights)

Figure 3 : An access request

One weakness of the current FUB system is, as we show in Figure 2, that a user can be assigned to more than one role. This could happen when a colleague becomes ill or is on holidays, but also in more permanent cases where a clerk works in branch A in the morning and branch B in the afternoon. In the RBAC96 model a user must choose which role to activate for a session. However, the session concept does not exist in the FUB. So when a user logs onto a system he has all the access rights of all the roles to which he is assigned. This creates problems with respect to the principle of Least Privilege and Separation of Duties. Careful administration and monitoring of user/role assignment is needed to prevent security violations and conflicts.

2.4 The Number of Roles

A role in the FUB system is defined using the official position within the organisational hierarchy and a description of the job function. These data are delivered by the human resources system. From now on we will refer to a role using the *construct function/Official Position*. We will use lower case for functions and title case letters for positions. An example of this would be *financial analyst/Group Manager* indicating that somebody has the function of being an analyst and holds the official position of a group manager. Theoretically the total number of roles would be the product of every official position and every function. However, the actual number of roles is a subset of this, as certain possible roles such *as secretary/Member of the Board* do not occur in reality.

Within the Bank there are 65 official positions that can range from an ordinary Clerk in a branch, through the Branch Manager, to a Member of the Board. This can be represented as a partial order. Hierarchies arise with further organisational indicators such as cost centres, departments or project groups.

These positions are combined with 368 different job functions provided by the human resources database. Although there would be a possible set of 23,920 roles, the number of roles that are currently in use is about 1300. As the access control system is constantly updated it is subject to changes occurring when functions and positions are created, and more importantly, deleted.

Each night human resources data about employees, their function, rank and organisational unit are transferred into the FUB. However, not every single employee can actually be seen as an active user of the FUB system (e.g. cleaning staff, catering, internal postal services etc.). Thus, there are only about 40,000 users in the FUB.

These figures match an oral estimate that was given at the RBAC2000 Workshop [10], suggesting that the number of roles in a role-based system is approximately 3-4% of the user population. With 40,000 FUB users and 1300 Roles, we obtain a role/user ratio of approximately 3.2%. However, this distribution is not really uniform due to the pyramid shaped official position hierarchy of any organisation. There are always many more clerks than Head of Divisions in an organisation and so we have many more roles such as *financial analyst/Clerk* than *financial analyst/Head of Division* roles. It might be interesting to further analyse the role/user ratio according to the position hierarchy, yet in our case we concentrated on the ??absolute?? role/user ratio only as we do not have the needed information for any further analysis.

Another issue in the administration of the system is that it has also to provide access control services to users which can not be considered as permanent staff. This group of users includes third party consultants, temporary staff and freelancers. They work for the Bank during projects with a varying length from several weeks to years. Apart from obtaining the needed manpower, hiring a consultant or freelancer is also a form of outsourcing. Many of them come for the duration of a project, leave, and often come back shortly after the project has finished, working for another, sometimes related project.

Ideally, this group of users should always get a new account when starting to work and their accounts should be deleted when leaving the project. However, this creates overheads. Thus, information about this group is not held in the Human Resources database, but is locally administered by the FUB staff. User accounts are created once and are usually kept (although not activated) when a consultant leaves.

The overheads occurring with the administration of these users should not be underestimated as this user group, containing hundreds of users at a time, is subject to constant changes. However, we did not take this issue into consideration when providing above figures. Thus the role/user ratio only applies to full-time staff.

2.5 Role Administration and Control Principles

The actual definition of roles and the assignment of users and access rights to the role occurs at different levels within the organisation (Figure 4).

- Human Resources Department
  Role Definition and *User/Role Assignment*.
- Application Administration
  Access Right Definition and *Application/Access Right Assignment*.
- FUB Administration
  Role/Application Assignment.

Users and roles are initially created in the human resources department. A user is given a unique number that serves as his user identification number. Also the roles are defined there, maintaining and combining functions and official positions. A user is then assigned to a role. This makes sense and reflects the close relation between role-based access control and other areas such as human resources and organisation.

The assignment of access rights to a role is done for each individual application through the application administrator. This takes the form of assigning a set of numbers, representing specific access rights, to an application identifier (e.g. *PKI = Private Customer Instruments*). The semantics of these numbers are only known to the application administrator (e.g. *203 = Create new account in the Private Customer Instruments Application*). The benefit of this is that it remains unknown to the FUB administrator responsible for the role/application assignment what access right a specific number represents within the application domain (Figure 5). In addition the application administration process is subject to the principle of Dual Control. One person can alter data, whereas a second person has to confirm these data.

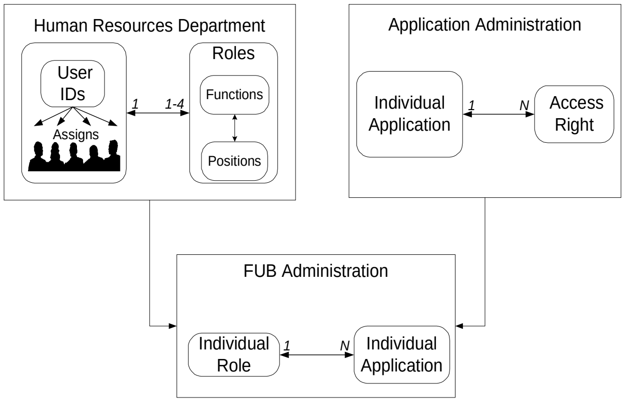


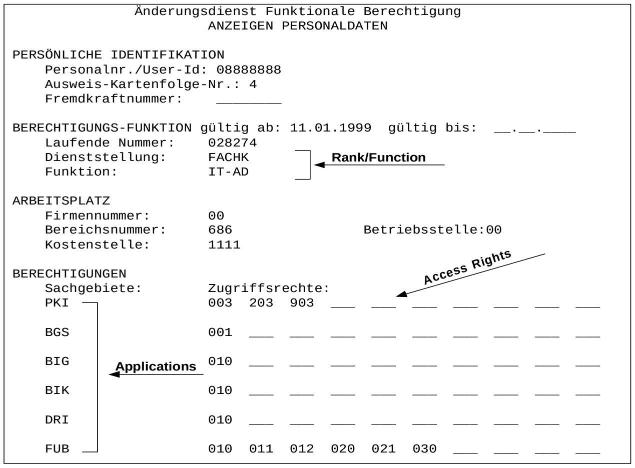
Figure 4 : Access Control Administration

The FUB administrator can only do the role/application assignment and thus there is a strong level of Separation of Duties in the entire administration process. In addition, the generation of evidence by logging any administrative actions is obligatory and has to follow the Bank's policy on logging and evidence generation.

However, some of these control principles are clearly broken when it comes to the administration of freelancers or consultants as outlined in section 2.4 (Not shown in Figure 4). This group is locally administered by the FUB staff. Here, the FUB administrators knows about the identity of a user. A user identifier is given to the user by the administrators and applications and are assigned as needed when the consultant works on a project. The bank is aware of this problem and will address it in the future.

Note the distinction between the principles of Separation of Duties and Dual Control in this context. Separation of Duties requires a task to be split into separate sub-tasks, with different people required to carry out the sub-tasks. On the other hand, the principle of Dual Control demands the participation of at least two people in the completion of a single task.

The task of revocation of role and permission assignments is done very elegantly in the bank's system. By coupling the information delivered by the human resources department any user/role assignment ceases to exist when the user leaves the company. Also any user/role, role/application assignment becomes invalid when the organisational structure changes. This tight coupling bears certain dangers, because deleting a user from the human resources database will subsequently delete all his work when he ceases to exist in the system. Additional controls and organisational measures are needed to prevent valuable work from being lost.



```
               Änderungsdienst Funktionale Berechtigung
                        ANZEIGEN PERSONALDATEN

PERSÖNLICHE IDENTIFIKATION
    Personalnr./User-Id: 08888888
    Ausweis-Kartenfolge-Nr.: 4
    Fremdkraftnummer:    ________

BERECHTIGUNGS-FUNKTION gültig ab: 11.01.1999  gültig bis:  __.__.____
    Laufende Nummer:    028274
    Dienststellung:     FACHK
    Funktion:           IT-AD

ARBEITSPLATZ
    Firmennummer:       00
    Bereichsnummer:     686                 Betriebsstelle:00
    Kostenstelle:       1111

BERECHTIGUNGEN
    Sachgebiete:        Zugriffsrechte:
    PKI                 003  203  903  ___  ___  ___  ___  ___  ___  ___

    BGS                 001  ___  ___  ___  ___  ___  ___  ___  ___  ___

    BIG                 010  ___  ___  ___  ___  ___  ___  ___  ___  ___

    BIK                 010  ___  ___  ___  ___  ___  ___  ___  ___  ___

    DRI                 010  ___  ___  ___  ___  ___  ___  ___  ___  ___

    FUB                 010  011  012  020  021  030  ___  ___  ___  ___
```

Figure 5 : A FUB Administration Screenshot

2.6 System Development Goals

In the current system access rights can only be allocated to the combination of function/hierarchical position and organisational unit. Further possibilities of allocating access rights, especially on a per user basis, do not exist. This would be a feature that violates the principle of separation of users from access rights by means of roles. However, in the case of certain access rights (e.g. set of access rights representing the power of attorney for an employee) it is desirable to assign these directly to the user.

2.6.1 Access right allocation

In the current system access rights can only be allocated to the combination of function/hierarchical position and organisational unit. Further possibilities of allocating access rights, especially on a per user basis, do not exist. This would be a feature that violates the principle of separation of users from access rights by means of roles. However, in the case of certain access rights (e.g. set of access rights representing the power of attorney for an employee) it is desirable to assign these directly to the user.

2.6.2 Grouping mechanisms

(a) Grouping employees

In the current system, employees can only be grouped according to the combination of function/hierarchical position and organisational unit. It is not possible to group employees according to other criteria and assign group-specific access rights. A grouping mechanism will provide ease of administration as only the group needs to be assigned a role and not each individual.

(b) Grouping access rights

The current system does not allow for the grouping of access rights which naturally belong together. An example is the grouping of access right 101 for *create account,* and 102 for *delete account* into a single group G1 for *account manipulation*. These grouped rights will provide an easier assignment of access rights to roles.

2.6.3 Covering/Standing-in regulations

In the current system regulations for the covering of one employee by another (e.g. holidays or illness) do not exist. There are a variety of unresolved issues. The ability to only partially delegate application-specific access rights is one of these. Another problem considers the ability of one person to stand in for two others at the same time. This might violate any Separation of Duties requirements.

2.6.4 Competences and Constraints

The current system does not allow the specification of competences and constraints in the security profile (e.g. authorisation to sign contracts up to DM 100,000 only).

2.6.5 Mapping of access control system to organisational structure

When mapping the access control system to the existing organisational structure we have to bear the following in mind:

(a) Continuous organisational change

(b) Flexible support of business strategies

(a) Continuous organisational change

Organisations of the size of Dresdner Bank are subject to a constant change process in their structural and functional organisation. This is due to a continuous orientation of the bank's business to the market needs. Also unforeseen acquisitions or mergers can cause a major organisational change. The current access control system does not provide the flexibility to meet these changes without a major administrative effort.

(b) Flexible support of business strategies

Certain strategies in the private customer business require that an employee can be related to a certain organisational structure. This could be in the form of branches or cost centres but also more abstract structures such as enterprise-wide working groups, task forces or projects. The access control system must be able to reflect these structures.

3. The FUB System and the RBAC Model

An obvious question is how far the Bank's access control system can be compared with other models of role-based access control. A candidate for comparison is Sandhu's RBAC96 model [2]. It is well-defined, easy to use and most importantly it can be configured to support various access control policies, according to the specific need. In addition it forms the basis for a proposed NIST standard [1].

We consider two issues:

- Access right inheritance through a role hierarchy;
- Groups of users.

3.1 Access Right Inheritance through a Role Hierarchy

The RBAC96 model has the concept of access right inheritance through a role hierarchy. If a partial ordering of roles is defined, then superior roles inherit all the positive access rights of their inferiors. When looking at Figure 2 we can immediately see that there is no role inheritance structure of this kind. The Bank's access control system does not offer any inheritance features. Why is this the case, and should it be changed?

In the RBAC96 model "Role" is an atomic concept, defined as "..a named job function within the organization ….". However, its natural counterpart in the FUB system is the FUB Role, which consists of *both* function and position; see examples in Table 1. The partial ordering of the FUB role can therefore be defined upon either or both of function and position. We discuss two possible orderings below.

Role	Function	Official Position
A	financial analyst	Clerk
B	financial analyst	Group Manager
C	financial analyst	Head of Division
D	financial analyst	Junior
E	financial analyst	Senior
F	financial analyst	Specialist
G	financial analyst	Assistant
…	…	…
X	share technician	Clerk
Y	support e-commerce	Junior
Z	office banking	Head of Division

Table 1 : Functions and Official Positions

3.1.1 Hierarchy of Official Positions

There is a strict partial order in the organisation of official positions (denoted by the > symbol). For example:

Head of Division > Group Manager > Clerk

This has little meaning by itself, but when combined with function it is often reflected by a real hierarchy of actual power. For example, the *financial analyst/Group Manager* role (which we will call Role B) has more power than the *financial analyst/Clerk* role (role A). Table 2 shows that Role B has as many or more access rights in the Money Market Instruments, Derivatives Trading and Interest Instrument applications and access rights to one more application (Private Customer Instruments). On the other hand there is, as we might expect, no similar hierarchical relationship of power between *office banking/Group Manager* and *financial analyst/Clerk* because they work in different functional areas.

Therefore we could define a role hierarchy in which one role is superior to another if its position is superior and their functions are identical. More formally, using the "." symbol as a selector:

$$Role(x) > Role(y) \Leftrightarrow Role(x).Position > Role(y).Position \wedge$$
$$Role(x).Function = Role(y).Function$$

Role	Application	Access Right
A	Money Market Instruments	1,2,3,4
	Derivatives Trading	1,2,3,7,10,12
	Interest Instruments	1,4,8,12,14,16
B	Money Market Instruments	1,2,3,4,7
	Derivatives Trading	1,2,3,7,10,12,14
	Interest Instruments	1,4,8,12,14,16
	Private Customer Instruments	1,2,4,7
…	…	…

Table 2 : Roles, Applications and access rights

In the example above, given that Group Manager > Clerk, we could economise on access right definition and Table 2 could be rewritten as Table 3.

Role	Application	Access Right
A	Money Market Instruments	1,2,3,4
	Derivatives Trading	1,2,3,7,10,12
	Interest Instruments	1,4,8,12,14,16
B	Money Market Instruments	7
	Derivatives Trading	14
	Private Customer Instruments	1,2,4,7
…	…	…

Table 3 : Rewritten Table 2, assuming that B inherits access rights from A

3.1.2 Hierarchy of Functions

There could also be a partial ordering of functions. An example for the two functions *inspector* and *finance accountant* would be*:*

inspector > finance accountant

This is an "*isa*" hierarchy, meaning that in order to carry out the functions of an *inspector*, one has to be a *finance accountant* and needs all the access rights of one. It is always true that an *inspector* is a *finance accountant* because otherwise he would not be competent to do the function. Regardless of official position it might therefore be a good idea to generate a role hierarchy based on functions (without regard to position), so that superior functions inherit access rights from their inferiors.

This, alternative, role hierarchy would be defined as having one role superior to another if its function is superior, regardless of position. More formally:

$$Role(x) > Role(y) \Leftrightarrow Role(x).Function > Role(y).Function$$

3.1.3 Discussion

In discussing the pros and cons of using an inheritance structure in the Bank, there are several issues to consider:

(a) Choice of role hierarchy

(b) Compatibility of role hierarchy with other organisational needs

(c) Fine grained access control

(d) Separation of duties and other control principles

(a) Choice of role hierarchy

For practical reasons the Bank should only make one choice for the role hierarchy (Although the RBAC96 model allows multiple hierarchies). We have noted two candidates above:

- Position hierarchy with matching function (3.1.1)
- Function hierarchy (3.1.2)

We do not have sufficient knowledge of the organisation to know which would be the best choice. It would be necessary to study the organisation's access control structure in detail, paying particular attention to the gained simplification in each case, and to the other issues which we discuss immediately below.

b) Compatibility of role hierarchy with other organisational needs

It has been recognised [2] that it is not possible to simply pick up an existing organisational hierarchy and import it into a RBAC system as conflicts might arise. We have given an example above (Section 3.1.1) in which the use of the position hierarchy with matching function for our role hierarchy would simplify the access rights table for *Group Manager*. However, we do not know whether this would also be appropriate for *Head of Division*. Perhaps access right inheritance would give him more rights than he needs to carry out his job, thus violating the principle of Least Privilege.

In order to deal with problems of this kind it may be necessary to create a hierarchy which is compatible with, but not identical to the original hierarchy using the concept of private roles as suggested in the RBAC96 model.

(b) Fine grained access control

If we institute access right inheritance, and subsequently it turns out that there is an access right which is needed for *financial analyst/Clerk* but not for the *financial analyst/Group Manager* role, then we need to deconstruct, wholly or partially, this portion of the role hierarchy. We cannot, in the RBAC model, solve the problem simply by giving a negative right to the superior position, because negative rights are inherited downwards.

(c) Separation of Duties

Access rights inheritance may cause a breach of the Separation of Duties principle. We pointed out in [11] how this can happen. An example is a software company as described in [12]. It is not desirable to let the project manager inherit from the senior programmer the right to read the repository. The project manager does not necessarily have the technical knowledge and might consider shipping bad code if he had access to it before it is released by the senior programmer.

We can be confident in the example of Table 2 that the Bank has avoided any Separation of Duties conflict, because it has strict internal control procedures. If access right inheritance were to be introduced, the Bank would have to re-examine the rights table in detail, in order to ensure that no conflicts of this kind were introduced. This is likely to restrict severely the extent to which inheritance can be introduced

3.1.4 Advantages and Disadvantages

We cannot reach any conclusion about this particular case, as we have neither the knowledge nor the authority to do so. In favour of introduction of a role hierarchy for the FUB is the economy of access rights that this would bring. We have however pointed out a number of factors which would reduce these advantages, and they would need to be examined in detail before reaching a decision.

3.2 Grouping

As mentioned in section 2.6.2, the Bank is considering introducing a mechanism for grouping of employees. A grouping mechanism will provide ease of administration as only the group needs to be assigned to a role and not each individual.

The RBAC96 model does not allow for the assignment of groups to roles. It explicitly states that the assignment of users to roles is a relationship between (individual) users and roles. However, other approaches to RBAC, e.g. [3] have recognised the value of using the group as a basis for the definition of roles. The concept of domains as described in [13] can be used to provide a mechanism for grouping users.

4. Conclusion

4.1 The FUB System

We have provided a case study of an access control system in a major European bank that makes use of a role-based approach to determine the access rights that a user possesses within an application domain. In this system a role is defined by function and position within the organisation. This access control system is closely linked to the human resources database, making it flexible towards certain organisational changes. Also the system works on an enterprise-wide organisational level and thus it provides services to a variety of applications on different platforms. We have given the concrete number of roles in a system that has been in use for over a decade now. These figures should give tool developers the chance to provide further evidence that their tools work even under real conditions. In addition we discussed the issue of inheritance and made a distinction between inheritance as it occurs along official positions and inheritance between functions.

4.2 RBAC Models

This case study has brought out several points about RBAC models. First of all, we have more confidence in the fundamental correctness of the role-based access control approach because of the existence of a large and real system, with a high degree of compatibility with the RBAC model.

However, there are limitations of the RBAC model which should be taken into account before it is “fixed in stone” by standardisation.

- The potential conflict of the inheritance mechanism with control principles, particularly Separation of Duties, creates obstacles to the use of role hierarchies. One approach to solving this is the incorporation of constraints into the RBAC model. Work has been done on this by [14], [15], [16] among others, but no common agreement has been achieved yet.
- Another aspect of the RBAC model, which is simple to address, is the need to introduce a mechanism for defining groups of users to which roles can be assigned. The use of a general mechanism such as policy domains [13] needs investigation.

5. Acknowledgements

We gratefully acknowledge the help of Mr. Horlacher, Mr. Petersen-Nickelsen and staff at Dresdner Bank Frankfurt, in making this case study possible. The UK Engineering and Physical Sciences Research Council have provided funding for this work under EPSRC Scholarship number 99311141.

6. REFERENCES

[1] Sandhu R., D. Ferraiolo, and R. Kuhn, "The NIST Model for Role-based Access Control: Towards a Unified Standard." presented at 5th ACM RBAC, Berlin, Germany, 2000.

[2] Sandhu R., E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models." *IEEE Computer*, vol. 29, pp. 38-47, 1996.

[3] Nyanchama M. and S. Osborn, "The role graph model and conflict of interest." *Transactions on Information Systems Security*, vol. 2, pp. Pages 3 - 33, 1999.

[4] Lupu E., D. Marriott, M. Sloman, and N. Yialelis, "A policy based role framework for access control." presented at Proceedings of the first ACM Workshop on Role-based access control, 1996.

[5] Epstein P., Sandhu, R., "Towards a UML based approach to role engineering." presented at 4th ACM Workshop on Role-based Access Control, Fairfax, US, 1999.

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

[7] Awischus R., "Role based access control with the security administration manager (SAM)." presented at Proceedings of the second ACM workshop on Role-based access control, 1997.

[8] Sandhu R., Bhamidipadi, V., "An Oracle Implementation of the PRA97 Model for Permission-Role Assignment." presented at ACM RBAC, 1998.

[9] Sandhu R., and Epstein, J., "NetWare 4 as an Example of Role-based access control." presented at Proceedings of the first ACM Workshop on Role-based access control, 1996.

[10] 5th ACM Workshop on Role-based Access Control, Berlin, 2000.

[11] Moffett J., "Control Principles and Role Hierarchies." presented at 3rd ACM Workshop on Role Based Access Control (RBAC), George Mason University, Fairfax, VA, 1998.

[12] Schaad A. and J. Moffett, "The Incorporation of Control Principles into Access Control Policies (Extended Abstract)." presented at Hewlett Packard Policy Workshop, Bristol, 2001.

[13] Sloman M. S. and K. P. Twidle, "Domains: A Framework for Structuring Management Policy." in *Network and Distributed Systems Management*, M. S. Sloman, Ed.: Addison Wesley, 1994, pp. 433-453.

[14] Ahn G. and R. Sandhu, "The RSL99 language for role-based separation of duty constraints." presented at Proceedings of the fourth ACM workshop on role-based access control, 1999.

[15] Jaeger T., "On the increasing importance of constraints." presented at Fourth ACM workshop on role-based access control, Fairfax, VA USA, 1999.

[16] Tidswell J. and T. Jaeger, "Integrated constraints and inheritance in DTAC." presented at Fifth ACM workshop on Role-based access control, Berlin, Germany, 2000.