Role Mining - Revealing Business Roles for Security Administration using Data Mining Technology

Martin Kuhlmann
Dalia Shohat
SYSTOR Security Solutions GmbH
Hermann-Heinrich-Gossen-Strasse 3
D 50858 Cologne
[martin.kuhlmann|dalia.shohat]
@systorsecurity.com

Gerhard Schimpf
SMF TEAM IT-Security Consulting
Am Waldweg 23
D 75173 Pforzheim
Gerhard.Schimpf@smfteam.de

ABSTRACT

In this paper we describe the work devising a new technique for role-finding to implement Role-Based Security Administration. Our results stem from industrial projects, where large-scale customers wanted to migrate to Role-Based Access Control (RBAC) based on already existing access rights patterns in their production IT-systems.

The core of this paper creates a link between the use of well established data mining technology and RBAC. We present a process for detecting patterns in a data base of access rights and for deriving enterprise roles from these patterns. Moreover, a tool (the SAM Role Miner) is described. The result allows an organized migration process to RBAC with the goal of building a single point of administration and control, using a cross-platform administration tool.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection – Access Controls; H.2.0 [**Information Systems**]: General – Security, Integrity, and Protection; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection.

General Terms

Management, Security.

Keywords

Role-Based Access Control, Enterprise Systems Management, Provisioning, Identity Management, Data Mining, Migration, Role Engineering, Security Administration, Security Data Models, Security Management, Single Point of Administration and Control.

SACMAT'03, June 1-4, 2003, Como, Italy.
Copyright 2003 ACM 1-58113-681-1/03/0006…$5.00.

1. Introduction

For several years now, many large-scale enterprises have been realizing savings through a reduction of the overall workload and through quality improvements in their enterprise wide identity-based security administration. The notions of provisioning and identity management stand for automation, productivity increase and security policy compliance. Enterprises have demonstrated how to cope with the complexity and dynamics of granting access rights to huge user populations across diverse computing platforms accessing a multitude of legacy and new applications. A key factor is the availability of commercial software for enterprise wide identity management to build a uniform single point of administration and control [1] [2].

A case study from a real-life organization may be helpful to clarify the situation. This large bank has over 45.000 employees working at headquarters, several associated companies and 1.4000 branch offices serving 5 million customers. They are running over 40 highly diverse productive systems with 65.000 User-Ids and 47.000 user groups. In a first step, a cross-platform administration tool was implemented as a single point of administration and control. Over time this bank has migrated to RBAC using different methods to define roles. Today a user obtains role assignments based on attributes, such as function key, company and department. To minimize manual intervention, the role assignment process has been integrated and automated using an import system from their HR database into their central repository.

Their benefits are:

- changes in user authorizations may take place at short notice,
- a high degree of consistency in terms of cross platform access rights,
- gains in end-user productivity and reduced error rate in security administration.

When the first tools for cross-platform security administration appeared on the market around the middle of the nineties, it became apparent that the abstraction of the access control concept using role semantics was necessary to exploit the full potential of these administration tools [5]. At the same time, research provided the first formal role-models [6] [7]. A method however which was both scalable and could be validated to engineer roles (i.e. define and maintain) within a large enterprise was not yet in sight. Now that research and practical issues on RBAC have converged, practitioners and researches are closely working together on the most important topics [8] [9] [10] [11] [14].

In this paper we present a new methodology for role-engineering. We describe a way to find roles in large enterprises using data mining technology to extract knowledge contained in existing access rights. This work was done to support projects migrating to RBAC within the framework of the Security Administration Manager (SAM) [2] [20] software product.

1.1 Motivation

Looking at the essential IT security administration processes, access control and granting access rights to users are of central importance. Managing authorizations is an onerous process.

In this context, the role-engineering issue is gaining more and more importance because it has become clear that role-based access control is the basis for what is today called "provisioning", i.e., the automation of identity-based security administration. The cost reduction and productivity potentials of automation and of more efficient manual administration are a high motivation for organizations to implement RBAC.

However, organizations have been reluctant to move to RBAC because of the envisioned high cost for role-engineering. It is therefore crucial to look for systematic and tool-supported ways for a facilitation of the role-engineering process. This is expected to lower the entry barrier to RBAC considerably.

To start off role engineering with formal business process modeling (“top-down approach”) would be desirable [5] [9]. However, in a complicated production environment with dozens of security systems, tens of thousands of user definitions, and millions of authorizations, this is a difficult task. And organizations often argue that any business oriented analysis of roles cannot ignore that, even without knowing the concept of RBAC, the business has been running for quite a while and that access control so far has been more or less effective. Access controls are in place and cannot be ignored for the RBAC concept. A pure approach of role generation "from scratch" is therefore in most cases impracticable.

So we concentrated on a way to analyze what we typically find in a production environment. The new approach taken here is feasible because of a SAM function, which initially loads all enterprise security data in a consolidated form to one access rights database. After this initial load a comprehensive meta data structure is available for analysis in the SAM repository, a large relational database. We may assume, that by data mining (i.e. knowledge discovery in the SAM database) we will be able to discover knowledge about roles or other access control patterns inherent to the business.

2. Role Engineering

2.1 RBAC

Already in the 1st ACM Workshop on RBAC in 1995, Edward Coyne [12] pointed out that role-engineering is an essential task for practical implementation of RBAC. His observations concerning role-engineering are still valid today without any restriction: "Definition of the roles with their assigned permissions must be accomplished before all the benefits of RBAC can be realized. The goal is to define a set of roles that is complete, correct and efficient."

Since then, practical implementations of RBAC in large corporations and institutions revealed that there are even more requirements to role-engineering than the ones identified by Coyne. There are requirements on the process of role-engineering as well as on the resulting roles themselves. Taking previous research and practical work into account [9] [12] [13] [14] [15], we were looking for a solution for RBAC in a SAM environment. SAM is positioned between two architectural layers and designed to link different views: the business layer where policies and business processes are in place (on top of Figure 1) and the access control systems layer (on the bottom of Figure 1). This leads basically to two approaches for role engineering: top down, starting from the business and policy side or bottom up, starting from the access control systems side.

The top-down approach starts from the business process oriented role descriptions. Very often, there is no policy at all, so we have to start from scratch by defining all privileges that a particular job function must have. A role for this job function is created, holding all relevant privileges.

In contrast, the bottom-up approach starts from the existing definitions for a known job function. An ID for such a job function is searched for privileges which are as close as possible to a desirable standard. A role is now created after this ID.

We have used both approaches for SAM, with the majority of the SAM customers preferring the bottom-up method, or a mixture of both. However the existing security definitions in grown production systems are almost always to some degree inconsistent and wrong. A simple comparison of users that are supposed to have the same role will show sometimes considerable differences and will be of little help in the definition of the role. It was therefore apparent to study modern clustering techniques as used in data mining to assist in the analysis and migration phases of a RBAC project.

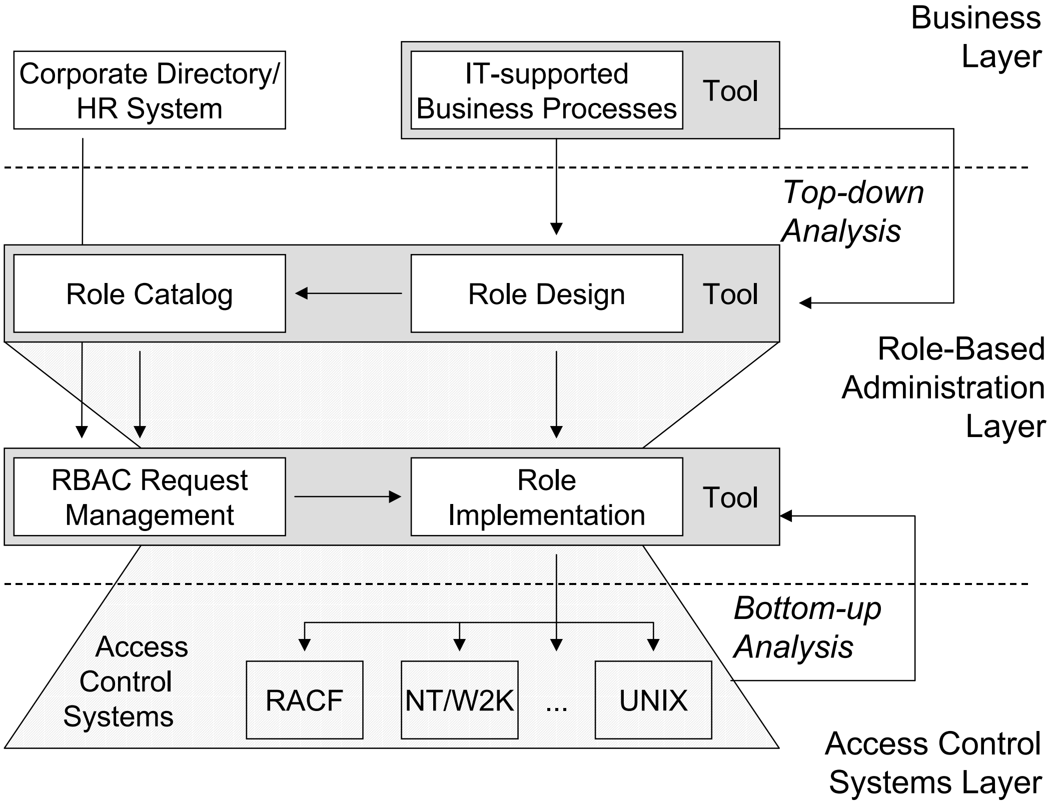

Figure 1 System Architecture [22]

2.2 Supported RBAC Model

The RBAC model guiding work around SAM is the ERBAC (Enterprise Role-Based Access Control Model) recently presented by Kern [20] and previously mentioned by Kern, Kuhlmann, Schaad and Moffett [14] , which is based on RBAC96 and on the NIST role standard draft. The idea of ERBAC is to enhance RBAC by the definition of enterprise roles, spanning different IT systems. ERBAC supports role hierarchy, but does not feature dynamic control of the actual user session, because the idea of enterprise-wide administration does not allow for a specific session concern. Static separation of duties (SOD) and other constraints can be checked according to predefined rules during role assignment to user and during role maintenance.

The enhanced ERBAC suggested in [20] addresses also the problem of role-housekeeping . The ERBAC model is enhanced with features that reduce the complexity and amount of roles. By parameterization of the roles on some of the defining attributes (like location or amount-limit), it is possible to create standardized roles related to business functions, letting the different variable values control access scopes as well as (e.g. location-dependant) resource assignments on platform/agent level.

Our bottom-up role finding process is based on the ERBAC model. For clarity reasons, we restrict ourselves to a simplified ERBAC without role hierarchy, constraints and SOD properties. We describe how to get a flat and 'raw' set of roles that reflect the truth as known by active access control systems combined with business (organizational and functional) information. The role finding technique that we describe may however be enhanced in a further step in order to refine the roles with static constraints and SOD, hierarchy definitions and parameterization (see chapter 7).

3. Applicable concepts of Knowledge Discovery and Data Mining

3.1 Introduction

Data mining tools are used widely to solve real-world problems in engineering, science and business.

The application of database technology, modelling techniques and statistical analysis to learn hidden facts about the mass of data available in an enterprise database is a natural development in the axis

Data -->Information --> Knowledge.

In commercial applications, developing such tools has been in the foreground of the data-warehouse packages in the last decade. The process of extracting previously unknown information from existing data sources is called ‘data mining’ or ‘knowledge discovery’.

It is only natural that when we were faced with the need to find some meaning (i.e., roles) in the cross-platform access control data we resorted to these tools. The fact that in our context access control data from all relevant systems are stored in the central SAM repository facilitates the mining process considerably.

The methods used in data mining are of two main types:

Descriptive methods and predictive methods.

- Predictive methods learn from the data the instances in which a specific value is attained and define a model. On the basis of this model they try to predict this value for other instances. For example, a model of a reliable loan recipient could be learned from the bank data, and used to predict whether a new applicant is reliable or not. Other fields of implementation could be in minimising risk in stocks portfolios or in new drugs developments . The methods belonging to this group are classification and regression.
- Descriptive methods describe hidden facts about the massive data available. Their use is in scientific applications (astronomy, genes research, meteorology) where recorded observations have to be searched for meaning and patterns, as well as commercial applications where consumer buying habits have to be understood. Clustering, associations and pattern discovery are classified as descriptive methods.

The role engineering processes applied until now in real life situations and in theory assume selection of single 'role-representative' users whose access rights are viewed as templates for the role. This approach relies rather on personal knowledge of administrators and not on the massive data stored in the enterprise database.

What we suggest is to create a more reliable insight to what we assume to be the existing underlying security roles in the corporate access control data by applying descriptive data mining methods.

3.2 Data Mining Techniques used

We use the IBM Intelligent Miner for Data [21] as the mining engine of the security data. The tool enables us to select the data portions to be analysed (both data subsets and attributes to be considered), recognise invalid data, iterate the mining processes with different data subsets or statistical threshold selections until an acceptable result (in terms of granularity, cohesion and meaning) is reached, and finally to produce role definitions based on the attained grouping.

The following data mining methods are used for our role finding process :

Association – The association algorithm creates rules that describe how often events occur together. The result has to parts :

- Item sets – sets of items that appear together (in a transaction) with their frequency (support)
- Rules – rules that can be deduced from the item sets with statistical measures such as confidence.

Clustering **–** Segmenting the data into disjoint groups (clusters) of records having similar values in certain attributes (active fields).

For our role engineering approach, we choose the 'demographic clustering' feature of IBM's Intelligent Miner for Data which has the following properties :

1. Hierarchical clustering, in an iterative refining process.
2. Performs partitioning clustering when the maximal number of clusters is given.
3. Uses the Condorcet criterion to evaluate the clusters' quality (inter-cluster-separability and intra-cluster-homogeneity).
4. Deals with quantitative as well as discrete variables. Can treat numerical data also as discrete variables.
5. Can perform weighting of attributes so that the clustering is affected differently by similarity or difference of a specific attribute.

A detailed explanation of these methods leads beyond the scope of this paper and can be found in [19].

By using the IBM Intelligent Miner for Data as our data mining engine, the enterprise consolidated organisational and operational security data is scrutinized and analyzed to the single attribute level. This process results in statistical and semantic information which is key for role finding.

4. Role Mining

Companies which plan to go for RBAC are usually in the situation of having a collection of different legacy and standard security systems on different platforms providing conventional identity-based access control (IBAC).

By „role mining“ we understand a method of generating roles from the access control information (ACI) of this collection of systems. "Role Miner" is a commercial software product that supports the mining process.

Our approach starts with collecting and consolidating the HR, ACI and other security related user information within a global database of access rights definitions. We apply clustering and association data mining techniques to the data. The structural and statistical information we receive is used to derive roles.

The result is a role schema (see Figure 2). We distinguish two different categories of roles: an organizational role is considered to be the basic role of a user in the organization and comprises all basic accounts and access rights in the different systems. Functional roles are supposed to describe further access rights of the user related to additional functions or tasks.

In this context a role is a non-target-system-specific entity and is usually not represented by an entity in the target systems (like for example the group entity). However, connections of a role with target systems represent the existence of user accounts, and connections with target system groups point to resources. Direct connections with resources are avoided if the target system allows for group authorizations (which is usually the case).

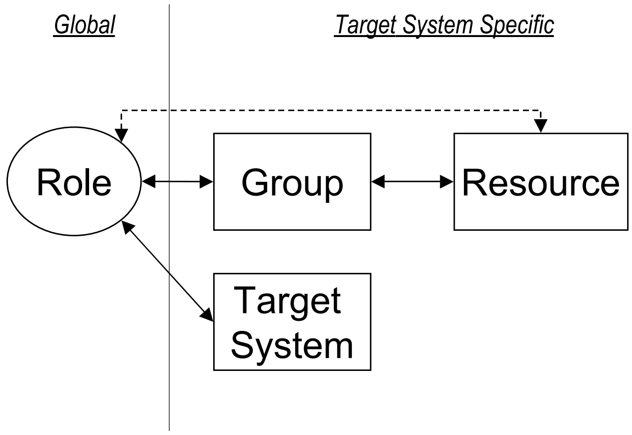

Figure 2 Role Schema

A role also bears values for user-related data (attributes) which are common to all users assigned to the role (as for example the same value for the attribute `location` if all assigned users work in the same location). This is a template-like aspect of roles; the values being applied to all users with the role assignment and controlled by the role.

As software tools for the support of the Role Mining approach we use

- a cross-platform security management system (Security Administration Manager (SAM), Systor) for the collection and consolidation of the ACI as well as for the implementation of the resulting roles.
- a data mining software (Intelligent Data Miner, IBM).
- a data preparation and role generator software (part of SAM Role Miner).

A main benefit of the Role Mining approach is the possibility to profit from data already available in corporate repositories rather than to build up a role repository manually. Unlike most role engineering approaches, we are able to check our assumed role description statistically against both corporate data and access control data, and we are able to modify the role description accordingly.

Furthermore, using software support for the whole role finding process, each step is easily repeatable with modifications to parameters and basic data. Thus it is possible to make role finding an iterative process and profit from the learning curve. The use of a cross-platform security management system allows for automated data collection and role implementation.

4.1 Role Mining database

For Role Mining we first choose a fixed set of users and a collection of security systems ("target systems") as an environment for which an RBAC scheme is to be set up. We assume that each user has at most one account on each target system (which is true for most productive environments in large companies).

The Role Mining is based on the following kinds of user-related data:

- For each target system, the information if a user has got an account on the system or not. In other words, we take the list of systems the user is connected with.
- The user-related system-specific data stored within each target system (information like dial-in access specification, login procedures, working times, home directory location, etc.).
- Assignment to groups or roles in the target systems.
- Direct authorizations for resources in the target systems.
- Global (i.e., not target system specific) user information. This comprises individual descriptors like user language as well as security-relevant organizational data like location names or hierarchy positions.

5. The Role Mining Process

5.1 Characteristics of Resulting Roles

The Role Mining process yields two types of roles: organizational roles and functional roles.

- An organizational role is the basic role of a user in the organization. A user has exactly one organizational role. Through this role he acquires all basic accounts and access rights in the different systems. The organizational role contains all global and system specific information which is common to the users having this role. It is connected with all target systems in which the users have an account. Furthermore, it is connected to groups in the different target systems. As far as groups are supported by the target systems, direct connections of roles with resources are avoided.
  An organizational role is described by one or more attributes. Examples are `org_unit` or the combination `(job, location)`.
- Functional roles are supposed to describe further access rights of the user related to additional functions or tasks, such as substitutes or projects. Any number of functional roles can be assigned to a user. Functional roles are connected with target systems and groups like organizational roles. To avoid ambiguities, they do not contain global and further system specific information.

  A functional role might be described by attributes or combinations of attributes like `project_name` or `(hierarchy_level, authority_level, department)`.

5.2 Role Mining Steps

The steps that are performed in the role mining process are a refinement of the process of data mining as described for example in [19]. After each step, it is possible to return to one of the preceding steps if the results are not satisfying. This ends up in an iterative role finding process.

1. Choice and provision of information sources

We already described the scope of data for the mining process. Of course, from the available data a subset has to be selected which is most promising and on-target to yield suitable information for role creation. This applies to the subset of users as well as to the semantics of the attributes (data fields). The chosen information sources must be up to date, correct and as stable as possible (i.e.: not subject to frequent change which would result in a frequent role redesign). The possible number of field values for each attribute should be such that a clustering of the desired granularity seems to be achievable.

At this stage it might also be necessary to perform a first cleaning of data in order to eliminate or correct information that is obviously or known as incorrect.

2. Data preparation

After identification and first cleaning, the data is collected from the various locations and transformed into a format in which it can be processed by the data mining software.

3. Exploration and Learning

The exploration and learning phase is the crucial phase in the Role Mining process. It will provide a "feeling" for the data contents and the expected role scheme.

The demographic clustering and association techniques are applied for the following purpose:

- Clustering is performed on the user attributes to receive organizational roles
- Association is performed to receive group connections and authorizations for organizational roles and to set up functional roles with their group connections and authorizations.

If the attributes which describe the roles are not yet determined, the first ("free") data clustering rounds serve to find suitable attributes or to make their choice plausible. As soon as these attributes are clear, a special weight can be put on them in a further mining run to form clusters or associations accordingly.

The results of this phase are suitable attribute sets for the unique representation of organizational and functional roles and suitable data mining parameters like active clustering fields, thresholds, etc. (see [21]).

4. Mining

In the mining step the final data mining is performed. The resulting statistical reports are thoroughly examined. Parameters are fine-tuned if necessary. The result of this step is the basis for the role creation.

5. Role creation

The statistical reports of the final data mining run are used to automatically derive organizational and functional roles, their connections to target systems, their group memberships and authorizations. According to the relative frequency of attribute values in each cluster or association, a decision is made if global or target-system-specific attributes will assume a specific value for the role (and thus for all potential users having this role).

As our policy is to avoid direct authorizations of roles to resources, we create a group in each target system to which the role authorizations are tied (see point 6 below), instead of tying them directly to the role. The groups are then connected with the role.

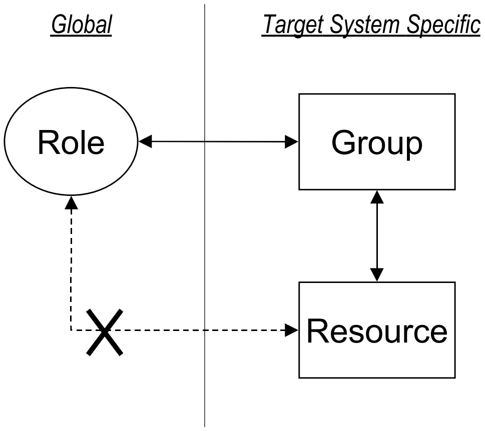

Figure 3 Role-Group Connection

6. Check, approval and implementation of resulting roles

The resulting roles have to be checked for plausibility and correctness. In most companies the involvement of the IT auditing department will be required. After approval, the roles can be implemented using the cross-platform security management system.

7. User Assignment

The last step in the Role Mining process is the assignment of roles to users. Having done this, redundant and incorrect access rights as well as individual access rights may be deleted.

5.3 Rollout of RBAC Transformation

As the Role Mining process is easily repeatable, it makes sense to roll out the RBAC transformation in the company step by step.

Different parts of a large company may have very similar but also very different security management policies. The Role Mining processes for two branches are usually very similar, but for the central administration department and the net of branches they can be completely different in terms of attributes for role description, etc.

6. Case Study and quantitative Data

The above described role mining concept has been implemented as a commercial software product called SAM Role Miner.

The following two case studies are based on initial customer reactions and demonstrate the power and usefulness of this tool.

Case A: Productivity Gains

This company has the following business profile

- one of the top five insurance companies in Germany
- a global player with offices in North America, Europe and Asia Pacific (14 locations in Germany)
- 50 Million clients world wide (4 Million clients in Germany)
- 140.000 employees world wide (18.000 employees in Germany).

Based on an existing SAM [2] installation with production data at the corporate headquarter in Germany, SAM Role Miner was pilot-tested. For their 18.000 User-Ids, 930 SAM models where in existence. These models were previously developed in a long and cumbersome manual process by their security department.

SAM Role Miner was then used to analyze production data stored in the SAM database. Within several hours, all 930 roles could be determined. With SAM Role Miner, the role-engineering time improved by two orders of magnitude, i.e. from several months to several hours.

Case B: Business Benefits

Profile:

- This company has 10.000 users and an estimated number of 400 roles
- it has spent $5.000 per role for manual role creation
- the turnover time for roles is estimated to be 3 years
- the maintenance cost (i.e. redesign) per role is $2.500.

Assuming a 10% growth of staff and 5% growth of the number of roles per year, this results in the following cost scenario:

- the cost for the initial role engineering is about 2m$,
- the yearly cost for new roles creation and role maintenance is $433.000.

Using role mining, we assume a saving potential of about 60% for initial role creation, and a cost reduction of about 50% for maintenance.

- This yields a 1.2m$ saving in the setup phase and another $226.500 annual cost reduction for role maintenance.

This estimated cost reduction takes into account that the role mining process considerably eases the task of proposing to the role engineer which access rights bundle constitutes a role. The possibility to easily do iterations yields additional effort reduction.

However, our current process still needs major manual interventions to make the discovered clusters plausible from a business perspective and adjust them properly. An 80-90% automation of role finding is thus unrealistic with this approach. It would therefore be interesting to investigate if the marriage of the role mining process with a top down business process analysis approach can increase the degree of automation. This seems to be plausible for organizations which have good business process descriptions in place.

7. Conclusions and further work

After the successful foundation work to establish a variety of general RBAC models, it seems promising to search for systematic automated approaches to role engineering. This is for two reasons: Industrial projects are looking for ways to implement RBAC in large companies and are struggling hard with the transformation process. On the other hand, role engineering techniques will hopefully be of help in the effort of standardizing RBAC.

In this context we have presented a method for finding roles from an existing database of cross-platform access rights. This method is a “bottom-up” approach in the sense that IBAC access right definitions are “lifted” to a role schema. The approach is an alternative in contrast to the “top-down” approach presented in [9], where roles are derived from a business process analysis. Depending on the actual customer situation, both systems may have strengths and weaknesses. They may however complement each other. In the Role Mining approach we may have the difficulty of appropriating a semantic business meaning to the clusters that are found. The business process approach, on the other hand, leaves us with the problem of securing a smooth transition to RBAC in a situation where already established access rights are in place as IBAC.

Thus, it looks reasonable to investigate possibilities how to integrate these two approaches.

Another starting-point for further work is the extension of the method presented in this paper to the full ERBAC model:

- The presented role mining method yields a flat role structure. From our point of view, it would be most interesting to enhance our technique to role hierarchies. Once the flat roles scheme is loaded into SAM , which supports an hierarchy of roles, one could think of using mining (association) to find higher roles. This would of course happen before the basic roles are deployed and made operational (assigned to users).
- It would be interesting to use data mining techniques for aggregating existing permissions not only with user/role-focus as we describe, but first with application/permission focus or in the "privilege plane". That is, creating aggregates of authorized objects/operations as applications or privileges [18][15][16]. This could shed a light on SOD and constraints issue (see 2.2 above).
- The parametrization of roles would be an interesting feature to investigate in the role mining context. The following is a simplified illustration of parameterization in a bank:

  As a result from the mining process, one finds that the triple [Job, location, max_pay_off_amount] describes a role from a business point of view, and the role miner generates one role for each such combination.

  For example, the following role has been found with their permissions by the role miner:

  `'teller,munich1,30000'` which gives access rights to bank applications A,B,C, the authorized amount for paying off cash being 30,000 Euro.

  One finds, however, from the existing access rights data, that application access additionally depends on a server id and that this server id is different for different tellers. Now, instead of enlarging the role identifying attribute triple by a fourth attribute, it is advisable to use the server id as a user attribute and making it a variable role parameter.

As the role mining is a kind of automated reengineering and the business policies are not directly emergent from the data, hierarchies, SOD and constraints are not immediately obvious after the fact. We feel that in this case, the engineering process must be led by business policies and assumptions, assisted by data mining techniques for validation and risk assessment.

8. REFERENCES

[1] G. Schimpf, "Security Administration and Control of Corporate-Wide Diverse Systems," in *ACM SIGSAC Review*, vol. 15(1), 1997.

[2] "Security Administration Manager (SAM), Release 2.4. Concepts and Facilities," Systor GmbH & Co. KG, Köln, Germany (1999).

[3] B. J. Biddle and E. J. Thomas, "Role Theory: Concepts and Research". New York: Robert E. Krieger Publishing Company, 1979.

[4] D. F. Ferraiolo and R. D. Kuhn, "Role-Based Access Controls," presented at 15th NIST-NCSC National Computer Security Conference, Baltimore, MD, USA, 1992.

[5] R. Awischus, "Role-Based Access Control with the Security Administration Manager (SAM)", presented at 2nd ACM Workshop on Role-Based Access Control, Fairfax, Virginia, USA, 1997.

[6] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based Access Control Models", IEEE Computer, vol. 29(2), 1996.

[7] R. S. Sandhu, V. Bhamidipati, and Q. Munawer, "The ARBAC97 model for role-based administration of roles", *ACM Transactions on Information and System Security*, Vol. 1 (No.2 Feb.), 1999.

[8] A. Mönkeberg and R. Rakete, "Three for One: Role-based Access Control in Rapidly Changing Heterogeneous Environments", presented at 5th ACM Workshop on Role-Based Access Control, Berlin, Germany, 2000.

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

[10] R. S. Sandhu, "Engineering Authority and Trust in Cyberspace: The OM-AM and RBAC Way", presented at 5th ACM Workshop on Role-Based Access Control, Berlin, Germany, 2000.

[11] R. S. Sandhu, D. F. Ferraiolo, and R. D. Kuhn, "The NIST Model for Role-Based Access Control: Towards A Unified Standard", presented at 5th ACM Workshop on Role-Based Access Control, Berlin, Germany, 2000.

[12] E. J. Coyne, "Role-Engineering", presented at 1st ACM Workshop on Role-Based Access Control, Gaithersburg, MD, USA, 1995.

[13] E. B. Fernandez and J. C. Hawkins, "Determining Role Rights from Use Cases", presented at 2nd Workshop on Role-Based Access Control, Fairfax, VA, USA, 1997.

[14] A. Kern and M. Kuhlmann, A.Schaad and J. Moffett, "Observations on the Role Life-Cycle in the Context of Enterprise Security Management", presented at SACMAT 2002, Monterey, CA, USA.

[15] D.F. Ferraiolo, "An Argument for Role-Based Access Control", presented at SACMAT 2001, Chantilly, VA, USA.

[16] P.A. Epstein, "Engineering of Role/Permission Assignments" – doctoral dissertation 2002, GMU Fairfax, VA, USA.

[17] T. Jaeger, "On the Increasing Importance of Constraints", presented at 4th ACM Workshop on Role-Based Access Control, Fairfax, VA, USA, 1999.

[18] M. Nyanchama and S. Osborn, "The Role Graph Model and Conflict of Interest", *ACM Transactions on Information and System Security*, Vol. 2 (No. 1, Febr), 1999.

[19] J. Grabmeyer and A. Rudolph, "Techniques of Cluster Algorithms in Data Mining", IBM Informationssysteme GmbH, December 10, 1998.

[20] A. Kern, "Advanced Features for Enterprise-Wide Role-based Access Control", 18th Annual Computer Security Applications Conference, Las Vegas, NV, December 2002.

[21] IBM Intelligent Miner for Data, User Manual.

[22] H. Röckle and G. Schimpf, "Rollen-Engineering im IT-Berechtigungsmanagement" KES Zeitschrift für Kommunikations- und EDV Sicherheit 5/00, 2000.