PROROLE: A PROCESS-ORIENTED LIFECYCLE MODEL FOR ROLE SYSTEMS

Leveraging Identity Management and Guiding Role Projects

Fuchs, Ludwig, Universität Regensburg, Universitätsstraße 31, 93053 Regensburg, Ludwig.Fuchs@wiwi.uni-regensburg.de

Pernul, Günther, Universität Regensburg, Universitätsstraße 31, 93053 Regensburg, Guenther.Pernul@wiwi.uni-regensburg.de

Abstract

The complexity of modern organisations' IT-Landscapes has grown dramatically over the last decades. As a result, user handling has reached a degree of complexity where no single administrator can give satisfactory evidence about which users have access to certain information and who has granted those permissions to them. Compliance issues put even more pressure on the responsible managers. In-house Identity Management (IdM) has undoubtedly cashed in on that development as companies are forced to launch projects to regain control over what their users are doing within the IT-Systems. Identity Management itself, however, is only the starting point for getting compliant. The introduction of roles leverages IdM to the next level by simplifying the connections between users and resources and strengthening the overall security level. Therefore many companies initiate role projects in order to reorganise their access structures. Lacking experience and know-how, they are looking for a generic approach structuring the tasks within role projects. This paper presents proROLE, to our knowledge the first comprehensive process-oriented lifecycle model for role systems. It helps companies understand the issues surrounding roles and the steps they have to take in order to create and maintain a working role system.

Keywords: Compliance, Identity Management, Information Security, Lifecycle Model, Roles

1 INTRODUCTION AND MOTIVATION

Big companies have to manage a large number of identities within their IT-Systems. Employees' identities are managed as user accounts and necessary for administrating users, allocating and revoking resources, and auditing, restricting, and controlling their actions within the IT-Systems. As a result of manual account management and inadequately enforced security policies users accumulate a number of excessive rights within the organisations' IT-Systems over time, violating the principle of the least privilege (Ferraiolo et al 2007). This situation bears significant risks and results in a so called identity chaos companies have to face nowadays. Projects like (Larsson 2005) and popular studies (Dhillon 2001) show that major security risks arise because of employees gaining unauthorised access to resources as a result of non-standardised application-specific rights- and user management. According to Stanton et al (2005) in respect to confidentiality and integrity, the users themselves, rather than popular viruses, phishing, or pharming attacks represent the main threat.

Implementing a basic Identity Management Infrastructure (IdMI) as presented in Fuchs et al (2007), however, is only the starting point for getting compliant. Even though it has become a popular means to deal with the identity chaos, companies realise that it takes more than a basic IdM solution in order to regain the control over the information flows and access rights. When it comes to the next level of Identity Management, practical experience from our engagement in different projects has shown that the usage of roles (Sandhu et al 1996) bears potential for significant improvements. The theoretical work in this area, amongst others, covers numerous Role Models, Administration Models, and Role Development Methodologies.

However, what companies are looking for is a comprehensive methodology that structures the introduction of roles, providing assistance in putting together its different technical and organisational pieces. The lack of know-how as well as the level of complexity of the internal rights structures and the organisation-wide and continuous character of the projects easily results in a negative attitude of business sponsors towards implementing organisational roles. Research still fails to provide a model that spans the complete lifecycle of a role system from the very first definitions and project plans up to the daily role management. This paper presents proROLE, an UML-based lifecycle model for a role system, closing the gap between theoretical academic work and industries' need for guided assistance in implementation projects. A simplified model overview as described in section 3 can be used at the beginning to discover the main building blocks of a role system before dealing with the process-oriented project details according to the UML activity diagram of proROLE. Acting as a guideline, proROLE supports enterprises in structuring their role projects, producing the adequate documentation and finally succeeding in managing roles and keeping them up-to-date over time.

This paper is structured as follows: In section 2, related work is presented, existing lifecycle models are examined, and their shortcomings analysed. Section 3, subsequently, introduces the main layers of proROLE while section 4 goes into detail explaining parts of the method using UML activity diagrams. Finally, conclusions and future work are given.

2 RELATED WORK AND BASIC DEFINITIONS

2.1 Identity Management

Over the last years IdM has grown up and established itself as a core component of enterprise security management. It deals with the storage, administration, and usage of digital identities during their lifecycle (Fuchs et al 2007). In-house Identity Management is strongly connected with one of the key requirements in open and closed networks: The secure and efficient administration of numerous personal attributes that make up digital identities. We define the term *identity* according to Pfitzmann et al (2000) as a subset of attributes or characteristics of an entity which make the entity, for example a person, uniquely identifiable within a set of entities. Every identity has to be created, maintained, and erased separately. The aforementioned identity chaos needs to be faced by implementing a centralised Identity Management Infrastructure as shown in Figure 1.

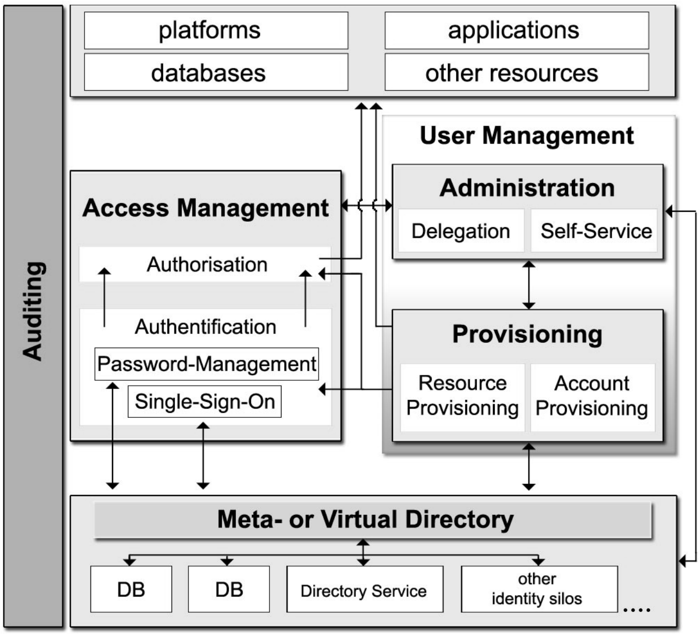


Figure 1: IdMI within an organisation (Fuchs et al 2007)

Such an IdMI represents the realisation of defined processes and policies using adequate technical measures. Its main building blocks are a Directory Service, a User Management, Access Management, and Auditing Module. Directory Services provide synchronised identity information that is facilitated by the other IdMI components. User Management e.g. deals with the provisioning of users, granting and revoking access to resources. When users logon to certain applications, Access Management controls the access to the requested resource while users' as well as administrators' activities are logged within the Auditing module. IdM duties cover rather simple tasks like automatic allocation and revocation of user resources, e.g. Microsoft Windows accounts. However, they also include sophisticated tasks like role management. Reducing security risks and getting compliant are the two major drivers of modern role-based Identity Management solutions. National and international regulations like Basel II (2006), the Sarbanes-Oxley Act (2002), and the EU Directive 95/46 (1995) together with internal guidelines and policies force enterprises to audit the actions within their systems, providing proof of evidence about who has access to vital information and who has granted those permissions to them. Additionally, the reduction of administration costs due to efficient user management as well as grown security awareness leverage the introduction of role-based IdM for the purpose of secure and standardised rights management.

2.2 Roles and basic definitions

Role-Based Access Control (RBAC) is a widely used access control mechanism in which roles act as an intermediary between users and permissions. Users are assigned to roles and roles are associated with permissions that determine what operations a user can perform on information objects acting as a role member. Additionally, various kinds of constraints, hierarchical structures, and other attributes of roles can be specified (Ferraiolo et al 2007). In the planning report 02-1 of the National Institute of Standards and Technology (NIST), Gallaher et al (2002) state the numerous advantages of roles for organisations. Besides a more secure and efficient user- and resource management, manual administration efforts are minimised and compliance issues addressed. A large body of work has been conducted in the area of roles for the last 15 years. Numerous Role- and corresponding Administration Models have evolved as a result of special industry needs. Possible fields of application range from application-specific access control and organisation-wide IdM up to Identity Federations and automated provisioning processes. These various different application areas, however, have resulted in a large number of research directions within the role community resulting in a fuzzy terminology. In order to avoid misunderstandings, the terms used in this paper are clarified in Table 1. Afterwards the paper will focus on existing lifecycle models for roles, their characteristics, and shortcomings.

Name	Definition
Role Concept	The Role Concept represents the basic, model independent, understanding of roles. Most commonly used are the rights-based Role Concept (where roles are related with access rights) and the task-based Role Concept (which connects roles with business tasks).
Role Model	The Role Model represents the theoretical framework consisting of different concepts related to the usage of roles and their interdependencies. Existing Role Models greatly differ in terms of complexity and functionality.
Administration Model	The Administration Model represents the theoretical framework for managing the entities included in the Role Model.
Role Development Methodology	The task of creating a role catalogue including all roles used within an organisation is referred to as role development. The Role Development Methodology can be split into data gathering/cleansing and the role definition process, which can follow a Role Mining, Role Engineering, or a hybrid approach.
Role Mining	Role Mining is the technique to define roles bottom-up according to the existing user rights within the organisation's IT-Systems.
Role Engineering	Role Engineering is the technique to define roles top-down based on input information from the business-level.

Table 1: Basic definitions

2.3 Existing role-lifecycle models

Even though a lot of work has gone on in the area of roles in general, only a few researchers have focussed on the process of role development and the lifecycle of role systems, even though it is essential to follow a clearly defined roadmap in order to successfully implement and maintain roles in an organisation. Questions relating to the lifecycle of roles have been touched on in literature several times, e.g. in Lupu (1998), who presented the lifecycle of roles in the context of a pattern-based state machine specification. However, regarding the focus of this paper we discuss the two existing significant organisation-wide approaches of Kern et al (2002) and Schimpf (2000) in the remainder of this chapter. Both approaches rely on techniques and phases used in modern software engineering methods which provide an abstract description of the structured and methodical development and modification process of a software artefact.

Kern et al (2002) observed the role-lifecycle in the context of Enterprise Security Management. Their very basic and generic lifecycle model for the introduction and usage of roles aims at providing a better understanding of the technical and organisational dimensions of roles in the context of the Enterprise-RBAC (ERBAC) Role Model according to Kern (2002).

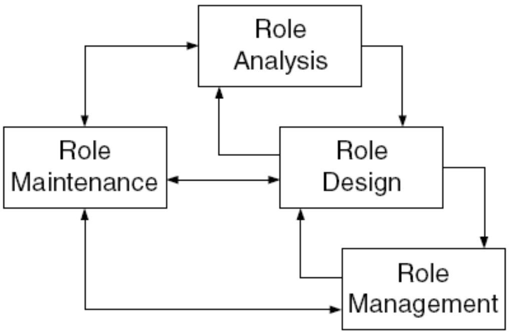


Figure 2: Lifecycle in the context of Enterprise Security Management (Kern et al 2002)

Figure 2 shows the four modules of their lifecycle in the context of Enterprise Security Management. The authors propose to use their model in an iterative and incremental way, not defining a strict structure or sequence of tasks that have to be completed. The *Role Analysis* represents the main task of identifying roles as they occur within the target domain in which the system will be placed. In contrast to the analysis, which is mostly based on acquiring knowledge about the current organisational context of a role, the *Role Design* phase tries to convert this knowledge into concepts that can be used by the later target system. *Role Management* includes the daily administration of users and the operation of finding changes in the Role Model. Closely related with these issues, *Role Maintenance* deals with major changes that affect the basic understanding of the Role Concept. These activities comprise changes in the mapping of organisational structures to roles and in the definition of user-role and role-permission relationships.

Contrary to Kern et al's approach which explicitly focuses on the lifecycle of roles, Schimpf (2000) deals with the lifecycle of roles in terms of organising Role Engineering projects. His main goal is to give an overview over the critical success factors for the development of roles on the basis of business processes. In order to deal with the complex nature of role projects he suggests following a formal and structured phase plan which is shown in Figure 3. Like Kern et al's approach, his model also draws on the structure of software engineering projects using the Analysis, Design, Build, and Maintain stages. The *Analysis* phase is needed to define the basic deliverables of the Role Engineering process and the aspired tool support. Companies' security policies have to be adjusted to the new world of RBAC and tools for the definition and later implementation of roles identified. During the *Design* stage, amongst others, the RBAC-Model is defined, prototypes constructed, the role-finding process organised, and the role catalogue built. The consecutive *Building* phase aims to transform the theoretical role catalogue into a productive system. Using a cross-platform administration tool like an IdM solution users and roles are handled. After the first introduction the *Maintenance* phase ensures that company-wide processes to cope with changes and their impact on roles are installed. This includes the addition, deletion, and modification of roles.

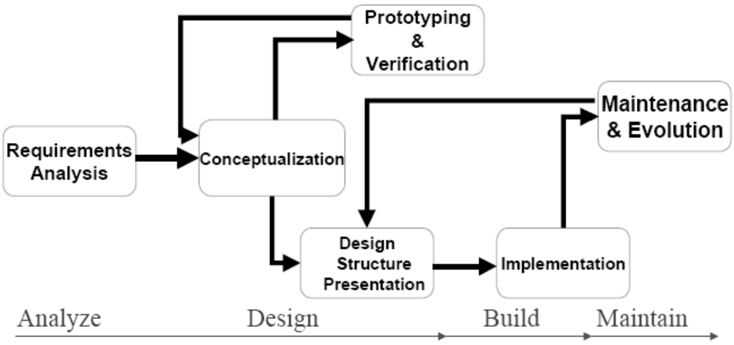


Figure 3: Lifecycle Management for Roles (Schimpf 2000)

2.4 Shortcoming of existing role-lifecycles models

After having presented the main concepts of the two existing role system lifecycles, this subchapter aims at highlighting their shortcomings. Keeping in mind that the most important aid companies are looking for is a structured and documented guideline for the implementation of a role system, both existing approaches fail in giving the required support.

The main goal of both lifecycle models is to point out important duties within role projects, essentially resulting in an unordered list of tasks. Although this helps companies pay attention to certain aspects of the role system, it cannot direct them through the overall process of role introduction. Even though they offer a list of to-do-tasks integrated in a high-level lifecycle model they don't focus on structure in a process-oriented manner. Their biggest shortcoming is that they itemise single duties without going into detail about their interdependencies or the order in which they have to be conducted. Our engagement in different IdM- and role projects has shown that business representatives as well as IT-staff are indeed looking for an easily understandable guideline that structures the tasks and communication within role projects in order to minimise possible mistakes, budget overruns, or schedule delays. Closely related to that issue is the insufficient level of detail of both existing approaches, not allowing an in-depth analysis of certain tasks. In addition to that, Kern et al don't mention any documentation duties during the role implementation projects at all while Schimpf only touches this issue on the brink. Insufficient documentation of results increases the complexity of communication between the different involved parties. Another shortcoming of both lifecycles is their limited view as they are used for implementing role structures according to the RBAC- (Sandhu et al 1996) or ERBAC-Model (Kern 2002). Even though most enterprises are likely going to start by implementing basic role structures, they might consider extending them with more sophisticated attributes. Using the existing lifecycle models they cannot in detail decide which role features they want to implement. In order to be able to do that, they need a generic Role Model including all possible concepts related to roles represented as interchangeable modules. Overcoming this shortcoming by proposing a comprehensive and modular Role Model that easily can be customised by organisations according to their specific needs is one of our current research focuses.

After discussing the existing lifecycle models and their shortcomings, the following chapters are going to present proROLE, our new comprehensive role system lifecycle. Chapter 3 gives a brief overview over the general structure, while chapter 4 is going to introduce the used notion as well as two central model components based on UML activity diagrams in more detail.

3 PROROLE: A PROCESS-ORIENTED LIFECYCLE MODEL FOR ROLE SYSTEMS

Companies are looking for a process-oriented and standardised way to introduce roles for the sake of getting compliant, reducing administration costs, leveraging the efficiency of user handling, and strengthening the overall IT security level. Focusing on the view of a role manager, proROLE takes these business needs into consideration and offers a technology-independent process for guiding role projects. Even though it also is loosely based on the main phases of software engineering, its most important goal is to guide responsible managers in a process-oriented manner by structuring tasks and their interdependencies during a role project. This way schedule and budget overruns as well as communication problems as a result of missing documentation can be avoided.

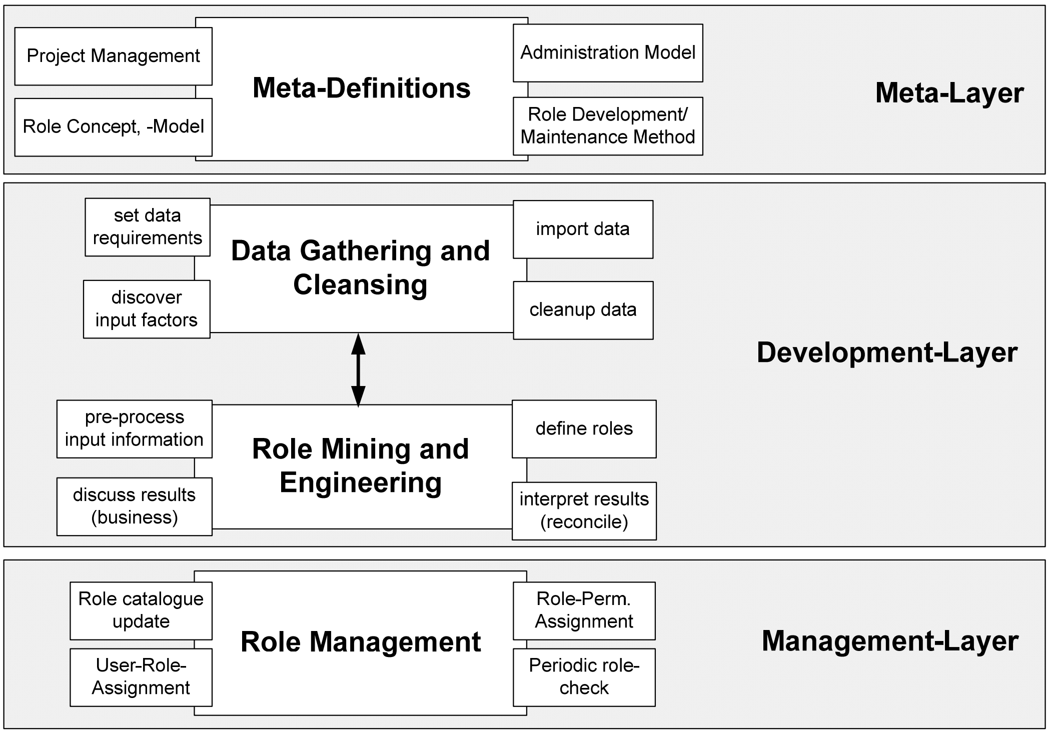


Figure 4: Layers of proROLE

Figure 4 gives a high-level overview over proROLE. It is divided into three stages representing the main layers of a role system, namely the Meta-, Development-, and Management-Layer. In order to introduce a role system, organisations have to successfully complete each of these stages consecutively, ending up in a loop of role administration at the Management-Layer. Within the layers several iterations and refinement procedures can take place. Above all, the actual definition of the role catalogue is coined by the mutual interaction between used input data, its cleansing, and the Role Development Methodology.

3.1 The Meta-Layer

Within the Meta-Layer the theoretical basis of the role system, namely the Role Model, the Administration Model, and Role Development Methodology is defined. The main outcome of this stage is the documented models that will be used in the consecutive lifecycle steps. We consider this collection of theoretical models as Meta-Definitions because they represent the abstract and rather stable foundation for the actual, organisation-specific implementation of the role system. As for any other large-scale IT initiative, strong project management skills, adequate sponsorship, and business-unit support are critical for the introduction of roles. Therefore different project management duties like setting up the project structure, recruiting appropriate team members, as well as locating affected security policies, systems, or infrastructural components are to be completed in addition to the role-centric tasks. As mentioned beforehand, ongoing research work deals with the development of a generic and customisable Role Model. Organisations will be able to configure the complexity of their Role Model by choosing modules they want to include in their IT-architecture using a categorisation of role attributes.

3.2 The Development-Layer

After the Meta-Definitions are finished, the role system enters the second stage of its lifecycle - the creation of roles in order to generate a comprehensive role catalogue. The main goal is to define stable as well as flexible role patterns for the specified application scenario based on given input information. The Development-Layer is divided into two structured task groups, namely the data gathering/cleansing and the actual role creation.

During the data gathering process needed input information is discovered within the enterprise. This includes IT-related knowledge like user information from the IdM-System as well as business-related documents like process models or job descriptions. After the sources for the needed information are located and quality requirements defined, data cleansing needs to take place in close collaboration with data owners before roles can be derived. Many Role Development Methodologies, specifically most of the top-down representatives like Neumann et al (2002) or Roeckle et al (2000), lack an appropriate cleansing of the used input information in terms of business-process-reengineering. Even Role Mining approaches like Vaidjy et al (2006) only mention the importance of data cleansing – without providing an approach for reducing inconsistencies within the input information. Without revising and correcting the input data, however, the results of the creation methodology are suboptimal as errors propagate during the development process of roles. This results in unused theoretical role patterns that are not accepted by the users as they don’t represent the actual situation within the organisation.

In the second phase of the Development-Layer role candidates are derived on the basis of cleansed and pre-processed input information. Additionally, existing application-specific or other available role definitions e.g. from previous projects can be used as input information. The methodology can rest on existing employee information as well as business-related documents or a combination of both. We propose to use Role Engineering with its information representing the business-view as well as Role Mining, representing the real-life situation within the company. Independent of the used methodology collaboration with data owners and business representatives is of high importance. Results from iterations have to be documented and interpreted in coordination with all involved parties. Our future work in this phase of the lifecycle will concentrate on defining a hybrid Role Development Methodology connecting Role Mining and Role Engineering in an iterative process.

3.3 The Management-Layer

After the theoretical role catalogue has been defined and documented it is instantiated using modern IdM solutions. Following the final testing it can be used in the productive systems. The management tasks include everyday administration duties like User-Role-Assignment or Role-Permission-Assignment according to the given Administration Model. In addition to these rather simple tasks, the most important and sophisticated management duty is the up-to-date keeping of the role catalogue. None of the proposed Role Development Methodologies, Role Models, or Administration Models focuses on this task. Therefore role-checking will be part of the in-depth presentation of proROLE in chapter 4. In one of our projects the consequences of organisational changes are currently examined in order to give advice for potential updates of the role catalogue. Data mining algorithms like neuronal networks are used to compare the existing role catalogue with the actual situation within the company in order to identify discrepancies that have to be resolved, e.g. no longer used organisational as well as functional roles.

4 IN-DEPTH ANALYSIS OF PROROLE

After presenting the main layers of proROLE, this chapter goes into detail about its structure and various entities. In this work only parts of the complete model and its documentation will be presented due to space restrictions. Focusing on shortcomings of the available models we decided to give a detailed description of the data gathering/cleansing process as well as the periodic role-check loop that up to now have not been considered in existing approaches. In general, proROLE consists of the following entity types listed in Table 2. The letter symbol in the lower left corner symbolises additional clarifying documentation (e.g. special features and peculiarities of the entity) which is included in the UML source.

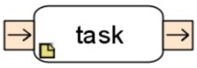
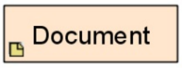
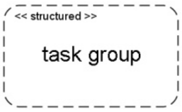
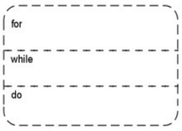
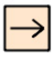
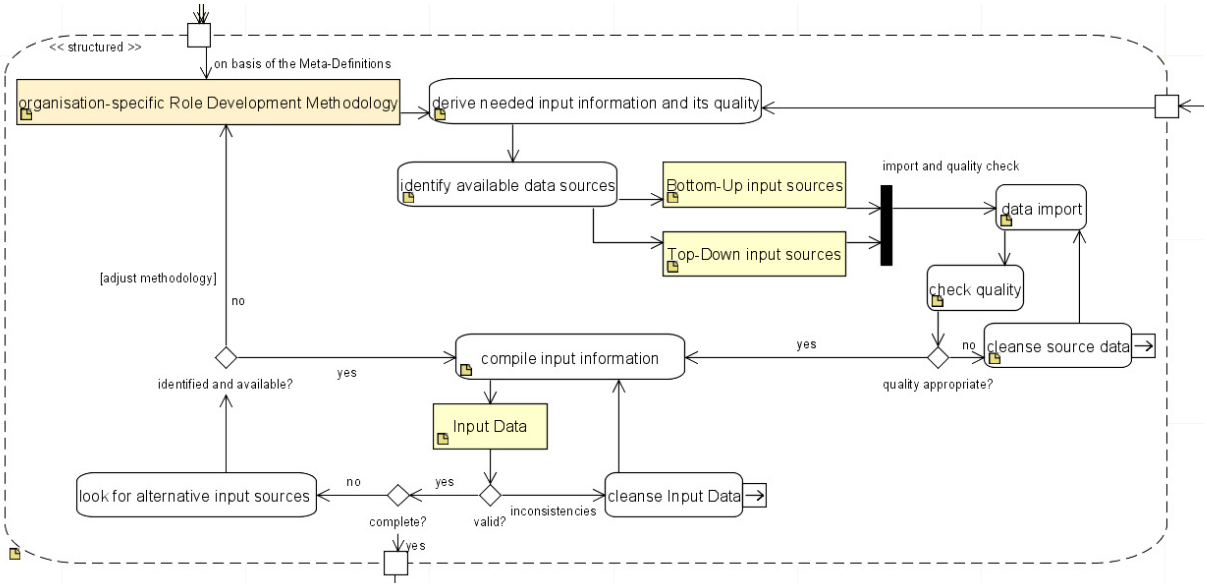
Symbol	Name	Definition
	Task	Duties within the role lifecycle. Every task can have certain prerequisite entities, input information, and a result which acts as input for the following entity.
	Document	Explicitly documented entities within the project, acting as basis of further development and being means of communication between involved parties.
	Task group	Collection of sub-tasks being part of a superior task within the lifecycle model. They consist of tasks, documents, and loops. All tasks included in a structured task group serve the overall goal of the parent structured task group.
	Loop	Task blocks within the lifecycle model that need to be carried out several times or even on a regular basis.
	Input/ Output nodes	Required input information and output documents or actions of entities.

Table 2: Entities of proROLE

4.1 Data gathering and cleansing

The data gathering and cleansing process is modelled as a structured task group including iterative loops (see Figure 5). It is located on the Development-Layer of proROLE as seen in Figure 4. On the basis of the Meta-Definitions, above all the Role Development Methodology, its goal is to collect and prepare the needed input information for the role creation process.

After the input information and its sources are derived and documented, the data has to be imported and checked to ensure it is of an appropriate quality. An example of invalid information could be user accounts of former employees. The data import is based on a replication processes between the Role Development System in place and source systems for input information, e.g. the IdM-System. Compiling and pre-processing input information is a major step within the data gathering process. It represents the matching of differently formatted data within one single role development environment. The result of this task is a repository of input information which can then be used within the role development process. In an iterative refinement its quality needs to be improved using data cleansing mechanisms. The main issues arise from the various source formats of the imported information. Similar attributes of user accounts could for example differ. In contrast to the earlier conducted system-specific data cleansing within the source systems themselves, this task now focuses on the already compiled input information, assuming that invalid or outdated account data has already been fixed in the source systems. However, the first cleansing does not take interdependencies between the various source systems into consideration. An employee could for example have different department or location values in two different systems. These issues are solved by cleansing the compiled input information. As a last check within the data gathering and cleansing task group, the input data is checked for completeness in terms of the prerequisite requirements of the Role Development Methodology. In case of an insufficient amount of available information, the project team has to locate alternative source systems for input information. If they cannot find any, the Role Development Methodology itself might have to be adjusted.



<< structured >>
on basis of the Meta-Definitions
organisation-specific Role Development Methodology
derive needed input information and its quality
identify available data sources
Bottom-Up input sources
Top-Down input sources
import and quality check
data import
check quality
[adjust methodology]
no
no
cleanse source data
yes
compile input information
yes
identified and available?
quality appropriate?
Input Data
look for alternative input sources
no
yes
cleanse Input Data
complete?
valid?
inconsistencies
yes

Figure 5: The data gathering and cleansing task group

4.2 Periodic role-check

As stated beforehand, executing periodic role-checks is one of the most sophisticated and important task loops within the Management-Layer of proROLE. It is of high importance to identify IT- as well as business changes affecting the role definitions in order to keep the role catalogue up-do-date. Examples of this include mergers, creation of new organisational units, or major restructuring efforts within a business area. The responsible role manager has to be able to react adequately and update the used roles in order to reflect a new situation. Even though this task will never be fully automated, he or she needs pre-processed information giving hints about possible changes. Therefore the periodic role-check is heavily based on the used Role Development Methodology and the appropriate data mining tools with their decision supporting capabilities. This helps minimise the manual effort when comparing the existing user-rights and role structures with the last valid situation.

The role-check is specified by
IR(t) = Input of role-check at time t
OP(t) = Synchronised output of the productive User- and Role Management Systems
OR(t) = Output of the role-check at time t

with

$$IR(t)=OP(t)+OR(t-1) \qquad (1)$$

During the checking process at time t Role Model values (for example user rights) of the last role-check OR(t-1) representing the last valid *"to-be situation"* have to be compared with their actual values within the productive systems OP(t). Focussing on user rights, one has to try to identify changes that lead to the definition of new roles, the update of existing roles or the deletion of unused outdated roles. In addition to that, changes in the organisational structure can give hints for potential alterations of the role catalogue. Located on the Management-Layer, the periodic role-check is modelled as a loop reflecting its recurring execution, either time-triggered or manually started. Figure 6 shows the detailed UML activity diagram of this loop included in proROLE. In order to avoid interdependencies with the productive system, the user information and the input information of the role system have to be copied to a test environment. Within this setting the comparison of old vs. new data takes place using well defined thresholds for the amount of changes of certain attributes. These discrepancy levels are needed to separate minor role changes from major role changes affecting the productive role catalogue or even Meta-Definitions.

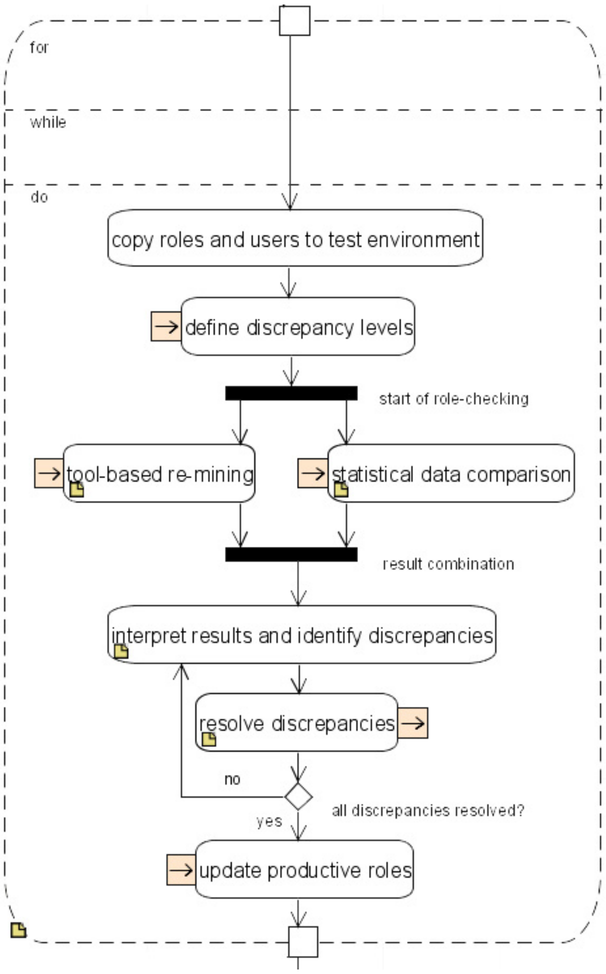

for
while
do
copy roles and users to test environment
define discrepancy levels
start of role-checking
tool-based re-mining
statistical data comparison
result combination
interpret results and identify discrepancies
resolve discrepancies
no
yes
all discrepancies resolved?
update productive roles

Figure 6: The periodic role-check loop

The role-check itself is split into two tasks, namely the statistical comparison of OP(t) and OR(t-1) on the one side and the re-mining of OP(t) on the other side. The statistical comparison is needed to identify changes in the organisational structure and user population, for example when employees change the department or a large number of new employees are hired. Re-mining is needed to visualise changes in the rights structure in an automated way. This can be done using neuronal networks or other clustering algorithms. As stated before we are currently focusing on the detection and treatment of such changes in a research project based on neuronal networks (Self-Organising Maps). After the statistical comparison and the re-mining are conducted, the results have to be interpreted and discrepancies found. Major discrepancies have to be resolved in an analytical way together with business representatives and IT-staff. In general, one can say that this duty is, similar to the role development itself, an iterative process, reducing discrepancies all along the overall role-check process loop. Finally, after all major discrepancies (according to the pre-defined discrepancy levels) have been resolved, the productive role catalogue and user information is updated using OR(t).

5 CONCLUSION AND FUTURE WORK

This paper presents proROLE, a comprehensive lifecycle model for a role system representing a process-oriented guideline for organising role projects. Its overall goal is to leverage an installed Identity Management solution to the next level of compliance, security-, rights- and user management. Focussing on role-based IdM as one main application area, this paper shows the importance of a structured and detailed guide for role implementation projects. Research has so far failed to give adequate support to role managers, implementation teams, and analysts. This essentially led to a large number of project failures over the last years. Our involvement in different IdM projects has shown the need for a comprehensive role system lifecycle model in order to improve the situation, satisfy industry's needs, and overcome the still existing organisation-wide retentive attitude towards role projects.

In comparison to the actual situation the lifecycle not only presents a list of unordered tasks related with a role system but also structures their sequence and focuses on mandatory documentation issues that are needed to ease communication and improve the overall security level within an organisation. In addition, all three layers of a role system are integrated in one comprehensive lifecycle reflecting the different duties from the project start-up to the daily administration. Moreover, a detailed specified data gathering and cleansing process and periodic role-check are integrated within the lifecycle model of a role system.

It is our aim that proROLE develops to a reference model for role implementation projects which supports all involved parties in better planning and structuring their efforts. In general, the lifecycle model is needed to steer all tasks related to the role system. This includes simple duties that are carried out on a daily basis as well as sophisticated tasks like the role development or periodic role-checks. For future work we are going to develop an iterative hybrid methodology for role development as well as metrics for the identification and handling of role changes. In addition, the application of our lifecycle is going to be tested in a real-life scenario giving us hints for further development.

Acknowledgement

The work reported in this paper will be continued within the SPIKE project which is funded by the European Union under the Seventh Framework Program (Contract No. 217098).

References

Basel II, Bank for International Settlements BIS: International Convergence of Capital Measurement and Capital Standards: A Revised Framework - Comprehensive Version. Available: http://www.bis.org/publ/bcbs128.pdf (2006).

Dhillon, G.: Violation of Safeguards by Trusted Personnel and Understanding Related Information Security Concerns. Computers & Security, Volume 20, Issue 2 , Pages 165-172 (2001).

European Union: Directive 95/46/EC of the European Parliament and of the Council. Official Journal of the European Communities of 23 November 1995 No L. 281 p. 31.. Available: http://www.cdt.org/privacy/eudirective/EU_Directive_.html (1995).

Ferraiolo, D. F., Kuhn, R. D., and Chandramouli, R.: Role-Based Access Control, Second Edition. Artech House computer security series, ISBN 1-59693-113-2 (2007).

Fuchs, L. and Pernul, G.: Supporting compliant and secure user handling – a structured approach for in-house identity management. Proc. of the 2nd International Conference on Availability, Reliability and Security (ARES '07), Vienna, Austria (2007).

Gallaher, M.P., O'Connor, A.C., and Kropp, B.: The economic impact of role-based access control. Planning report 02-1, National Institute of Standards and Technology, Available: http://www.nist.gov/director/prog-ofc/report02-1.pdf (2002).

Kern, A.: Advanced Features for Enterprise-Wide Role-Based Access Control. Proceedings of the 18th Annual Computer Security Applications Conference, pp. 333–342, Las Vegas, Nevada, USA (2002).

Kern A., Kuhlmann, M., Schaad, A., and Moffett, J.D.: Observations on the role life-cycle in the context of enterprise security management. Proceedings of the seventh ACM symposium on Access control models and technologies (SACMAT), pp. 43-51 Monterey, California, USA (2002)

Larsson, A.: A Case Study: Implementing Novell Identity Management. Proceedings of the 33rd annual ACM SIGUCCS conference on User services. November 6–9, Monterey, California, USA (2005).

Lupu, E.: A Role-Based Framework for Distributed Systems Management. PhD Thesis, Department of Computing. London, Imperial College (1998).

Neumann, G. and Strembeck, M.: A Scenario-driven Role Engineering Process for Functional RBAC Roles. 7th ACM Symposium on Access Control Models and Technologies (SACMAT). 33-42, Monterey, CA, USA (2002).

Pfitzmann, A. and Köhntopp, M.: Anonymity, unobservability, pseudonymity, and identity management – a proposal for terminology. Lecture Notes in Computer Science, Volume 2009, Springer, Heidelberg (2000).

Roeckle H., Schimpf, G., and Weidinger, R.: Process-oriented approach for role-finding to implement role-based security administration in a large industrial organisation. Fifth ACM Workshop on Role-Based Access Control, pp. 103–110, New York, USA (2000)

Sandhu R. S., Coyne, E.J., Feinstein, H.L., and Youman, C.E.: Role-Based Access Control Models. IEEE Computer 29(2): 38-47. IEEE Press (1996).

Sarbanes, P. S. and Oxley, M.: Sarbanes-Oxley Act of 2002 (Pub. L. No. 107-204, 116 Stat. 745), also known as the "Public Company Accounting Reform and Investor Protection Act of 2002". Available: http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_bills&docid=f:h3763enr.tst.pdf (2002).

Schimpf, G.: Role-Engineering Critical Success Factors for Enterprise Security Administration. Position Paper for the 16th Annual Computer Security Application Conference, New Orleans, USA (2000).

Stanton J. M., Stam, K. R., Mastrangelo, P., and Jolton, J.: Analysis of end user security behaviors. Computers & Security 24(2): 124-133 (2005).

Vaidjy J., Atluri, V., and Warner, J.: RoleMiner: mining roles using subset enumeration. Proceedings of the 13th ACM conference on Computer and communications security, pp. 144-153, Alexandria, Virginia, USA (2006).