HyDRo – Hybrid Development of Roles

Ludwig Fuchs, Günther Pernul

Department of Information Systems
University of Regensburg, 93053 Regensburg, Germany
{Ludwig.Fuchs | Guenther.Pernul}@wiwi.uni-regensburg.de

Abstract. Defining valid enterprise-wide roles needs to be carried out on the basis of a predefined Role Development Methodology. Hybrid role development combining elements from Role Engineering and Role Mining is the most promising way to define enterprise-wide roles, however no such model has been published yet. We close this gap by analysing existing approaches and proposing HyDRo, a tool-supported methodology that facilitates existing identity information and access rights without neglecting the importance of information like managers' knowledge about their employees.

Keywords: Role Development Methodology, Role Engineering, Role Mining, Identity Management, Information security.

1 Introduction and Motivation

As a result of ineffectual account management within organisations, users accumulate a number of excessive rights over time, violating the principle of the least privilege [1]. Major security problems arise because of employees gaining unauthorised access to resources as a result of manually handling user accounts ([2], [3]). This situation results in the so called identity chaos. In-house Identity Management (IdM) has become a means to solve the aforementioned identity chaos. It deals with the storage, administration, and usage of digital identities during their lifecycle. Roles acting as intermediary between employees and their access rights are an essential element of IdM. They allow companies to ease and secure provisioning processes, i.e. the allocation of digital and non-digital assets to employees, and access to resources in their IdM Infrastructure (IdMI) [4]. However, the most expensive challenge before achieving the benefits of role usage is the preliminary definition of valid roles [5]. Some companies deal with this issue by installing resource-intensive procedures based on organisational and operational structures. These approaches are known as Role Engineering Methodologies. In contrast, Role Mining Methodologies create roles using data mining tools that analyse and cluster existing user permissions providing a high degree of automation. This paper underlines the need for hybrid role development combining Role Engineering and Role Mining as the most promising approach for defining enterprise-wide roles. Up to now, to the best of our knowledge, no such model has been published. Hence, the main goal of this work is to close this gap by proposing HyDRo, a hybrid Role Development Methodology (RDM) that integrates Role Engineering and Role Mining elements into a comprehensive framework for role creation. Central modelling requirements of HyDRo are the

shortcomings of existing models, literature analysis, practical experiences, and requirements from industry partners.
This paper is structured as follows. In section 2 we present and compare existing Role Development Methodologies in order to show their shortcomings. Subsequently, section 3 introduces HyDRo, a methodology for hybrid development of roles. In section 4 we provide an overview over contROLE, a role development tool that supports HyDRo. Conclusions and future work is given in section 5.

2 Existing Role Development Methodologies

Role Development Methodologies can in general be categorised according to the input information they are based on (see figure 1): *Role Engineering* is considered as the theoretical way of developing roles where roles are derived Top-Down based on information from the OOS (Organisational and Operational Structures) layer within an enterprise. This includes knowledge about hierarchical structures, process- or workflow definitions, or employees' task bundles. Role Engineering following an aggregation or decomposition approach offers the chance to define a role catalogue that is closely aligned to the business perspective within a company. Decomposition approaches define roles and break them down into permissions needed while aggregation works the opposite way [6]. *Role Mining* on the contrary is the tool-based Bottom-Up approach discovering roles using existing identity information and access rights from the Directory layer. It in general investigates users and their existing access rights and is usually based on clustering algorithms which can be divided into statistical clustering or usage of neuronal networks.

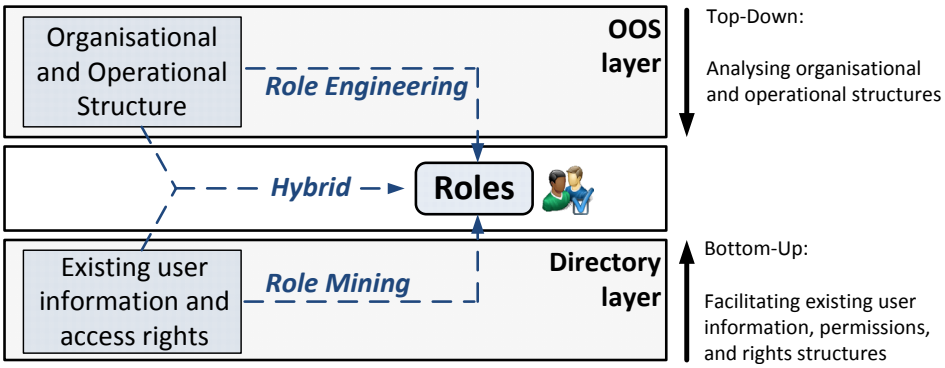

Fig. 1. Role Development Methodologies

We define role development as the umbrella term for Role Engineering and Role Mining. Role development can be carried out hybrid or non-hybrid. Several publications explicitly mention that a hybrid combination of Role Engineering and Role Mining is necessary to define a good collection of roles ([7], [8], [9], [10], [11], and [12]). However, although considered to be the most promising way for developing roles on a company-wide level, no such hybrid RDM has been published up to now.

2.1. Role Engineering (Top-Down)

The importance of Role Engineering was first mentioned by Edward Coyne [13] after the upcoming of the original RBAC (Role-Based Access Control) model [14] in 1996. Role Engineering following the **decomposition approach** involves an in-depth analysis of business processes and functional structures in order to break down these elements to system-specific features needed to fulfil certain tasks. Crook et al. [9] showed that using organisational structure to define roles has significant advantages by providing a clear focus for analysts and users eliciting requirements. Roeckle et al.'s approach [8] on the contrary integrates business processes into the Role Engineering duties. They aim at finding the complete IT supported set of job functions performed in an organisation. While decomposition is used mainly for defining system-independent roles, **aggregation approaches** are adopted in the process of developing application-specific roles. They are based on use case- or scenario descriptions (scenarios can be regarded a specific representation of a use case [15]), goals, or other input information. In general, aggregation approaches define the way of interaction with an application and the bundles of permissions needed to fulfil certain tasks within this application. In order to streamline the mainly manual aggregation process, Strembeck presented a tool-based technique for defining scenarios ([16], [17]) and extract RBAC-models from BPEL4WS processes [18].

Shortcomings
Role Engineering significantly depends on human factors and the amount and quality of input information available. Above all in settings where the quality of organisational charts and job descriptions is high, Role Engineering is a promising approach to find role candidates. However, on the other hand it is primarily a manual task involving extensive communication between stakeholders [19]. A comparison of existing Role Engineering methods showed that only decomposition approaches are feasible for developing system-independent roles. Aggregating single elements like tasks comprehensively into roles is not applicable in an enterprise-wide project as most approaches are lacking any tool support. With dozens of business processes, thousands of users, and millions of authorisations in big organisations, this is seemingly a difficult task. Besides the high complexity and costs the collection and preparation of input information are the main drawbacks ([8], [19]). Practical experience has moreover shown that Role Engineering neglects existing access rights and thus the actual situation within a company. Hence, relying solely on Role Engineering for defining company-wide roles is not feasible.

2.2. Role Mining (Bottom-Up)

As a result of the presented shortcomings of Role Engineering, Role Mining has over the last years evolved as the pragmatic approach to rapidly define adequate roles. It specifically focuses on the usage of data mining technology for definition of system-independent roles that can, amongst others, be used in IdMIs for user management. Role Mining automates the development process by using tools to identify potential roles. In contrast to Role Engineering, Role Mining is based on the assumption that the actual roles already exist within the IT infrastructure. Existing permission assignments are aggregated to define role candidates using statistical clustering algorithms or neuronal networks. Statistical clustering can be carried out

hierarchically or partitioning whereas neuronal networks use unsupervised learning methods for role development. In [7] Vaidya et al. surveyed existing Role Mining approaches that mostly present heuristic ways to find a set of role candidates. Kuhlmann et al. ([20], [21]), ORCA [19], and Vaidya et al. [22] are identified as the most important publications in that area. Kuhlmann et al. propose a clustering technique closely related to the k-means algorithm. In [19], Schlegelmilch et al. facilitate an agglomerative hierarchical clustering based algorithm, which discovers roles by merging permissions appropriately. Additionally, Vaidya et al. [22] propose RoleMiner, an approach based on subset enumeration. Recently [10], [23], [24], and [25] have presented specific improvements, integrating cost and performance decisions as well as semantics into Role Mining.

Shortcomings
Even though providing a high degree of automation, Role Mining has several serious unaddressed drawbacks: If the input quality is erroneous the role candidates discovered are also incorrect. Existing approaches assume that cleansing already took place before the role definition. We argue that this issue needs to be addressed by introducing a mandatory customisable data cleansing and -preparation phase as shown in [11] in order to ensure an appropriate quality level of the input information. Investigating existing literature has moreover shown that most publications only present algorithms for finding the optimal role set without taking into consideration that business needs have to be involved in a role development project. As it is not their main focus, none of them adheres to existing methodological requirements.

3 HyDRo – A Methodology for Hybrid Development of Roles

As aforementioned, neither pure Role Engineering nor pure Role Mining leads to an optimal role catalogue. Our analysis and practical experiences with existing RDMs underline these findings stating that a hybrid approach is the most promising basis for role creation. On the one hand the automation capabilities of Role Mining are needed while consideration of business functions and organisational structure is a mandatory element of a RDM on the other hand. None of the existing approaches shows how Role Engineering and Role Mining can be combined and how the information flows can be structured. Based on shortcomings of existing models, literature analysis, and practical experiences we are now going to introduce HyDRo, a new hybrid methodology for developing roles. The goal of HyDRo is the definition of system-independent roles usable within IdMIs. HyDRo considers existing user information and access right structures without neglecting the importance of organisational structures and information like managers' knowledge about their employees. It can be easily integrated into the proROLE framework [11] representing a role system lifecycle. The underlying philosophy is perform a joint Role Mining/Engineering approach and integrate OOS layer representatives (managers, executives, CIO) as frequently as necessary but as infrequently as possible.

Methodological Background
In complex environments role development projects need to be carried out on basis of a predefined methodology in order to derive a consistent role catalogue and reducing failure risks. However, existing Role Engineering and Role Mining approaches lack a

clear definition of mandatory method elements. HyDRo overcomes these significant shortcomings by being modelled based on method elements following well-defined Method Engineering principles ([26], [27]) (see figure 2, adapting the notation used by Brinkkemper [28]). We propose the **Procedure Model** as the central element of our methodology. Grey colouring marks elements directly integrated in the HyDRo procedure model: Necessary **Activities** and **Techniques** structured in six **Phases**, involved **Roles** (i.e. stakeholders), the documentation of **Results**, and the **Tool** used throughout the entire process. HyDRo is fully supported by the *contROLE* role development software which will be presented in section 4. HyDRo furthermore uses busiROLE [29] as **Meta-Model**.

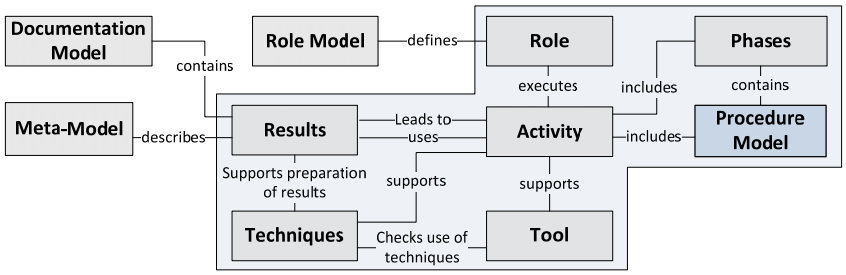

Fig. 2. Method Elements of HyDRo

3.1. Overview and Characteristics

In this section we analyse the different phases of HyDRo on basis of the aforementioned method elements. The methodology consists of six consecutive main phases and the respective interfaces in form of Quality Measurement (QM) and Execution Decision (ED) activities. Organisations applying HyDRo need to complete one phase to a predefined extent to be able to move on to the next phase. However, HyDRo is designed to provide maximum flexibility during role development; hence, users of the methodology can move back to previous phases or within phases in an incremental and iterative fashion. Companies applying HyDRo can re-run a single phase if the resulted quality is insufficient or new input is provided changing existing results. Figure 3 provides a high level overview of the main phases: The HyDRo process starts with the import of necessary input data (**Data Gathering**) and consecutive **Data Cleansing**. In order to define suitable roles, the input data is then classified and selected in a separate phase (**Data Preparation and Selection**). The role development process itself is split into three phases, namely the definition of **Basic-, Organisational-**, and **Functional Roles**. We will discuss each of the phases in more detail in the following sub-section 3.2.

Quality Measurement and Execution Decision
One central business requirement for a hybrid RDM is the definition and measurement of partial results during the methodology execution. Business- as well as IT representatives desire milestones during the role development project that form the basis for further execution decisions. Figure 3 indicates the partial result measurement and -decisions at each transition between two phases. Every HyDRo phase ends with a QM and ED process step. Depending on the phase, different

indicators provide information about the result quality. Data cleansing output quality can e.g. be measured by analysing the percentage of corrected input data. In contrast to the QM task, where the result quality of one phase is measured, ED activities take general project drivers into account. Even if the result quality of one phase is sufficient, companies applying HyDRo still might want to abort the role development as result of e.g. lacking funding, other prioritised projects, or time schedule issues.

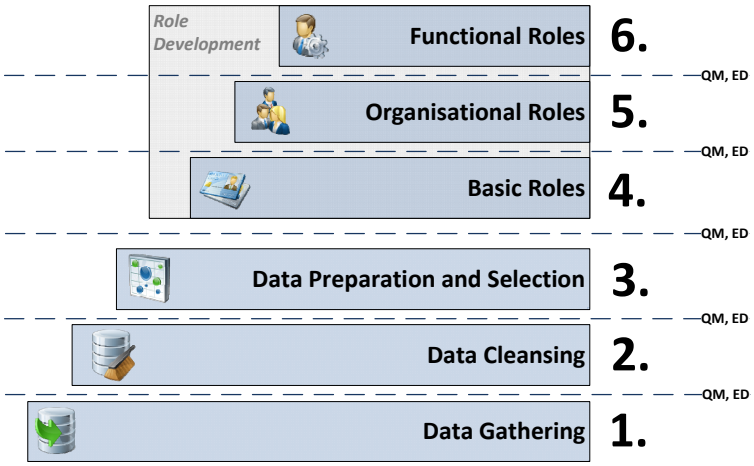

Fig. 3. Phases of HyDRo

3.2. HyDRo Phases

In the following we are going to introduce HyDRo on basis of its **Procedural Model** split into the six main **Phases** shown in figure 3. In order to highlight the interdependencies and information flows between the OOS- and Directory layer we use a visualisation schema integrating the **Phases** and **Activities**, the included stakeholders (**Roles**), the used **Tool**, as well as the derived **Results** (figure 4, 5, and 6). This allows for a clear distinction of Role Mining and Role Engineering elements.

3.2.1 Data Gathering

Within the data gathering phase input information from Role Engineering and Role Mining sources needed for hybrid role development is imported. The goal of this phase is the compilation of a consistent raw input information repository representing the basis for further data cleansing-, data preparation-, and role development activities. Thus, after the kick-off of the role development project using HyDRo the various available input sources are identified. Information needs to be imported and checked for consistency. HyDRo considers existing user rights in form of a LDAP-repository or a .csv-file as mandatory raw data from the Directory layer. If this information is not available, HyDRo is not applicable or, if exclusively input information from the OOS layer is available many Role Mining activities are not executable. Besides the mandatory identity information, input from the OOS layer is optional but highly desired. It might be available in forms of defined job positions, task bundles, processes, or already existing local role definitions from certain departments. Existing Top-Down knowledge is imported in order to compose the raw

input information repository consisting of all available OOS- and Directory layer input information. After a basic quality measurement which checks if enough input information is available for consecutive HyDRo phases, business representatives like the manager of an organisational unit, need to review the information available for his department. This way he can alter basic information about his employees and identify employees with active user accounts but who are no longer working for the company.

3.2.2 Data Cleansing

If a certain minimum of input information is available, HyDRo continues with the data cleansing phase (see figure 4). The overall goal of this phase is to improve the data quality of the input information and thereby overcome the deficits of most existing approaches which do not include any data cleansing mechanisms. Data Cleansing in HyDRo is split into syntactic- and semantic cleansing. While syntactic checks might be fully automatable, semantic checks cannot be processed without human intervention. Consider an employee in a multinational organisation. One can imagine that misspelled user attributes like his location attribute within the global identity repository can easily be identified. However, if a user has a wrong assignment of a valid location, it is not possible to resolve this inconsistency without any information from the OOS layer. During this phase Role Engineering- and Role Mining activities need to be combined to achieve the best results. HyDRo provides tool support by facilitating various syntactic checks, including duplicate checks or attribute checks against valid values. Even more important, it also provides various functions for semantic analysis of the underlying input information. For instance Self-organising maps (SOM) [30] are used to identify users which have untypical attribute values assigned. Again, consider our previously mentioned example of an employee working for a multinational organisation. After he changed his location and has been assigned to new privileges he might still be assigned to a number of his former access rights. SOMs can be used to identify and highlight such a user because some of his assigned privileges are typical for a different location than the one he is assigned to.

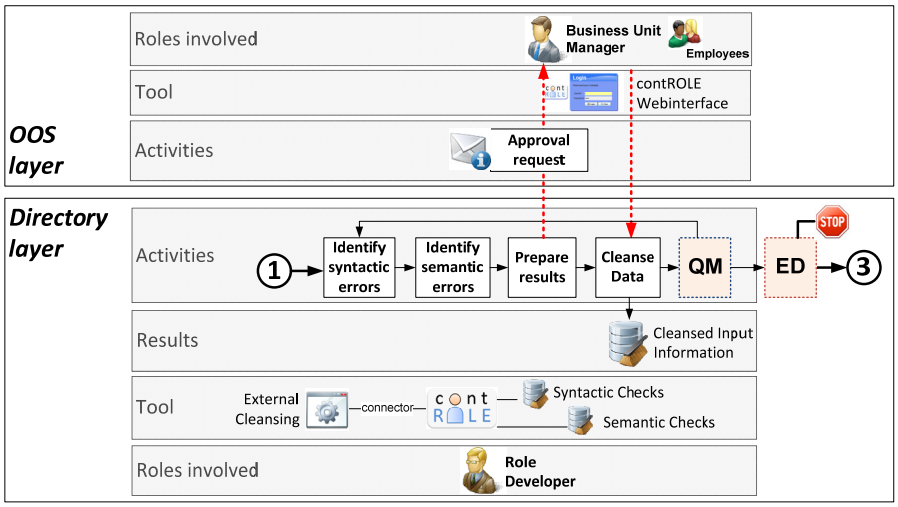

Fig. 4. The Data Cleansing Phase of HyDRo

3.2.3 Data Preparation and Selection

After input data has been cleansed the role development process moves on to the Data Preparation and Selection phase which exclusively comprises activities at the Directory layer (figure 5). Its goal is to generate additional knowledge about the underlying input information. We argue it is mandatory to allow companies to choose the appropriate part of the raw input data to be included in the role development. HyDRo starts by analysing the underlying input information on a global level. One aspect is the exclusion of further manually administered rights. Our experience has shown that a high number of rights are only held by a small number of users making them not feasible for role-based allocation. After the global classification and selection process, single hierarchical elements are classified locally. A hierarchical element is a unit in the organisational structure of an enterprise, for example a business unit, a department, or a unit within a department. Techniques like statistical analysis, clustering algorithms, or results from previous RDM phases can be used. The generated information might be of high relevance for business representatives as well as role developers: In contROLE, for example, a green traffic light in front of an organisational unit represents simple user- and access rights structures or a high amount of cleansed input information. In this case the respective organisational unit is a candidate for rapid role development. On the contrary, a red traffic light classifies organisational units as improper for easy role development as a result from e.g. a lack of cleansed input information. However, automatically classifying input information needs to be carefully parameterised. Fundamentals are the predefined classes of users and access rights, the amount of cleansed input information, or, even more complex, the grade of interdependencies between hierarchical elements.

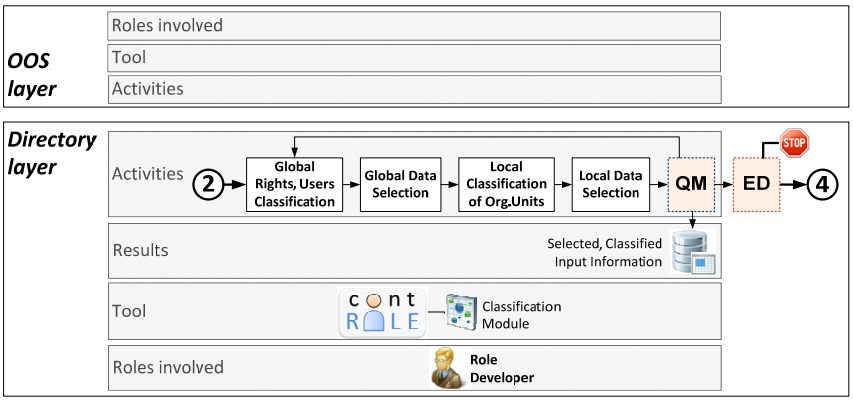

Fig. 5. The Data Preparation and Selection Phase of HyDRo

3.2.4. Role Development

After the input information has been cleansed and prepared, phases 4, 5, and 6 represent the actual role development phases of HyDRo. The outcome of each of those phases is a set of defined roles of a certain type: Basic Roles (phase 4) bundle common access rights. Organisational Roles (phase 5) represent job positions while Functional Roles (phase 6) correspond to the task bundles of employees. By allowing for the definition of business role types [29], HyDRo supports incremental role development. Phases 4, 5, and 6 from figure 3 are modelled similar, even though the underlying algorithms, the extent of hybrid communication, and the importance of

OOS layer input information are varying. The amount of required Top-Down input is constantly increasing while the usage of Role Mining algorithms is decreasing as a result of growing complexity of role definition. However, companies can decide whether they want to define Functional Roles or whether they abort HyDRo after the definition of Basic- and Organisational Roles. In some cases a large part of the access rights might be already administered using those types of roles in which case the additional flexibility gained by the usage of Functional Roles is outweighed by the large amount of time and money spent for their definition.

Basic Roles

HyDRo starts with defining Basic Roles that are assigned on basis of organisational membership of users in different hierarchies. In general, they represent the bundle of access rights that are granted to every employee in a certain hierarchical element independent from his job position. They can be inherited, depending on the hierarchy type. Basic Roles could include rather common rights like "Internet Access" and are derived using Role Mining algorithms. The permissions assigned to a certain percentage of the employees within an element, e.g. more than 90%, are bundled and marked as a possible Basic Role. The manager of this hierarchical element is then informed via email that he needs to approve or alter the found role candidates.

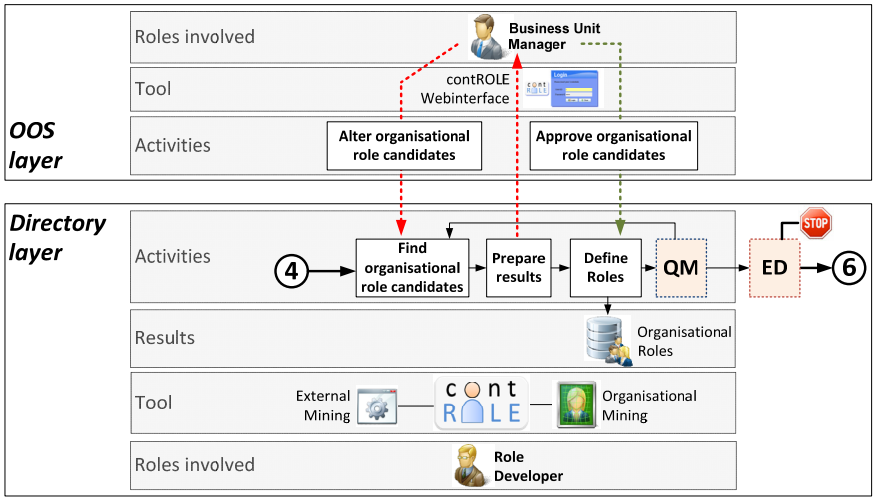

Fig. 6. Role Development in HyDRo (example: Phase 5 "Organisational Roles")

Organisational Roles

In phase 5 Organisational Roles, i.e. job positions of employees within the organisational structure are derived (see **Fehler! Verweisquelle konnte nicht gefunden werden.**6). It is likely that in some hierarchical elements positions are already defined. In other situations this might not be the case and Role Mining technologies are needed to cluster employees with the same access rights. These clusters need to be visualised appropriately and presented to the OOS representatives. Quality criteria for this phase might be the number of employees which are assigned to a certain Organisational Role, the quality of feedback derived from the OOS representatives, or the percentage of permissions that is administered by the usage of the defined Organisational Roles.

Functional Roles
Functional Roles represent task bundles of employees and are amongst others used for delegation purposes. They might be specific for a hierarchical element or valid throughout the whole enterprise. On the one hand, the employees' job positions have to be split up into various task bundles. On the other hand the task bundles might represent special duties of a number of employees, independent from any organisational hierarchy. The development of Functional Roles needs to be heavily supported by human interaction. Finding task bundles purely based on Role Mining algorithms is hardly possible. Automatically analysing employees with different job positions whose permissions overlap is one possible approach.

4 The contROLE Role Development Tool

HyDRo is fully supported by *contROLE*, a role development tool which currently is being implemented at our department. ContROLE is not designed as a Role Mining tool according to section 2.2. It is rather an infrastructural tool that fosters the hybrid integration of Role Engineering by providing a maximum of process- and communication automation. Figure 7 shows the main interface of contROLE during Data Cleansing with the single phases of HyDRo seen in the upper window. Using client-server architecture, OOS representatives as well as Role Developers can use contROLE during the hybrid role development loops. Due to the limited space we only present selected features of contROLE and give a short quality analysis of implemented Role Mining algorithms using various predefined input data sets.

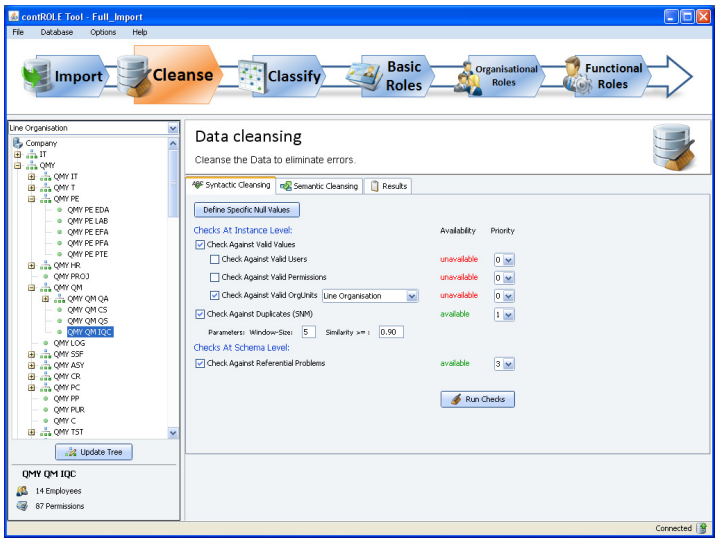

Fig. 7. ContROLE Role Developer Interface

As stated beforehand, contROLE offers a wide variety of data cleansing and data preparation functionalities that have to the best of our knowledge not been integrated in any other RDM. Syntactic- and semantic data checks can be used for cleansing the input information. This includes checks against valid permissions, users, and

organisational hierarchy elements. Moreover contROLE is able to detect outliers and suspicious user-permission assignments. The role developer can design and parameterise a data cleansing process according to the available input information. For detecting and visualising outliers contROLE amongst others implements a connection to the *SOM Toolbox*, an already existing implementation of SOMs developed in the GHSOM Project [31]. We facilitate the capabilities of SOMs to find suspicious users in terms of wrong attributes or erroneous rights allocation. Figure 8 shows the visualisation of Directory layer information of one of our industry partners. One can see various suspicious data elements (arrows). These elements are automatically marked by contROLE and sent to the respective managers for approval together with a proposed attribute value.

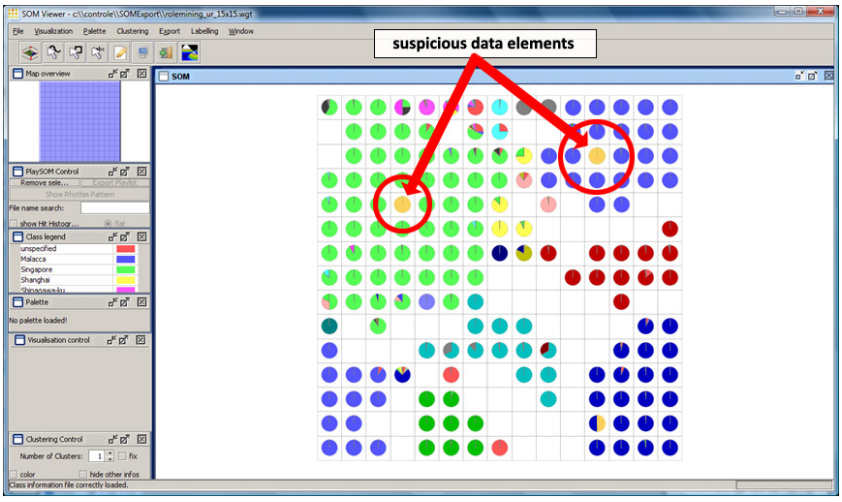

Fig. 8. SOM Representation of Access Rights

As mentioned beforehand many access rights might still be administered manually because they represent special permissions held by only a very small number of employees. Figure 9 represents the examination of access rights structures of one representative line organisation department of a large industrial company consisting of 103 users and 223 different access rights. The visualisation used orders the different access rights on the horizontal axis according to the number of users being granted this right (ascending). It can be seen that a very large number of rights are only held by one user (108 rights). This underlines the need for a careful data selection in order to ensure the definition of usable roles. ContROLE can automatically pick rights that should be further manually administered while the role developer can additionally disable rights for the consecutive role development phases. A further analysis of figure 9 points out that a relatively small number of rights are held by all users of this department, representing possible Basic Roles.

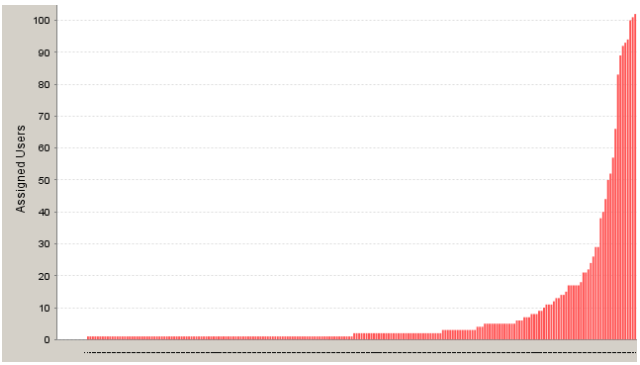

Fig. 9. Access Right Structures within a Hierarchical Element

ContROLE Role Mining Performance Analysis

Regarding the definition of role candidates, contROLE implements the Role Mining algorithms ORCA [19], FastMiner, and CompleteMiner (both presented in [22]). The quality of the clustering results presented in this section led to the development of our own Role Mining algorithm which is currently being implemented and tested. In order to validate the result quality of the already existing algorithms we carried out a performance and clustering analysis on basis of evaluation principles shown by Pries-Heje et al [32]. We decided to conduct an ex-post evaluation using artificial and naturalistic input datasets (see table 1).

Table 1. Input datasets for the Role Mining quality analysis

Dataset	Users	Permissions
Artificial Set 1 (AS1)	6	4
Artificial Set 2 (AS2)	13	4
ContROLE Artificial Set (AS3)	168	29
Small Naturalistic Set (NS1)	362	366
Big Naturalistic Set (NS2)	1211	805

AS1 and AS2 were included in the corresponding Role Mining publications [19] and [22] representing simple illustrative examples of user and permission structures. They include only a small number of user permission assignments. AS3 has been designed for contROLE functionality tests and represents a middle sized company with 168 employees. In contrast to the artificial sets AS1 and AS2 it includes organisational structures and hierarchies. NS1 and NS2 are both naturalistic datasets provided by our user partners, gathered from their global Identity Management System in place. They each represent user permission assignments of one department within the line organisation including a large number of sub-departments and hence complex hierarchical structures. A performance analysis revealed that most Role Mining algorithms were able to finish the computation process in a reasonable time (up to 360 seconds depending on the dataset size). Due to its complexity the CompleteMiner was the only exception not able to finish the role candidate discovery on basis of the big naturalistic set NS2. The hardware used for the computation was a machine with an Intel Core2Duo CPU and 3GB RAM. Our in-depth analysis moreover revealed that

the result quality of the implemented algorithms strongly varies. Table 2 gives an overview over the number of derived role candidates (RC) using the given datasets from table 1 as input information for the Role Mining Algorithms implemented in contROLE. Note that the number of clusters using ORCA varies as a result of a random component within the algorithm. Hence we display an average value computed during various test loops. The reduced role candidate number computed using a minimum role membership (RM) limit is also given if that feature was provided by the corresponding algorithm.

Table 2. Found role candidates

Dataset	ORCA	FastMiner	CompleteMiner
AS1	4 Cluster	3 RC	2 RC
AS2	2 Cluster	4 RC	4 RC
AS3	18 Cluster	27 RC	27 RC
		14 RC (min=10 RM)	*14 RC (min=10 RM)*
NS1	~295 Cluster	1648 RC	2121 RC
		108 RC (min=20 RM)	*301 RC (min= 20 RM)*
NS2	~621 Cluster	14386 RC	
		647 RC (min=20 RM)	

Especially the analysis of the naturalistic datasets revealed that Role Mining algorithms tend to discover a large number of candidate roles which is not feasible in practical scenarios. Using NS2 as input information the FastMiner, e.g., discovered 14386 role candidates for 1211 users. Even though FastMiner and CompleteMiner both support the parameterisation with a minimum number of role members RM, the found role candidates still only can be seen as preliminary results. The biggest drawback of existing Role Mining algorithms is the missing integration of additional input information in the role discovery process. They exclusively facilitate user-permission assignments as input information, neglecting existing hierarchical and operational structures within a company. During the further development of contROLE we hence focus on the integration of OOS layer information into the role definition process.

5 Conclusions and Future Work

In this paper we have argued the need for a hybrid Role Development Methodology integrating Role Engineering and Role Mining functionalities. A literature analysis has investigated the existing models and shown that none of them sufficiently meets the requirements of a RDM. In order to close this gap we proposed HyDRo, to the best of our knowledge the first hybrid Role Development Methodology. HyDRo considers existing user information and access right structures without neglecting the importance of information like managers' knowledge about their employees. It overcomes the shortcomings of existing RDMs by being based on a well-defined method engineering basis and providing a comprehensive set of role development phases. Above all the Data Cleansing and Data Preparation and Selection phase have not been included in existing approaches. HyDRo moreover considers various business requirements in order to ensure applicability within real-life scenarios. One

big advantage is the seamless tool-support. The contROLE software ensures the hybrid integration, cleansing, and selection of input information from various sources throughout an iterative and incremental role development process.
For future work we are focussing on the extension of the functionality of contROLE, especially the extension of classification- and Role Mining algorithms as well as the improvement of the user-interface. We furthermore are going to deal with performance issues arising when working with large datasets. This task affects above all the usage semantic Data Cleansing algorithms and the training of neuronal networks which can be a longsome process.

References

1. Ferraiolo, D. F., Kuhn, R. D., Chandramouli, R.: Role-Based Access Control. Artech House, Boston, Mass./London (2007)
2. Larsson, E. A.: A case study: Implementing Novell Identity Management at Drew University. In: Proc. of the 33rd annual ACM SIGUCCS conference on User services (SIGUCCS'05), pp. 165-170. ACM, New York (2005)
3. Dhillon, G.: Violation of Safeguards by Trusted Personnel and Understanding Related Information Security Concerns. Computers & Security 20 (2), pp. 165-172 (2001)
4. Fuchs, L., Pernul, G.: Supporting Compliant and Secure User Handling – a Structured Approach for In-house Identity Management. In: Proc. of the 2nd Int. Conference on Availability, Reliability and Security (ARES '07), pp. 374-384. IEEE Computer Society, Washington (2007)
5. Gallaher, M. P., O'Connor, A. C., Kropp, B.: The economic impact of role-based access control. Planning report 02-1, National Institute of Standards and Technology, Gaithersburg, MD (2002), http://www.nist.gov/director/prog-ofc/report02-1.pdf
6. Epstein, P., Sandhu, R.: Engineering of Role/Permission Assignments. In: Proc. of the 17th Annual Computer Security Applications Conference (ACSAC'01), IEEE Computer Society, Washington (2001)
7. Vaidya, J., Atluri, V., Guo, Q.: The role mining problem: finding a minimal descriptive set of roles. In: Proc. of the 12th ACM Symp. on Access Control Models and Technologies (SACMAT '07), pp. 175-184, ACM, New York (2007)
8. Roeckle, H., Schimpf, G., Weidinger, R.: Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. In: Proc. of the 5th ACM workshop on Role-based access control, pp. 103-110. ACM, New York (2000)
9. Crook, R., Ince, D., Nuseibeh, B.: Towards an Analytical Role Modelling Framework for Security Requirements (2002), http://mcs.open.ac.uk/ban25/papers/refsq02.pdf
10. Colantonio, A., Di Pietro, R., Ocello, A.: Leveraging Lattices to Improve Role Mining. In: Proc. of the 23rd Int. Information Security Conference (SEC 2008)
11. Fuchs L., Pernul G.: proROLE: A Process-oriented Lifecycle Model for Role Systems. In: Proc. of the 16th European Conference on Information Systems (ECIS), Galway, Ireland (2008)
12. Shin, D., Ahn, G., Cho, S., Jin, S.: On modeling system-centric information for role engineering. In: Proc. of the 8th ACM Symp. on Access Control Models and Technologies (SACMAT '03), pp. 169-178. ACM, New York (2003)
13. Coyne, E. J.: Role Engineering. In: Proc. of the 1st ACM Workshop on Role-based access control, ACM, New York (1996)
14. Sandhu, R. S., Coyne, E. J., Feinstein, H. L., Youman, C. E.: Role-Based Access Control Models. IEEE Computer 29 (2), 39-47 (1996)

15. Sadahiro, I.: A Critique of UML's Definition of the Use-Case Class. In: UML 2003. LNCS, vol. 2863, pp. 280–294, Springer, Heidelberg (2003)
16. Neumann, G., Strembeck, M.: A scenario-driven role engineering process for functional RBAC roles. In: Proc. of the 7th ACM Symp. on Access Control Models and Technologies, pp. 33-42. ACM, New York (2002)
17. Strembeck, M.: A Role Engineering Tool for Role-Based Access Control. In: Proc. of the Symp. on Requirements Engineering for Information Security (SREIS), Paris, France (2005)
18. Mendling, J., Strembeck, M., Stermsek, G., Neumann, G.: An Approach to Extract RBAC Models from BPEL4WS Processes. In: Proc. of the 13th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), pp. 81-86. IEEE Computer Society, Washington (2004)
19. Schlegelmilch, J., Steffens, U.: Role mining with ORCA. In: Proc. of the 10th ACM Symp. on Access Control Models and Technologies (SACMAT'05), pp. 168-176. ACM, New York (2005)
20. Kuhlmann, M., Shohat, D., Schimpf, G.: Role mining - revealing business roles for security administration using data mining technology. In: Proc. of the 8th ACM Symp. on Access Control Models and Technologies (SACMAT '03), pp. 179-186. ACM, New York (2003)
21. Kern, A., Kuhlmann, M., Schaad, A., Moffett, J.: Observations on the role life-cycle in the context of enterprise security management. In: Proc. of the 7th ACM Symp. on Access Control Models and Technologies (SACMAT '02), pp. 43-51. ACM, New York (2002)
22. Vaidya, J., Atluri, V., Warner, J.: RoleMiner: mining roles using subset enumeration. In: Proc. of the 13th ACM Conf. on Computer and Communications Security (CCS '06), pp. 144-153. ACM, New York (2006)
23. Colantonio, A., Di Pietro, R., Ocello, A.: A cost-driven approach to role engineering. In: Proc. of the 2008 ACM Symp. on Applied Computing, pp. 2129-2136. ACM, New York (2008)
24. Molloy, I., Chen, H., Li, T., Wang, Q., Li, N., Bertino, E., Calo, S., and Lobo, J.: Mining roles with semantic meanings. In: Proc. of the 13th ACM Symp. on Access Control Models and Technologies (SACMAT '08). ACM, New York (2008)
25. Vaidya, J., Atluri, V., Guo, Q., and Adam, N. 2008. Migrating to optimal RBAC with minimal perturbation. In: Proc. of the 13th ACM Symp. on Access Control Models and Technologies (SACMAT '08). ACM, New York (2008)
26. Braun, C., Wortmann, F., Hafner, M., Winter, R.: Method Construction – A Core Approach to Organizational Engineering. In: Proc. of the 2005 ACM Symposium on Applied Computing, pp. 1295-1299. ACM, New York (2005)
27. Gutzwiller, T.: Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen. Physica-Verlag, Heidelberg (1994)
28. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38, 275-280 (1996)
29. Fuchs, L., Preis, A.: BusiROLE: A Model for Integrating Business Roles into Identity Management. In: Proc of the 5th Int. Conference on Trust, Privacy, and Security in Digital Business (TrustBus), Torino, Italy (2008)
30. Kohonen, T.: Self-Organizing Maps. Springer Verlag, Berlin (2001)
31. "The SOMLib Digital Library Project", Information & Software Engineering Group, Vienna University of Technology, http://www.ifs.tuwien.ac.at/~andi/somlib/index.html
32. Pries-Heje, J., Baskerville, R., Venable, J.: Strategies for Design Science Research Evaluation. In: Proc. of the 16th European Conference on Information Systems (ECIS), Galway, Ireland (2008)