2009

Towards Business Driven Web Service Authorization – Project Experiences in German University Administrations

Martin Juhrisch
University of Technology Dresden, martin.juhrisch@tu-dresden.de

Gunnar Dietz
Multimedia Kontor Hamburg GmbH, g.dietz@mmkh.de

Jens Weller
University of Technology Dresden, jens.weller@tu-dresden.de

Werner Esswein
University of Technology Dresden, werner.esswein@tu-dresden.de

Towards Business Driven Web Service Authorization – Project Experiences in German University Administrations

Martin Juhrisch
Chair of Business Informatics,
esp. Systems Engineering
University of Technology Dresden
martin.juhrisch@tu-dresden.de

Gunnar Dietz
Multimedia Kontor Hamburg GmbH
g.dietz@mmkh.de

Jens Weller
Chair of Business Informatics,
esp. Systems Engineering
University of Technology Dresden
jens.weller@tu-dresden.de

Werner Esswein
Chair of Business Informatics,
esp. Systems Engineering
University of Technology Dresden
werner.esswein@tu-dresden.de

ABSTRACT

The management of web service access is a great challenge primarily in organizations with a high number of web services and various sorts of access authorization. Both organizational and technical hurdles have to be overcome. Technical barriers arise by different security requirements across various systems under consideration of economic conditions. Complex service interaction may be mentioned here, too. Organizational challenges comprehend the identification of requirements, managing the data access and management of changes to permissions. An important issue addresses the effective authorization of users via identity management systems. With established conceptual modeling languages the assignment of roles to the identity management software is an enormous organizational effort. To decrease administration costs we propose a direct connection between an identity management system and enterprise models which contain the organizational responsibilities. This paper describes a method developed at the University of Dresden exemplified in a research project for the construction and administration of an identity management system at the Universities in Hamburg.

Keywords

Enterprise Models, Identity Management, E-Government, Conceptual Modeling.

INTRODUCTION

Service-oriented architectures (SOA) were introduced to create a technological basis for reacting to business requirements in a distributed application environment (Erl 2005, Krafzig et al. 2005). Services encapsulate functionality to be reused in different processes and can be easily described in design models that should be the result of analyzing the business requirements described in enterprise models.

The quantity of large research programs (Stein et al. 2006, Elhadad et al. 2008) indicate a growing demand for management methods, helping organizations map business requirements documented in enterprise models onto information systems.

To get into specific problem domains the modeling approach is an important technique gaining essential benefit trough the reduction of complexity by abstraction which facilitates analysis of complex systems (Balzert 1994, Ferstl and Sinz 1994). Models can be understood as the result of a construction "... done by a modeler, who examines the elements of a system for a specific purpose ..." (Schütte and Rotthowe

1998). Hence, conceptual models describe mental representations of real world phenomena with constructs representing structural (e.g. things and their properties) as well as behavioral aspects (e.g. events and processes). A method to create enterprise models consists of a modeling grammar (language), providing a set of elements and rules to combine the elements and a procedure by which a grammar can be used (Wand and Weber 2002). For further scientific analysis, we will use conceptual models as synonyms to enterprise models.

The construction of enterprise models often occurs within the requirements engineering during the information system development (Wand and Weber 2002). The semantic mightiness of their modeling language elements has to cover non-formal aspects supporting a deep understanding of the business domain and of the potential of an information technology employment as well as formal aspects in order to support the system implementation (Frank 1999). Thus, we use semi-formal languages to model problems which are not well-structured, highly subjective, individual and finally not objectively well formed (Harel and Rumpe 2000). Objective of the present paper is the introduction of a model-based method to manage authorization in Service-oriented architectures (SOA). The present modeling purpose implies the transformation of the problem description modeled with semi-formal languages into a shape that allows an appropriate automatic processing. However, this process is nontrivial, iterative, and can not be fully automated.

This paper presents an approach to introduce artifacts to establish a consensus on language level which enhances the comparability of models and allows to semi-automate the transformation process by weakening the strict separation of language creation and language usage. In contrast to previous approaches, this approach enables a model based configuration of service-oriented architectures that allows an automation of activities of the SOA development process that are currently carried out manually. As a result, the technical knowledge relevant to cope with the task is reduced, which at the same time shortens the time and effort for solving the overall task. This promotes the goal-oriented configuration of SOA and its underlying access management. As specific technical knowledge is no longer needed, the modeler can concentrate on the analysis of the business problem. To methodologically support a model-driven solution of this problem belongs to the real challenges in the field of SOA. A semiautomated use of enterprise models in the context of a goal oriented adjustment between business requirements and SOA implies the following scenarios:

- If a SOA does not yet exist, the mapping of business requirements onto the application landscape concerns the preparation of the SOA design. Based on the process models, core functions shall be identified acting as candidates for a web service implementation. Due to the large amount of process models and its strong relationships, however, the task of identifying these service candidates has been realized to be very difficult. Thus, existing methods guiding such an identification (Ivanov and Stähler, 2005) could not been used. As the process models were documented using a modeling tool, they are available electronically. Thus, the idea grew up to support the identification of service candidates by using a formal algorithm implemented in a software system. The automated analysis of the ratios implies an integration operation of the several models. A comparison discloses the similarities and differences between models and is thus the basis of the integration operation. In this paper we introduce an approach to realize such a scenario. Ratios guide the identification of service candidates out of process models and show the feasablity of the approach by implementing in the context of the identity management project.
- If a SOA exists, the mapping of business requirements onto a service landscape concerns the examination whether certain business requirements formulated as business process models could be

realized through service compositions. An examination corresponds to a comparison (matching process) between analysis and design models (see Priemer 1995). Enterprise models must be adjusted to the available services, and later on prepared for a migration into service compositions models like BPEL. A prerequisite is the establishment of a connection between enterprise models and service models.

The same is true for security aspects of the (Web-) service Layer of the SOA. The accesses to each service in questions result in several possible access rights that should be granted or not granted to a certain person depending on his/her function within the organization. Since in normal identity management scenarios these access rights are controlled by roles and corresponding sets of rules and policies that map these roles onto access rights, business requirements also should result in a set of useful roles for this approach and a set of rules and policies. Responsibilities (organizational units, organizational functions which are recorded in personal data) for certain process activities would result in access rights if this process in question could be realized after model transformation as a service (candidate). Also in this case we have two scenarios that mirror the previous case:

- And identity management containing a certain set of roles is not yet existent. Then role candidates have to be identified based on modeled process responsibilities within process models. If these responsibilities are modeled in a way that mirrors the coverage of personal data within an identity management system, one could use this information during or after the identification of service candidates to look for useful sets of attributes to create a role which could be used to control access to the resultant services in question.
- If a set of roles is already existent, one needs to map the business requirements of process responsibilities and ownership to the already existing set of roles and to map these roles to the certain access rights for the services in question.

In both cases the result is primarily not just the set of roles but especially the set of rules and policies used for Role Based Access Control (RBAC), i.e. the connection between roles and access rights.

Both scenarios would result in certain model transformation. The identification of service candidates would result in composition models like BPEL or design models for implementing services that have been detected as good service candidates. The identification of roles would result in models that depicture the set of rules and policies that build the basis for an RBAC to be implemented within an identity management.

Our research is a matter of design science (Hevner 2004). Following the research method presented in (Verschuren and Hartog 2005), the paper is structured as follows. In the next section we briefly discuss the Description Kit approach. Based on this discussion, an application of this approach is introduced for the automatic analysis of business process models. Finally, the implementation of the approach within the research program is demonstrated. The paper finishes with a discussion, summarizing the research results and exposing open questions.

THE DESCRIPTION KIT APPROACH

In normal scenarios conceptual modeling is a methodology that – using semi-formal languages – has a high degree of freedom normally used to visualize certain aspects of a problem domain. However this freedom leads to many inconsistencies and conflicts. The automatic examination of enterprise models represents the main reason for restricting the freedom in modeling. First, considering the expected diversity of models inside a process landscape, manual evaluation will be strongly complicated.

Additionally, enterprise models prescriptively serve as a design for a target-structure of the organization and its information system that needs to be implemented. We discuss the restriction of the freedom of modeling in order to prepare enterprise models in a way that these could serve as an input for a transformation process in design models. In the literature the discussion about restriction of modeling freedom is limited to the missing standardization that hampers the comparability of models. However, to be able to express certain concepts embedded within some distinguished environment (purpose, culture, infrastructure, language, terminology) models have to be domain specific on the one hand, but comparable to models in other domains on the other hand.

This paper is based upon the Description Kit approach that at the same time offers a framework for restricted modeling without destroying the adaptability to certain different domains (see Dietz et al. 2009). The methodology includes an algorithm for comparing models in different domains and therefore is capable to not only dissolve certain standard comparability conflicts but also the conflict between language domains (e.g. between formal and material domains).

To serve as a method for model driven SOA and access management, the Description Kit Approach not only consists of integrated language and procedure descriptions of the model language(s) to use, but also of artefacts of the problem solving techniques and a reference to the problem domain. All this should be subsumed as the intentional aspect of modelling.

- Guidelines (that result in a constrained form of modelling) include the intentional aspect of the modelling process and can not be described by classical means of metamodeling (data or process modelling respectively). This intentional aspect interacts with the language and modelling process descriptions.

Only domain specific guidelines (which result in constrained modelling) with influences on the language and process based metamodel can complete a model based methodology for solving functional problems. To be useful in different domains, this should be done in a generic way. Generically should be described how real world phenomena and problems should result in guidelines for a restricted modelling. A real relation between the (real world) object and the model should be established. A modelling procedure model does not originally cover this aspect.

The Description Kit approach uses Description Kits (DescKits), which cover restricted describable ancillary information in adequately enriched enterprise models. DescKits represent the consensus of the speech community in terms of the amount and structure of certain linguistic concepts relevant for the business analysis. The Description Kit approach is generic enough to restrict every kind of modeling information in their description relating to the present modeling purpose.

Actual modeling constructs at ordinary modeling level (see Strahringer 1998) are so called Descriptions. Each Description uses a certain Description Kit and fills it with life (see Dietz et al. 2009).

Likewise, at Description Kit level constraints are created, which determine the parameters that may be included in Descriptions using the corresponding DescKit. Also it is indicated how Descriptions may be nested together.

A Description Kit Type (DescKitType) is a generic concept for Description Kits, which indicates of what type a Description Kit and accompanying Descriptions are. In particular, these types control the mapping algorithms described below. Beside the syntactic definition of a Description Kit and the constraints (presetting) for relationships between Descriptions of that DescKitType, a DescKitType

additionally references to the concepts of the utilized conceptual modeling language. A DescKitType corresponds to the actual concept behind a constrained modeled facet of the domain.

The Mapping Algorithm

Using the DK approach an algorithm has been introduced that is able to compare process models using a Description Kit Language (DKL) for source models and a design model describing services also using the same DKL as a target model. The original idea of this algorithm is to find service candidates for process functions within a process model, but it is described completely generic to be useful also in other scenarios (see again Dietz et al. 2009).

Mapping algorithm for (sub-)process chains

The mentioned mapping algorithm is able to compare Descriptions in a pairwise way. However, to identify service candidates for complete process chains the comparison of complete process chains or subprocesses is needed. Since this algorithm should be introduced generically (similar to the mapping algorithm) some preparations are needed. Every relation between Descriptions has a certain Relation Type (similar to Description Kit Types for Descriptions). For each Relation Type one needs to determine how information (in the sense of objects described by the Descriptions) "flow" along relations of this type. The goal should be to "fold" the information based on Descriptions along the relations into a very big (artificial) Description. The idea to do this is to understand (in a mathematical sense) each Relation Type as an operator that operates on the descriptions it connects to in a certain way (and results in a new Description).

Therefore for each Relation Type we need a set of operational rules, how to operate on Descriptions of a certain Description Kit Type. The operation itself should only be dependent on the Relation Type and the Description Kit Types. The main example is the following:

$$\begin{aligned}&\Big\{\text{Interface}\big\{\text{Input}\{\text{Object } A\}\big\}\big\{\text{Output}\{\text{Object } B\}\big\}\Big\}\\ &\qquad\xrightarrow{\text{Flow}}\Big\{\text{Interface}\big\{\text{Input}\{\text{Object } C\}\big\}\big\{\text{Output}\{\text{Object } D\}\big\}\Big\}\\ &\qquad\xRightarrow{\text{Result of the Folding}}\Big\{\text{Interface}\big\{\text{Input}\{A\cup(C-B)\}\big\}\big\{\text{Output}\{(B-C)\cup D\}\big\}\Big\}\end{aligned}$$

The operators ∪ and – here are predefined operators that are based on the previously mentioned mapping algorithm.

To determine $X \cup Y$ the mapping algorithm is applied to the Descriptions X and Y. In case of a matching the Descriptions X and Y will be joined, i.e. all embedded Descriptions, parameters, values and constraints will be matched to build up the resultant Description. In the case of no match both X and Y will be embedded itself into the resultant Description with no connection.

On the other hand, to determine $X-Y$ one again applies the mapping algorithm to X and Y. However, in case of a match, the difference of X and Y will be calculated, which means to remove all parts (embedded Descriptions, parameters, …) of X that match some part of Y. If X and Y even correspond completely, the result is empty. Vice versa, of X and Y do not match, the result of $X-Y$ is simply X. This example is motivated by the fact that in some process flow a certain process simply uses the result of a previous process, so that this information is just intermediate and can be ignored when considering the complete process (when trying to decide how applicable it is to realize it as a service).

These operators are then used to fold step-by-step the Descriptions that compose some (sub-)process chain along the relations to a single Description. This should be done in both domains to be compared.

Afterwards the normal mapping algorithm can be applied to the resultant Descriptions to get a measure for the correspondence of the complete (sub-)models.

MODEL DRIVEN ACCESS CONTROL

The requirement to manage the underlying security aspects for (Web) Services of a SOA comes of course along with the need of a management of the SOA itself. The security and access control not only changes when responsibilities within the organization change but they also have to be altered when either implementation aspects of the SOA or functional requirements change, that have to be covered by the SOA.

The security management based on an identity management relies on roles. While the identity management is mainly responsible for person to role assignments, synchronizing personal data within several systems and realizing or enforcing certain rights based on these roles within target application systems, the right management for a fine granular SOA is normally out of scope of an identity management system.

The following information is needed for this task:

- A catalog of all services within the SOA and all access rights possible for these services
- A catalog of all roles within the organization
- Rules for mapping roles to access rights
- Technical information about how to access the right management for particular services for providing the necessary information for enforcing rights

Especially the second point – to come to a set of roles that really covers all the needs for controlling the access right – together with the third point – as set of rules to do that – is one of the main challenges when introducing an identity management. But this is not only important during the introduction phase but in fact a role management that respects changes of requirements over time remains a very important task also when the identity management system is in production.

As mentioned, changing requirements may either alter the SOA itself or alter the roles and rules within the identity management. Therefore it is self-evident to use the same approach for a model driven control for both scenarios.

To achieve this goal we created Description Kits for entering role information (not necessarily roles itself) in a way that mirrors personal information stored within the IDMS into models. This DK is used to create descriptions for persons or roles respectively that should be able to use a certain process within a process model (e.g. “a student within department 3”, “secretary of head of department 3”, etc.).

The mapping algorithm mentioned above is now able not only to identify service candidates within process models or to (in a semi-automated way) map processes to services within a service catalog, but also – completely in the same way – identify role candidates based on the role information or map role information to certain already productive roles within some “role catalog” (offered by the identity management). So in addition to the (semi-automated) model transformation of an EPC – enriched by descriptions – to BPEL we also get a model transformation into a model the contains the list of services, the list of roles and all connections between these two sides.

Since this transformation process is completely electronically, the next step is to access this information and make it available to the identity management. To use this data, the identity management system (IdMS) must access the role information for each process, translate this information to account data and propagate this information to the Web service or user management that is responsible for the process in question. Therefore we created event-driven process chain (EPC) diagrams (Scheer 2000), which hold role information for each process, which is accessible via a Web service.

Using the Description Kit approach, it is easily possible to integrate role information (in an electronically accessible form) into models. As Description Kits specific roles are created. Each role can be formulated in different ways and by use of different parameters (e. g. course of study for a student, or the institute for a research staff member). This of course can be modeled as wildcard parameter. In addition to the specific roles, the organizational structure is modeled at Description Kit level. Thus, during the modeling process only the organizational unit has to be added for example to the role "employee".

Now, on the ordinary modeling level concrete roles like "student at the department of biology", "university administration staff member" or "head of the department 2.1" can be described using the Descriptions. The role information is either a given specific role – premodeled at DescKit level - or an arbitrarily created role descriptions using a certain Description Kit. Role information is added to each activity in an EPC chart.

In the concrete identity management scenario we have done the following: Using DescKitTypes we enriched ARIS models and created – among others – a "role" DescKitType together with corresponding ParameterTypes ("name", "description", …). With ValueTypes we can either create concrete roles ("student", "employee", …) at DescKit level or allow the possibility to create arbitrary roles (as DescKitType instances) on model level – or a combination of both.

Furthermore we created Web services, using E3+WS, to access this role data (see Juhrisch and Weller 2008). With the E3+WS method we are able to create Web services in a model-driven fashion. These Web services give access to the specific role descriptions (Descriptions of DescKitType ‚role') in the process model. Therefore, a Web service has been developed and was bounded to the DescKitType 'role'.

A first proof of concept was made for the IBM Tivoli Identity Manager ITIM (a modified workflow for account provisioning that accesses the model-data Web service) and a now second implementation was done for the Novell Identity Manager (in this case a driver that accesses the Web service to synchronize the role information with the identity vault and another driver to translate this information into entitlements to the services in question, see Figure 2). Now, if role information in an EPC diagram changes, the IDMS may react immediately to this changes and change the rights for the services in question.

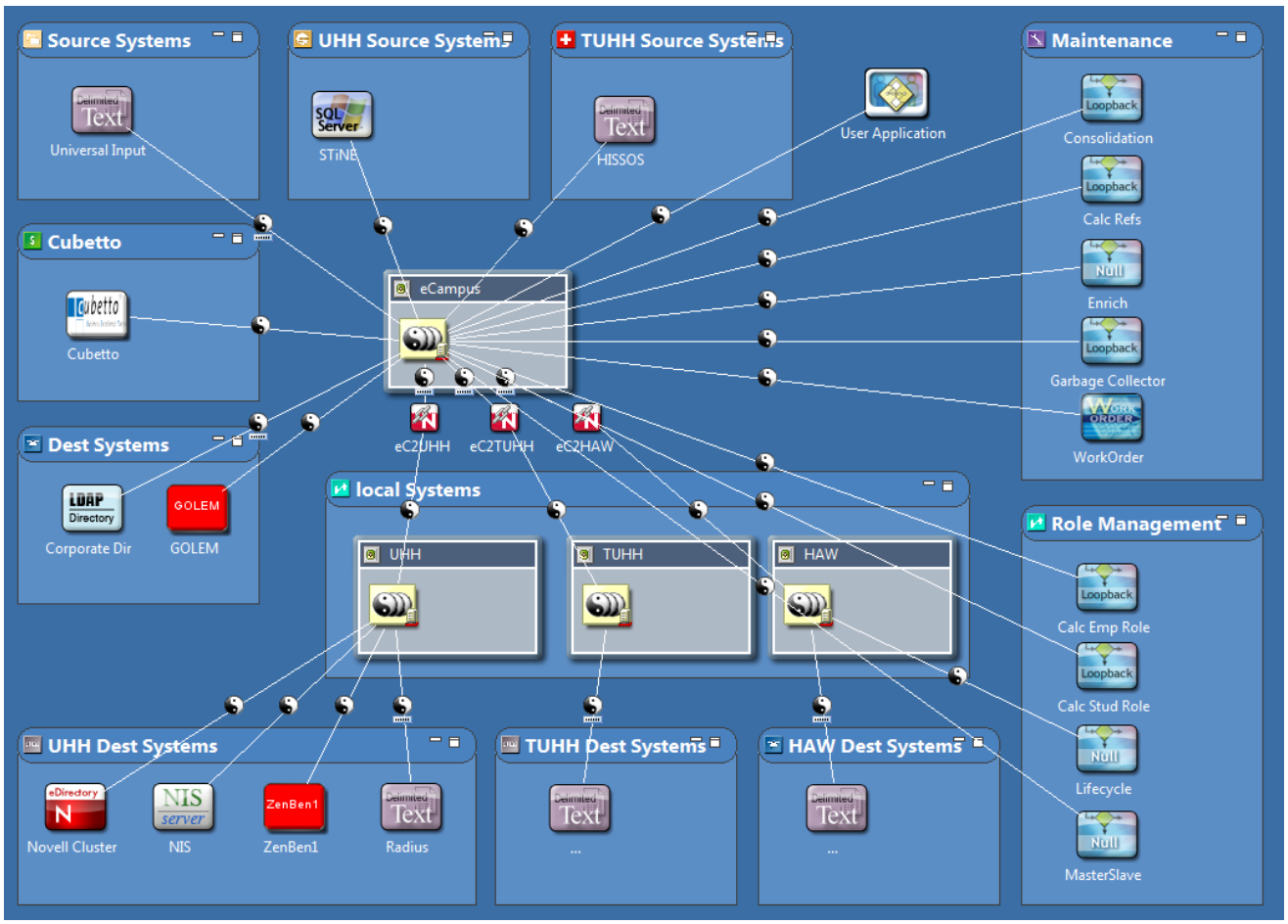

Figure 1. Design of the eCampus III IdMS and the integration of cubetto Toolset

IMPLEMENTATION

Project Description

Of course not only in Hamburg, Germany, as a city state with several universities, the increasing complexity of IT-based services and infrastructure creates an additional pressure to multilayered changes e.g. aroused by the Bologna process (European project for creating academic degree standards) or modern e-government. The six public universities of Hamburg, namely the University of Hamburg, the Hamburg University of Technology, the Hamburg University of Applied Sciences, the HafenCity University, the Hamburg University for Music and Drama and the University of Fine Arts, together with the university libraries in Hamburg have launched together with the MultimediaKontor Hamburg a project named eCampus, which is currently in the third phase eCampus III, running from 2009 to 2010.

Two main parts of this project are the joint implementation of an identity management system and synergetic adoptions of campus management systems. In the identity management project an architecture has been created that contains a central metadirectory which contains all identity data in a consolidated way and provisions local directories on each university. While the implementation phase is

still in progress the focus however moves from technical implementation details to the creation of a high service level and the optimization of processes. Therefore, starting with a pre-implementation process analysis now needs for managing the high complexity of changing requirements was one of the reasons for searching for a model-driven way to control the change management. Due to the cooperation of several universities with different requirements especially the aspect of distributed process modeling, comparism of models and model transformation came into the focus, which lead to a cooperation with the University of Technology Dresden (Germany) and the usage of the modeling tool "Cubetto Toolset" (Cubetto 2009).

The joint approach to identity management plans a cascading set of metadirectory implementations: one central metadirectory responsible for the consolidation of user data in an inter-university way and for the support of university-overlapping services; furthermore local metadirectories at least at each of the three bigger universities (the local services) that are responsible for the provisioning of internal university systems and to support internal services. The local services will be provisioned by the eCampus-IDMS (see Figure 2).

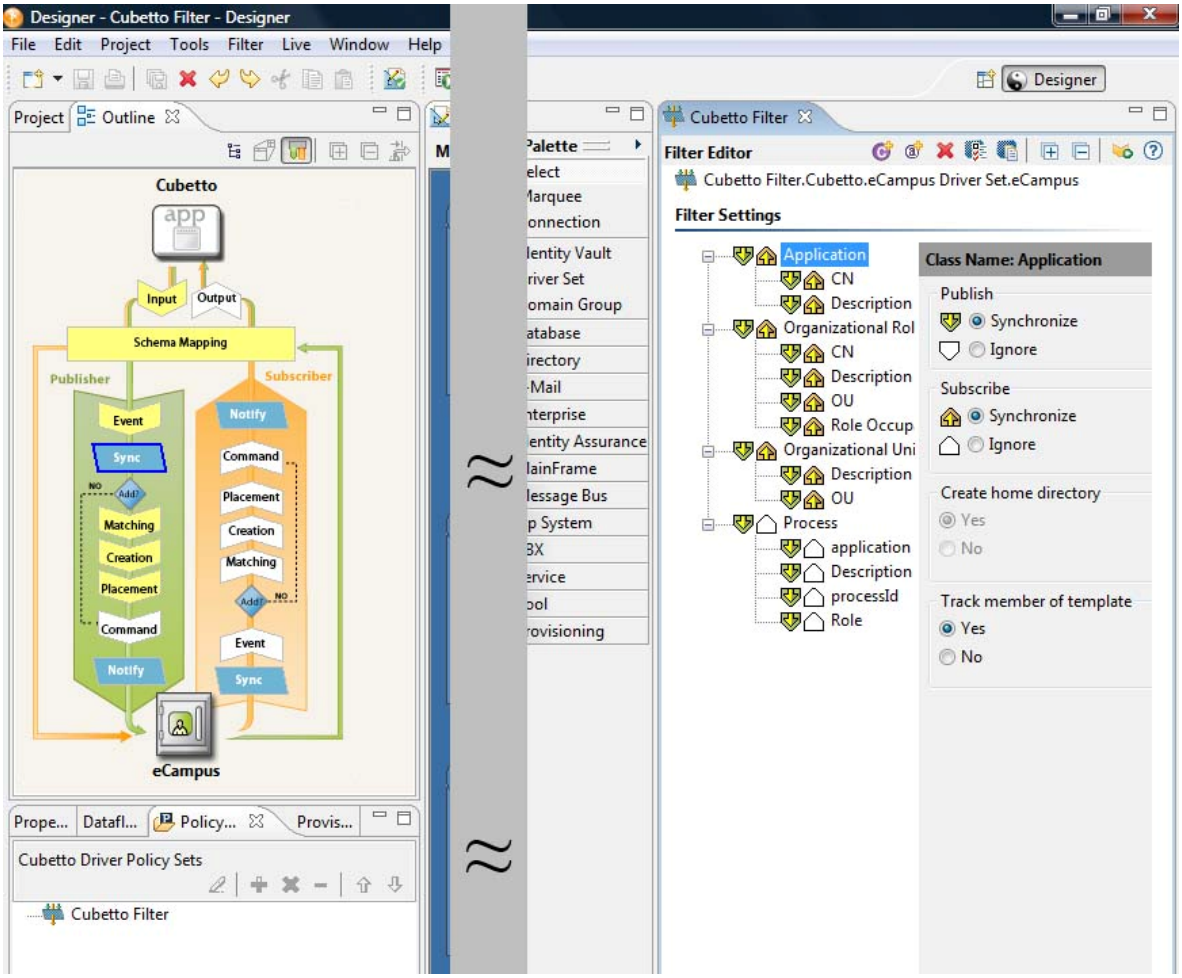

Figure 2: The cubetto driver for exchanging role data between models in cubetto and the IDMS

The software used in this project is the Novell Identity Manager, formerly known as DirXML. The Novell software supports so-called Role Based Entitlements (RBE) which assign dynamically and automatically entitlements based on the roles (or more generally some conditions of the attributes) of a person. The entitlements moreover control access to systems, assignment to groups (and therefore the assignment of some rights) or other things. All this can be controlled very flexibly by rules and policies.

Process documentation

For the documentation of the processes within the university project, we used the modeling tool "Cubetto Toolset" (Cubetto, 2009). The software was given to us free of charge. It offers three major advantages that were relevant within the project: Firstly, the tool allows the adjustment of the underlaying modeling language. Thus, it was possible to add concepts necessary for the creation of rations. Secondly, distributed modeling of processes is supported by an integrated configuration management (CM) system, including CM server. And thirdly, it is possible to increase the functionality of the tool by using so called plugins. Thus, the creation of ratios could be implemented easily.

Within the project the architecture of integrated information systems (ARIS) has been used. Thereby, the business processes were documented directly by the several departments (university and state library, student affairs office, computing center) and integrated afterwords into a large integrated model (containing of lots of sub-models) which was supported by the CM system. Due to the integration it was possible to identify web-services that are valid for the whole university. For implementing the approach, a plugin for the Cubetto Toolset was created. It analysis the integrated model and provides functionality to generate the introduced ratios.

CONTRIBUTION AND FURTHER RESEARCH

In this paper a framework for a model-driven control of identity management systems was introduced. A further analysis is necessary from the perspective of software development. In the present case, the service function could have serious impacts in the respective systems (e. g. removal from student registry by reasons of finally not passing the examination) and must be in a special way secured against abuse and manipulation. The next steps set out a completion of the technological aspects and a beta test with the first central users within the university. In addition to the technological aspects, there should be clarified which part of the data resides in the modelling tool and which part has to be saved in the identiy management system. The usefulness of the presented service catalogue lacks empirical evidence. Future work lies on the comprehensive evaluation in a pilot study conducted at the University of Hamburg. Going productive with cubetto® Toolset in the near future and involving local IT-support organizations of the university after that, the acceptance of the method will become apparent.

REFERENCES

1. Balzert, H. (1998) Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung. Spektrum, Heidelberg, Berlin, 571-574.
2. Ferstl, O. K. and Sinz, E. J. (1994) Grundlagen der Wirtschaftsinformatik. München, Wien, Oldenbourg Verlag.
3. Cubetto (2009) Specification, Semture GmbH, Dresden, http://www.semture.de/cubetto.

4. Dietz, G., Juhrisch, M. and Esswein, W. On the Restriction of Conceptual Modeling – Outlining an Approach to Enable Business Driven SOA. In *Proceedings of the 15th Americas Conference on Information Systems (AMCIS)*, San Francisco.
5. Frank, U. (1999) Conceptual Modelling as the Core of the Information System Discipline - Perspectives and Epistemological Challenges, Proceedings of the 5th America's Conference on Information Systems, AMCIS'99, AIS, Milwaukee, 695-697.
6. Erl, T. (2005) Service-oriented architecture: concepts, technology, and design, Prentice Hall PTR.
7. Elhadad, M., Balaban, M. and Sturm, A. (2008) Effective Business Process Outsourcing: The Prosero Approach. *International Journal of Interoperability in Business Information Systems*, 3, 1.
8. Harel, D. and Rumpe, B. (2000) Modeling languages: Syntax, semantics and all that stuff, part I: The basic stuff. Technical Report msc00-16, Weizmann Institute of Science.
9. Hevner, A. R. et al. (2004) Design Science in Information Systems Research. *MIS Quarterly*, 28, 1, 75-105.
10. Ivanov, K. and Stähler, D. (2005) Prozessmanagement als Voraussetzung für SOA. *OBJEKTSpektrum*, 12, 1-5.
11. Juhrisch, M. and Weller, J. (2008) Connecting Business and IT: A model-driven Webservice based Approach. In: *Proceedings of the 12th Pacific Asia Conference on Information Systems (PACIS)*, 1469-1479.
12. Krafzig, D., Banke, K. and Slama, D. (2005) Enterprise SOA – Service-Oriented Architectures Best Practices. Prentice Hall, Upper Saddle River, NJ.
13. Priemer, J. (1995) Entscheidungen über die Einsetzbarkeit von Software anhand formaler Modelle. Pro Universate Verlag, Sinzheim.
14. Scheer, A. (2000) ARIS - Business Process Modeling. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
15. Schütte, R. and Rotthowe, T. (1998) The guidelines of modeling: An approach to enhance the quality in information models, Lecture Notes in Computer Science (1507), 240-254.
16. Stein, S., Kühne, S. and Wagner, J. (2006) Das Forschungsprojekt OrViA – Orchestrierung und Validierung integrierter Anwendungssysteme. In Klaus-Peter Fähnrich, Stefan Kühne, Andreas Speck and Julia Wagner (Eds.) *Integration betrieblicher Informationssysteme, Problemanalysen und Lösungsansätze des Model-Driven Integration Engineering*. Eigenverlag Leipziger Informatikverbund (LIV), Leipzig, 3-12.
17. Strahringer, S. (1998) Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. In *Proceedings of the Modellierung 1998*, Astronomical Society of Australia.
18. Verschuren, P. and Hartog, R. (2009) Evaluation in Design-Oriented Research. *Quality & Quantity*, 39, 733-762.
19. Wand, Y. and Weber, R. (1990) An ontological model of an information system. *IEEE Trans. Software Eng.* 16, 11, 1282-1292.