PACIS 2007 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

2007

Towards a Model-driven Approach to Control Identity Management Systems

Martin Juhrisch
Westfälische Wilhelms-Universität, juhrisch@uni-muenster.de

Jens Weller
Technische Universität Dresden, jens.weller@tu-dresden.de

Gunnar Dietz
Westfälische Wilhelms-Universität, gunnar.dietz@uni-muenster.de

Recommended Citation

Juhrisch, Martin; Weller, Jens; and Dietz, Gunnar, "Towards a Model-driven Approach to Control Identity Management Systems" (2007). *PACIS 2007 Proceedings*. 149.
http://aisel.aisnet.org/pacis2007/149

135. Towards a Model-driven Approach to Control Identity Management Systems

Martin Juhrisch
Westfälische Wilhelms-Universität
Project MIRO
juhrisch@uni-muenster.de

Jens Weller
Technische Universität Dresden
Chair of Information Systems
jens.weller@tu-dresden.de

Gunnar Dietz
Westfälische Wilhelms-Universität
Project MIRO
gunnar.dietz@uni-muenster.de

Abstract

Identity management systems require consolidating the diversity of many user repositories, the creation of roles and their assignment to the identity management software. To decrease administration costs we propose linking the identity management system to business process models which contain the organizational responsibilities. Therefore, we introduce a modeling grammar to model web-services that access existing enterprise models. We illustrate the usefulness of our approach with a pilot study conducted at the University of Muenster (Germany).

Keywords: Enterprise Application Integration, Web-Services, Model Driven Architecture

Introduction

At the University of Muenster, Germany, the introduction of an identity management system is planned. The identity management will consolidate the diversity of many user repositories, currently existing at the university. Therefore, the individual person data are enriched with role information that will be used for a role based access control (RBAC) approach (Ferraiolo and Kuhn 1992) afterwards. Thus, the identity management will support the automation of inter-department workflows, avoid concurrent administrative operations and allow a better management of processes that manage accesses to several target systems.

The software that will be used in the identity management project is the IBM Tivoli Identity Manager (ITIM or TIM) (Buecker et al. 2006). The RBAC approach is supported by the so-called provisioning, which controls the creation, modification and deletion of accounts on target systems in a role based way. To do this, the software uses so-called provisioning policies (Buecker and Perttila 2006). A provisioning policy specifies under which circumstances an identity is entitled to have an account on one or more target systems (based on the role of the identity) and how the attributes of these accounts have to be set (based on the attributes of the identity). So one provisioning policy contains one or more entitlements and is triggered by one or more roles.

Even with the most powerful identity management software, however, two questions remain. Firstly, How to determine a useful set of roles and assigned them correctly to the users? Secondly, how to implement the provisioning policies so, that each user gets the correct rights on each target system? The configuration of an identity management system is usually an enormous effort with high complexity, especially at a university, where the number of different roles a person can have is very high. It involves an analysis of the organization structure and a detailed process analysis.

To ease that analysis and to avoid a manual adaptation of changing role assignment, in the university project, a model driven approach will be used. Therefore, we suggest an automated binding of functionality to enterprise models. We propose a scenario, where identity management supporting software directly accesses the enterprise models to adopt its behavior according to the described processes, organizational responsibility or whatever is stored within the enterprise models.

To ensure real-time access to the enterprise models, the content of the models should be available electronically. Due to the raising usage of modeling tools, this is not a problem in modern enterprises. Thus, a connection between the business software and the modeling tool (or the modeling tool data) is required. For such a connection, the SOA paradigm (Erl 2005; Stojanovic and Dahanayake 2005) offers interesting methods of resolution. Due to their advantages and potentials in the present case, the focus lies on the W3C's web-service standards.

In this paper, we present the grammar to model such web-services. Thereby we do not concentrate on a general modeling approach, but focus on the enrichment of existing enterprise models with web-service elements. To realize a solution for enterprise models in different modeling languages, our approach aims at the meta-model level. In addition to the modeling approach, we show its usability by demonstrating how the approach was used within the university project.

The paper is structured as follows. The next section introduces shortly the university research project. In section 3, we give a brief theoretical introduction into the specifics of enterprise models and enterprise architecture (EA). Section 4 introduces a framework showing how enterprise and web-service models can control university's systems functionalities. In section 5, we introduce our web-service modeling approach embedded in the method engineering field. Section 6 exposes the concept of the implemented approach. The paper ends with a discussion, summarizing the proposed ideas and exposing open questions regarding the realization of the application integration.

Project Description

The introduction of the identity management is part of the project MIRO. MIRO was launched November 15, 2005. During a period of five years the complex structure of one of Germany's largest universities is to be adequately provided with an innovative and powerful system for information management. In doing so, the requirements of the Bologna process (European project for creating academic degree standards) and of modern e-government must be taken into consideration. The same holds for the needs and wishes of about 40000 students, 5000 employees in 15 departments offering more than 130 study programs. MIRO strives to improve the information infrastructure in a systematic and continuous way to enhance the efficiency of key activities of the university (MIRO 2005).

The MIRO project consists of 10 work packages each of which is divided into several subtasks, 69 in total. Main tasks are the development of university's organization and services to a service-oriented architecture, the implementation of a central identity management, university portal and the enhanced development of innovative information retrieval techniques to improve on the efficiency of research, teaching and study by providing a powerful infrastructure for information search and delivery.

The authors of this paper are concerned with the analysis of core business processes and the establishment of identity management methods. The goal of the process analysis is to make visible the different processing steps and information flows within the different administrational units of the university. These will be documented in such a way that an assignment of roles and permissions is possible, what is a central issue for the technical infrastructure components of the identity management. This identity management with roles and permissions as its main concepts of access control permits a much more flexible administration of resources. The system will reduce the effort for user administration and lead to shorter response times, e.g. if a user's status changes or similar situations which require a lot of manual intervention in the old system. Furthermore the generated enterprise models help to deduce metadata and categories for the different types of university's organizational information.

The results of the project will be communicated to other universities encouraging them to adopt all or some of the solutions and to adjust them according to their own needs. For some results, this might also apply to companies and public administrations.

Background

Enterprise Modeling

In the course of business process reengineering activities modeling approaches are an important technique for an effective intervention into the problem domain (Hammer and Champy 1994). Essential benefit is attained through reduction of complexity by abstraction which facilitates analysis of complex systems (Balzert 1994; Ferstl and Sinz 1994). Hence, within the scope of first stage problem analysis and documentation, visual models are established – so called enterprise models representing business requirements for the actual enterprise situation (Fettke and Loos 2003). The semantic mightiness of enterprise modeling language constructs 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 need semi-formal languages to model problems, which are not well-structured, highly subjective, individual and finally not objectively well formed (Harel and Rumpe 2000).

Most grammars, however, do not possess a sufficient number of language constructs to model all phenomena in a domain (Agarwal et al. 1999). Consequently, we might need a multitude of modeling languages to model the dynamic changing business domains (Wand and Weber 2002). Therefore project or business situation dependent developed modeling methods have to be constructed adequately maybe only for one single project or only a short period of time (Brinkkemper 1996). Much time and effort is spent on applying standard modeling methods effectively in such projects since they are often too general and include parts, which are not suitable for the requirements of a specific problem context (Brinkkemper et al. 1998). The outcome of this is the need to adapt these methods or to construct them completely from parts of existing method fragments (Brinkkemper 1996; Harmsen et al. 1994). Any model or combination of it can be used, their meta-level languages transformed into or compared with each other hence we distinguish between object and meta-level language (Frank 1999). Software that supports generic method engineering is subsumed under the concept of meta-CASE tools like MetaEdit+ (Kelly et al. 2005) or Cubetto toolset (Cubetto 2007). In the course of an increasing number of enterprise modeling grammars, meta-CASE tools are needed to support language independent modeling. They provide a meta-modeling language to extend or to simplify modeling languages as well as to create new ones.

Enterprise Architecture

Understanding enterprises as complex information systems, there are not only business aims and business processes but also software and data. For a successful development of an enterprise, it is necessary to integrate these different views. Enterprise architectures (EA) have been established to fulfill this task (see Figure 1).

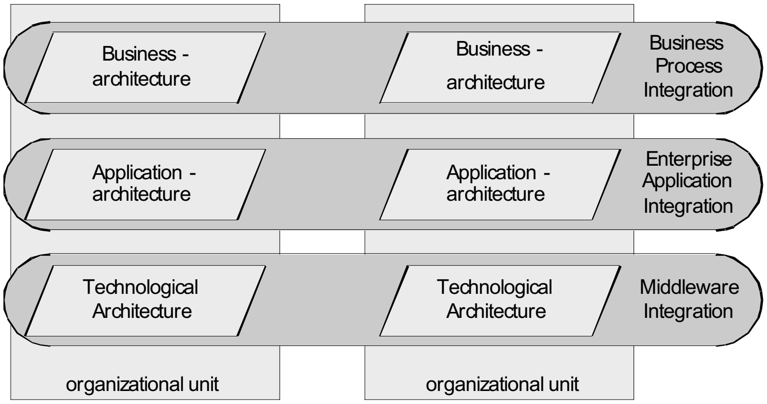

Business - architecture
Business - architecture
Business Process Integration
Application - architecture
Application - architecture
Enterprise Application Integration
Technological Architecture
Technological Architecture
Middleware Integration
organizational unit
organizational unit

Figure 1. Levels of integration (Hasselbring 2000)

The definition of EA varies (Beznosov 2000) but is often understood as a framework that "…provides a way of viewing a system from many different perspectives and showing how they are all related." (Sowa and Zachman 1992) Thereby, enterprise architecture should not been mixed up with software architectures, that are seen as the "…structure of the components of a program/system…" (Garlan and Perry 1995). Common understanding is that EA is the holistic view of horizontal and vertical integration of business processes, application architectures and the technological architectures. The horizontal architecture integration links different IT-applications, processes and data that exist separately side by side or creates new ones with the intention to optimize or develop new business processes stimulating synergy effects (Conrad et al. 2006). On the other hand vertical integration maps business processes, services and data to the IT-infrastructure using the vertical architecture layer. Essence is to merge the vertical layers to increase automation (Conrad et al. 2006).

Business process alignment (vertical integration) is currently reached with attached workflow management systems (WfMS) operating on workflow models standardized by the Workflow Management Coalition (WfMC 1994a; WfMC 1995b). These WfMS integrate applications via complex data interchanges or via orchestration languages like the Business Process Execution Language (WS-BPEL; Andrews et al. 2003). Prerequisites for WfMS-support is the ex ante modeling of all executable business processes by a central modeler (Krcmar et al. 1994). To raise flexibility of WfMS a multitude of business process variations ought to be modeled meaning a dropping number of executions per process definition, what is unprofitable using an extensive modeling technique (Kock et al. 1994). Functional adaptation is usually carried out by a complete new modeling and implementation.

SOA respectively Enterprise Application Integration (EAI) approaches (horizontal integration) have the idea to combine data- and methods of distinct software applications on the basis of a component based middleware, performing tasks like data transformation among heterogeneous formats as well as process control, security policy establishment and rollback functionality (Schmidtmann 2005). The separation between service interface and implementation is one of the fundamental characteristics of a SOA (Fremantle et al. 2002).

As a result, the business logic of an application system is separated from its implementation- and infrastructure details, too. Hence, the whole process within a web-service based architecture has to be modeled as a process model with involved flows, states and activities, implementing the coordinating business logic for a SOA (Schmidtmann 2005). Otherwise the coordinating logic ought to be implemented at any service consumer, what implies a complex and bad structured architecture. WS-BPEL for instance is a standardized XML based language to describe automated business processes, consisting of executable activities provided by various web service applications (Schönherr and Gallas 2003; Andrews et al. 2003). An executable WS-BPEL process definition represents a part of a combined business logic processed by communicating web-services (Aalst and Hee 2002). Each activity of a so described business process is associated to a corresponding WSDL-operation (Walsh 2002) in order of their chronological sequence. The execution of WS-BPEL processes is managed by an EAI-middleware (Schill 2000) that properly automates the flow of particular activities by providing connectivity to bounded applications.

While enterprise architectures can help understanding and managing the different layers, they generally do not solve problems arising when one view (or architectural layer) alters. In that case, the other views have to be adapted according to the dependencies existing between them. The proposed approach introduces a way for tied integration of enterprise models with the common EAI views: data, function, executable processes and protocols. That means, when an enterprise model (e.g. process model) changes, the views supporting that enterprise application have to be tailored. Our approach builds up on the paradigm of SOA but enlarges the domain to the integration with modeling CASE tools. The integration paradigm gains an additional aspect with the inclusion of enterprise model information.

Analogous to Arsanjani et al. the first step to integrate a software component into SOA or rather EAI architectures is functional decomposition (Arsanjani et al. 2003) meaning the extraction of functionality and data of a monolithic software system. The intention is to provide data and functionality to other components. This paper does not focus on legacy transformation in general but the introduced approach is related to the extraction of functionality and data of modeling CASE tools.

To sum up the recent section, modeling languages in the area of SOA derive from techniques out of the business process modeling scope. Besides the interface behavior the focus lies on composition of web-services as the main modeling aspect concerning in which manner web-services complements one another with the objective to model service-based business processes whereas major process parts are encapsulated as web-services communicating over organization boundaries. Integration between web-services and modeling languages arises from the objective to choose utilized web-services dependent from the actual business situation.

Integration Framework

The integration of modeling tools as web-services in a service-oriented architecture is spurred from the motivation to support real-time adjustment of application architecture to the business logic updated in enterprise models by domain experts. As shown in Figure 2 there exists two ways of integration between the application and the business architectural layer.

On the one hand, we can use formal WS-BPEL process definitions, and WfMS respectively to map business logic to an underlying web-service layer (invoked applications). This model organization results in a static separation between business logic from the underlying application architecture. Potentially this leads to a problem, as the business architecture and its application architecture evolve independently of each other.

Thus, the approach in this paper focuses on the control of addressable application components that are dependent largely on a business context represented in enterprise models. The context of a process flow is understand as the information passed between business process activities in an enterprise model together with the information provided by the modeler of the business process (Leymann, Roller and Schmidt, 2002). We achieve this transparency through the extension of conceptual modeling languages, e.g. ARIS (Scheer 2001) with language constructs that aim at the description of model information as web-services.

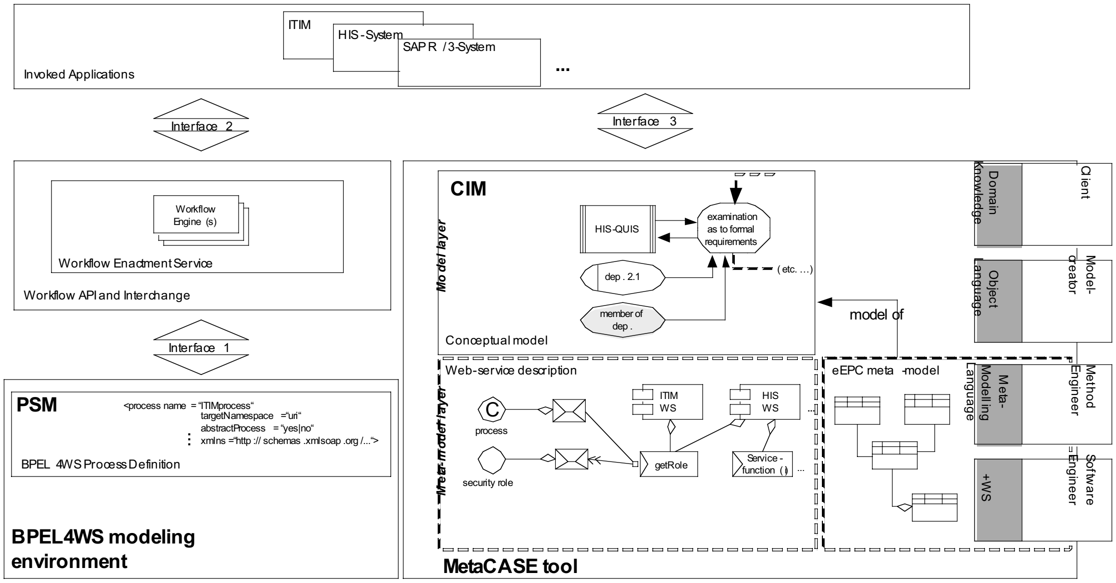


Figure 2. Different kinds of integration between application and business architectural layer

In terms of model-driven architecture's (MDA) levels of abstraction (MDA 2003) we can distinguish our approach from general workflow modeling approaches. Our integration framework is based upon computation independent models (CIM) representing domain specific information, independent of implementation technologies. By contrast, design languages (platform independent models, PIM) describe business logic combined with problem solutions in terms of software systems, thus these models are characterized by a significant higher degree of formalism. Accordingly these models induce platform specific models (PSM) that are the origin for executable IT artifacts. Recapitulating the three levels of abstraction in the MDA WS-BPEL process models serve as PSM representing executable web-service description and their specific compound behavior. Thus, an intersection between existing web-service composition languages and the present scope could not be found as XML tags are not sufficient for a semantic description in a computation independent modeling language.

Although different CIM are able to express requirements and behavioral models of target systems (Bézivin et al. 2004; Patrascoiu 2004; Thoene et al. 2002), no existing language supports integration and sharing of model information with IT applications through web-services of enterprise models. This is necessary, however, to implement the possible adaptation processes already before the analysis phase of the system engineering.

The property of web-services to be easily specifiable with formal design languages, leads to an evaluation of mapping possibilities from formal concepts of web-service design to the meta-level of conceptual modeling. In consideration of the outlined deficits we propose a semantically enriched meta-model for conceptual modeling. The generic of these web-services enables a flexible integration between infrastructure and enterprise models. This requires additionally to the description of system requirements during the modeling of business applications, a model of web-services, describing the services, which are used by the applications.

To summarize, the enterprise model orientation of our integration approach is the main difference to previous workflow modeling languages within the software engineering discipline (Busi et al. 2005). In contrast to the composition of web-services to executable business processes, our approach aims at a business model-driven control of a dynamic SOA infrastructure.

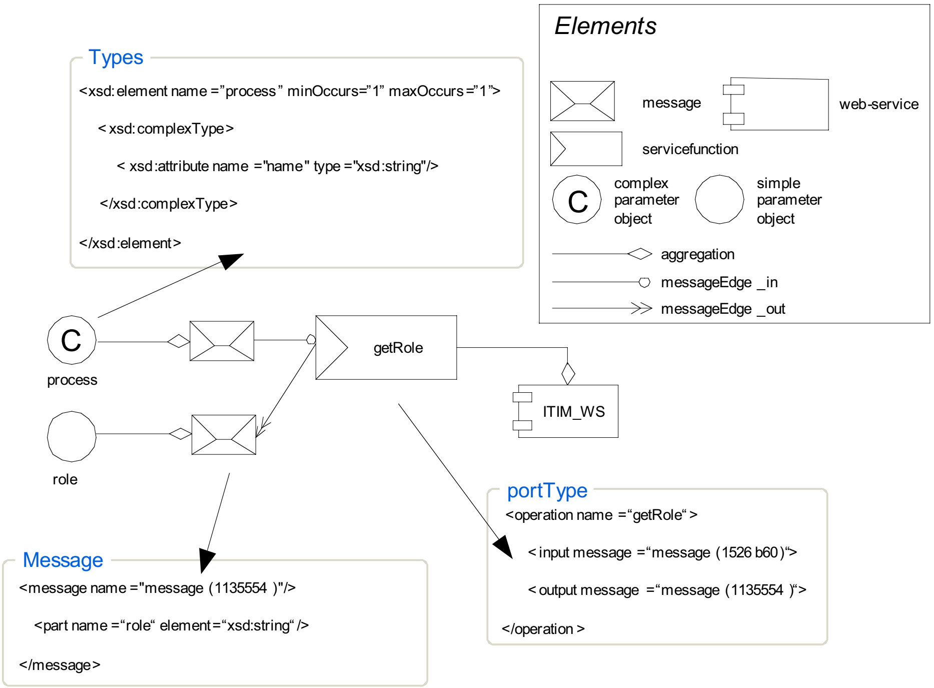
Modeling Approach

For language independent modeling of web-services, we extend an existing method of the method-engineering discipline. Thereby, web-service related constructs will be integrated into a meta-modeling language. We introduce the E³+WS method that on the one hand allows us to describe a web-service, based on meta-model information and on the other hand a model-driven realization of web-service functionality. The method engineering is based upon the existing E³-Model (E³; Greiffenberg 2004) classified as meta-meta-model on M3-Level of the Meta-Object-Facility Architecture (OMG 2002).

Abstracting from a determined notation, first step is the development of a language-based meta-model containing, in our case, all constructs to model a meta-model, a web-service interface and its implementation. In dividing the meta-model of E³+WS in separate views, we can simplify the process of construction. First, we turn our attention to the construction of the meta-model that we want to extend. We assume that all contained constructs and relationships between them are described over the E³-Model language conventions outlined in (Greiffenberg 2004) within an eE³-View (extended E³-View). Afterwards, we generate web-service descriptions within a separate web-service view. Thus, we generate separately a modeling language that on the one hand is able to express WSDL according to the W3C standard and on the other hand can be understood and adopted by target group users of the E³+WS method (see again Figure 2).

Centric element of our modeling approach is the web-service. Within a web-service-view, a web-service is constructed in two ways. The abstract definition takes place by adding its functionality and a unique name. Functionality is mapped through the aggregation of service-functions to web-services while the designator acts on the one hand as an ordinary identifier but also possesses a descriptive character by declaring the semantic of the web-service. A web-service element can be composed out of one or more service-functions by aggregation edges. Thus, web-services correspond to a number of similar service-functions. Furthermore, a service-function consists of an appropriate parameter assignment. Therefore, we define special message objects that act as containers for input respectively output parameter objects. Messages are connected with service-functions over so called messageEdge_in or messageEdge_out constructs depending on the role of the message regarding their related service-functions. Thereby a service-function can be connected to maximum one messageEdge_in and always exactly one messageEdge_out (see Figure 3). Hence, it is assumed that the possible service-functions communicate either with a request-response or with a notification pattern (Walsh 2002).

To define a single message parameter, we introduce the construct parameter object dividable into simple or complex objects. Simple parameter objects are assigned to simple data types. These types comprises on the one hand predefined XML schema data types and on the other hand own data types declared within the WSDL types tag. With self declared data types we are able to define for example appropriate report formats as potential result layouts for a model migration operation. A complex parameter object corresponds to design activities of a method engineer. Therefore, an E³-data type is introduced to establish ties between complex parameter objects and meta-model patterns. In the first move, we only tie E³ object types down to complex parameter objects. Message element and parameter object are analogical to service-function and web-service element connected over an aggregation edge. For more detailed information of the E³+WS modeling notation we reference the work of Weller et al. (Weller et al. 2006).



```
<xsd:element name ="process” minOccurs=”1” maxOccurs =”1”>
    < xsd:complexType>
        < xsd:attribute name ="name" type ="xsd:string"/>
    </xsd:complexType>
</xsd:element>
```

```
<operation name =“getRole“ >
    < input message =“message (1526 b60 )“>
    < output message =“message (1135554 )“>
</operation >
```

```
<message name ="message (1135554 )"/>
    <part name =“role“ element=“xsd:string“ />
</message>
```

Figure 3. Web-service modeling language

The main requirement for this framework represents the avoidance of any restriction to the ordinary modeling task on object level. Both domain (clients) and modeling experts (model creator) should not notice that their input into an enterprise model is made visible. Usually the construction of enterprise models requires “a specific competence that is neither covered by software engineering nor by organization science” (Frank 1999) so implementation related information have to be kept away. Additionally the integration of implementation aspects after the analysis phase of the system engineering would not facilitate the adaptation process of enterprise systems functionality.

Implementing the Approach

Within in the MIRO project, extensive process analysis activities were performed with the intention of complete transparency of university process organization (MIRO 2005). At the University of Muenster over 160 processes (e. g. “create invoice”, “create invoice reminder”, “cash payment”) were recorded using a modified ARIS modeling language (Scheer 2001). An electronic catalog of services has been created containing all recorded processes. For reasons of data retrieval, the processes are classified within the catalog according to departments (e. g. “human resources”) and/ or to specific events (e. g. “elections”).

To show that the presented theoretical approach is applicable to practice, we prototypically implemented the integration approach within the project MIRO using some of the recorded processes. Therefore, we extended the modeling language ARIS using E^3+WS to formulate role information appended to processes in EPC diagrams. Thereby, the ability to fulfill a process task (access right) is described by the associated role information. Figure 4 illustrates a change in role assignment at process “examination as to formal requirements”. The scenario is part of the EPC model of “approval for assignment to university course program”. The model change is probably initiated by a model creator as agreed with a domain expert. In our case the role as “head of dep.” at organizational unit 2.1 is for some reason not necessary anymore to process this activity with the HIS-QUIS system (HIS 2006). Hence, the role is changed to “member of dep.” as for example the head of dep. is requesting this task now from his secretary.

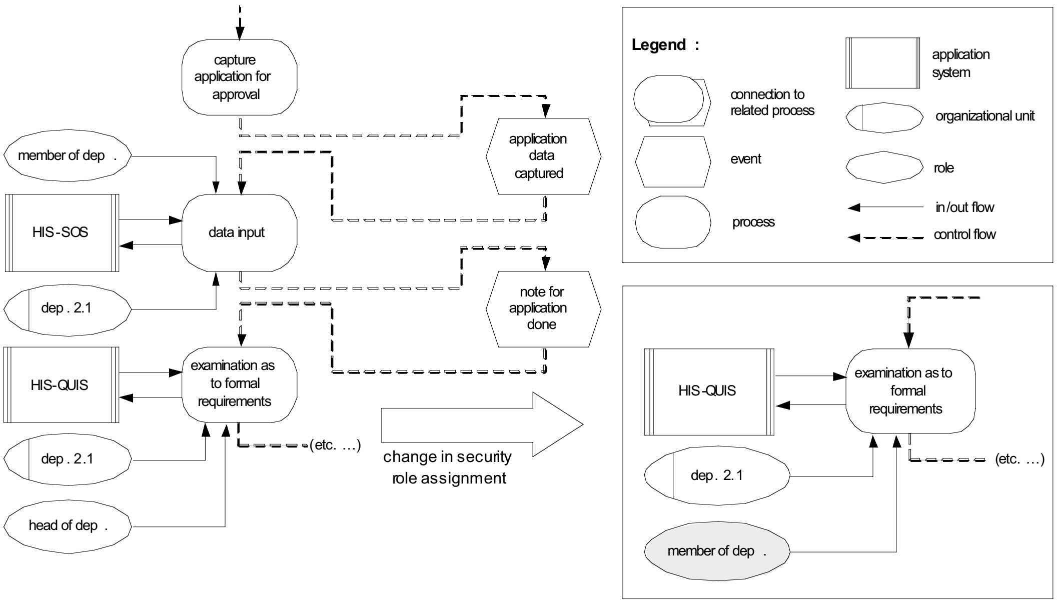

capture application for approval
member of dep .
application data captured
HIS-SOS
data input
dep . 2.1
note for application done
HIS-QUIS
examination as to formal requirements
dep . 2.1
head of dep .
(etc. ...)
change in security role assignment
Legend :
connection to related process
event
process
application system
organizational unit
role
in/out flow
control flow
HIS-QUIS
examination as to formal requirements
dep. 2.1
member of dep .
(etc. ...)

Figure 4. Changed role association to business processes indicate changed access rights

So now, consider that the EPC model contains role information for each process: For each process one or more roles are given that determine which role a person must have to use the (web) service for this process. This can be used to provide the identity management system with the needed information. For each process of the model, one can add a provisioning policy to the ITIM. This provisioning policy is used to provision the rights management of the application which holds the (web) service for this process. The ITIM communicates to the modeling tool via a web-service of the modeling tool to get the needed parameters (in this case the role information) of the process that corresponds to the provisioning tool. If some role for this process changes (as illustrated in Figure 4), the modeling tool informs the ITIM with the process ID for which this change happened. In the told way the ITIM then gets the new parameters (roles) for this provisioning policy. The modification of the provisioning policy then triggers the modification of all accounts in question. Figure 5 summarizes this process.

Until now, a first implementation of the approach is available for the meta-CASE tool Cubetto toolset (Juhrisch 2006). With the typecast of the extended ARIS meta-model, we enable on the one hand standard enterprise modeling based on the ARIS notation. Otherwise the described web-services in E^3+WS are automatically instantiated and enable the communication between Cubetto and TIM through automatically generated WSDL files. The implementation of a model-server for the automatic creation of associated web-services is in progress since the implementation of generic service-functions requires a complex model-driven technique to combine the referenced programming interface.

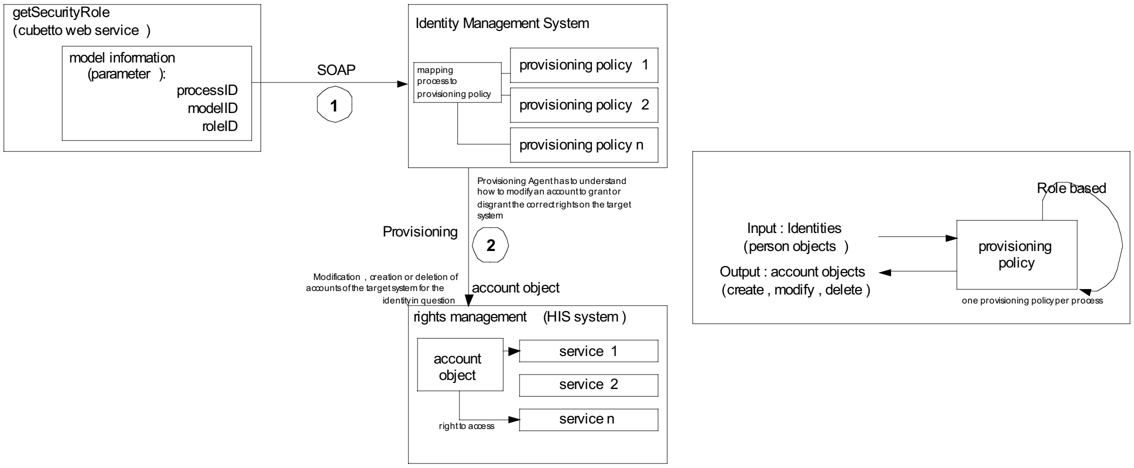

getSecurityRole
(cubetto web service)
model information
(parameter):
processID
modelID
roleID
SOAP
1
Identity Management System
mapping process to provisioning policy
provisioning policy 1
provisioning policy 2
provisioning policy n
Provisioning Agent has to understand how to modify an account to grant or disgrant the correct rights on the target system
Provisioning
2
Modification , creation or deletion of accounts of the target system for the identity in question
account object
rights management (HIS system)
account object
service 1
service 2
service n
right to access
Role based
Input : Identities
(person objects)
Output : account objects
(create , modify , delete)
provisioning policy
one provisioning policy per process

Figure 5. Web-service based integration of enterprise model and Identity Management system

Conclusions and Further Research

Within the MIRO project, over 160 university processes were recorded. With the approach presented in this paper, we are now able to create web-service descriptions that enable access to the information of the documented processes. Thus, we have created the foundation for a flexible architecture to control and manage rights in the identity management system through existing enterprise models.

The reuse of enterprise models by using them to manage application architectures comes up to a major challenge to increasing the use of enterprise analysis methods in businesses and organizations (Tissot and Crump 2006). Bringing model information out of the modeling tool Cubetto toolset to parameterize adaptive software application in an EAI environment helps to extend enterprise model utilization over its original domain of documentation to integration purposes with other software systems (Weller et al. 2006).

Our future work will focus the further technical implementation of necessary software parts of the framework. Until now, we realized the automated generation of WSDL files, but the web-services itself have to be created automatically as well. Thus, we will extend the model server administrating the (EPC) process models to create the WSDL files *and* the web-services implementation automatically. This will dramatically reduce the implementation effort of the necessary web-services.

Based on the technical implementation of necessary software parts, we will expand our pilot implementation to a detailed case study. Therewith, we can empirically evaluate the presented approach.

Additionally, we will discuss what kind of information should be added to the process models and what data should remain in the identity management system. This will also include a discussion about further scenarios of our approach that lay outside the identity management project.

References

Agarwal, R., De, P., Sinha, A.P. (1999). Comprehending object and process models: An empirical study. IEEE Transactions on Software Engineering, (25:4), 541-556.

Andrews, T. et al. (2003). Business Process Execution Language for Web Service, Version 1.1. at http://www-106.ibm.com/developerworks/library/ws-bpel.

Arsanjani, A., et al. (2003): Web Service: Promises and Compromises. In ACM Queue, 1, 1, 48-58.

Balzert, H. (1994). Development of Software-Systems: Principles, methods, languages, tools, BI, Mannheim, Germany.

Bézivin, J., Hammoudi, S., Lopes, D., Jouault, F. (2004). Applying MDA Approach for Web Service Platform. In Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC'04), ACM Press.

Beznosov, K. (2000). Information Enterprise Architectures: Problems and Perspectives. Technical report, School of Computer Science, Florida International University, Miami.

Buecker, A. and Perttila, J. (2006) Deployment Guide Series: IBM Tivoli Identity Manager. IBM Redbooks, http://www.redbooks.ibm.com/redbooks/pdfs/sg246477.pdf, retrieved 2006-04-19.

Buecker, A., Annas, A., Faustini, A., and Sanui T. (2006) Identity Management Advanced Design for IBM Tivoli Identity Manager. IBM Redbooks, http://www.redbooks.ibm.com/redbooks/pdfs/sg247242.pdf, retrieved 2006-05-10.

Busi, N., Gorrieri, R., Guidi, C., Lucchi, R., Zavattaro, G. (2005). Choreography and orchestration: A synergic approach for system design. In ISSOC'05, Lecture Notes in computer science / service-oriented computing, 228-240.

Brinkkemper, S. (1996). Method Engineering: Engineering of Information Systems Development Methods and Tools. Journal of Information and Software Technology.

Brinkkemper, S., Saeki, M., Harmsen, F. (1998). Assembly Techniques for Method Engineering. In Pernici, B., Thanos, C. (Eds.) CAiSE'98, LNCS 1413, 381-400.

Conrad, S., Hasselbring, W., Koschel, A., Tritsch, R.(2006). Enterprise Application Integration. Grundlagen – Konzepte – Entwurfsmuster – Praxisbeispiele. Spektrum Akademischer Verlag, 2-7, München.

Cubetto toolset (2007). Specifications, at http://wise.wiwi.tu-dresden.de/cubetto.

Erl, T. (2005). Service-oriented architecture: concepts, technology, and design, Prentice Hall PTR.

Ferraiolo, D.F. and Kuhn, D.R. (1992). Role Based Access Control, 15th National Computer Security Conference

Ferstl, O.K. and Sinz, E.J. (1994). Fundamentals of Information Systems, 1, R. Oldenbourg, Munich, Vienna.

Fettke, P. and Loos, P. (2003). Classification of reference models: a methodology and its application, Information Systems and e-Business Management, (1:1), 35-53.

Frank, U. (1999). Conceptual Modelling as the Core of the Information Systems Discipline – Perspectives and Epistemological Challenges. In Haseman, D.W., Nazareth, D., Goodhue, D. (Ed.) Proceedings of the Fifth America's Conference on Information Systems, AMCIS'99, AIS, Milwaukee, 695-697.

Fremantle, P., Weerawarna, S., Khalaf, R. (2002). Enterprise Services – Examining the emerging field of Web Services and how it is integrated into existing enterprise infrastructures. In Communication of the ACM, (45:10), 77-82.

Garlan, D. and Perry, D. (1995). Introduction to the Special Issue on Software Architecture. In IEEE Transactions on Software Engineering, (21:4), 269-274.

Greiffenberg, S. (2004). Method Engineering in Business and Government. Dr. Kovac, Hamburg, Germany.

Hammer, M. and Champy, J. (1994). Reengineering the Coorperation: A Manifesto for Business Revolution, HarperBusiness, New York.

Harel, D. and Rumpe, B. (2000). Modelling languages: Syntax, semantics and all that stuff, part I: The basic stuff. http://wisdom.weizmann.ac.il/archive/00000071/01/00-16.ps, Technical Report msc00-16, Weizmann Institute Of Science, retrieved 2006-02-20.

Harmsen, F., Brinkkemper, S., Oei, H. (1994). Situational Method Engineering for Information System Projects. In Proceedings of the IFIP WG8.1 Working Conference CRIS'94, North-Holland, Amsterdam, 169-194.

Hasselbring, W. (2000). Information System Integration. Communications of the ACM (43:6), p.43 33-38.

HIS (2006). Hochschul-Informations-System GmbH, at http://www.his.de.

Juhrisch, M. (2006) Model-oriented application integration using web-services at the Dresden University Clinic Center. Master Thesis, Department of Business Management and Economics, Chair of Business Informatics, Technical University Dresden, Germany, 2006.

Kelly, S., Rossi, M., Tolvanen, J.P. (2005). What is Needed in a MetaCASE Environment? In Frank, U. (Ed.), Enterprise Modelling and Information Systems Architectures (1:1), October, 22-35.

Kock, T., Rehäuser, J., Krcmar, H. (1994). Ein Vergleich ausgewählter Workflowsysteme am Beispiel eines Kreditvergabeprozesses. Arbeitspapier Nr. 66, Lehrstuhl für Wirtschaftsinformatik, Universität Hohenheim.

Krcmar, H., Schwarzer, B., Zerbe, S. (1997). Innovativer Werkzeugeinsatz zur Unterstützung prozeßorientierter Organisationen – Einsatz der IT zur Einführung prozeßorientierter Standardsoftware und zur Unterstützung flexibler Workflows. Chair for Business Informatics, Technical University of Munich, Germany, p. 20.

Leymann, F., Roller, D., Schmidt, M.T. (2002). Web Services and Business Process Management. In IBM Systems Journal, (41:2), 198-211.

MDA (2003). Guide Version 1.0.1, Document Number: omg/2003-06-01, Available at www.omg.com.

MIRO Project (2005). Integrated Information Management of a large university, IKM-Service, http://miami.uni-muenster.de/servlets/DerivateServlet/Derivate-2487/DFG-Antrag_Kurzfassung.pdf, retrieved 2007-02-03.

Object Management Group (2002). Meta Object Facility (MOF) Specification, version 1.4.

Patrascoiu, O. (2004). Mapping EDOC to Web Services using YATL. In Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC'04), ACM Press.

Scheer, A.W. (2001). ARIS – Modellierungsmethoden, Metamodelle, Anwendungen. 4. Auflage, Springer, Berlin, Germany.

Schill, A. (2000). Middleware by comparison. At http://www.competence-site.de/eaisysteme.nsf/C937534DE4B6BC14C1256A14005E5D60/\$File/middleware.pdf, retrieved 2006-09-19.

Schmidtmann, V. (2005). Web Services-basierte Referenzarchitektur für Enterprise Application Integration. Phd. Thesis, Institute for Business Informatics, Wissenschaftlicher Verlag, Berlin.

Schönherr, M. and Gallas, B.E.(2003). Komponentenbasiertes EAI-Framework unter Einsatz und Erweiterung von Web Services. In Uhr, W., Esswein, W. and Schoop, E. (Eds.), Wirtschaftsinformatik 2003, Medien – Märkte – Mobilität, Band II, 125-142, Dresden.

Sowa, J. and Zachmann, J. (1992). Extending and formalizing the framework for information systems architectures. In IBM System Journal, (31:3), 590-616.

Stojanovic, Z., Dahanayake, A. (2005). Service-Oriented Software System Engineering: Challenges and Practices, IDEA Group Inc.

Thoene, S., Depke, R., Engels, G. (2002). Process-Oriented Flexible Composition of Web Services with UML. In Proceedings of the Joint Workshop on Conceptual Modelling Approaches for e-Business

Tissot, F. and Crump, W. (1998). An Integrated Enterprise Modelling Environment. In Handbook on architectures of information systems, Bernus, P., Mertins, K. and Schmidt, G. (Eds.), 439-567.

Van der Aalst, W. and van Hee, K.M. (2002). Workflow-Management – Models, Methods, and Systems. MIT Press, Cambrigde, Massachusetts.

Wand, Y. and Weber, R. (2002). Research Commentary: Information System and Conceptual Modelling – A Research Agenda. Information System Research, (13:4), 363-376.

Walsh, A.E. (2002). UDDI, SOAP, and WSDL: The Web Services Specification Reference Book. Prentice Hall Professional Technical Reference, New Jersey.

Weller, J., Juhrisch, M. and Esswein, W. (2006). Towards using visual process models to control enterprise's systems functionalities. In Proceedings of the 12th Americas Conference On Information Systems (AMCIS 2006), August 4-6, Acapulco, Mexico.

WfMCa Workflow Management Coalition Members (1994). Glossary. Brussels.

WfMCb Workflow Management Coalition (1995). The Workflow reference Model. http://www.wfmc.org/standars/docs, retrieved 2006-08-04.