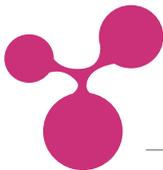


Technische Universität Dresden – Fakultät Informatik
Professur für Multimediatechnik, Privat-Dozentur für Angewandte Informatik

Prof. Dr.-Ing. Klaus Meißner
PD Dr.-Ing. habil. Martin Engelen
(Hrsg.)



GENEME '07

GEMEINSCHAFTEN IN NEUEN MEDIEN

an der
Fakultät Informatik der Technischen Universität Dresden

Unter Mitwirkung der
Comarch Software AG, Dresden und der
GI-Regionalgruppe Dresden

am 01. und 02. Oktober 2007 in Dresden
<http://www-mmt.inf.tu-dresden.de/geneme/>
geneme@mail-mmt.inf.tu-dresden.de

B.2 Steuerung Service-orientierter Architekturen durch Geschäftsprozessmodelle

Martin Juhrisch¹, Jens Weller²

¹Westfälische Wilhelms-Universität Münster, Projekt MIRO

²TU Dresden, Lehrstuhl für Wirtschaftsinformatik, insb. Systementwicklung

1. Einleitung

Im Jahr 2005 wurde an der Universität Münster (WWU) ein Projekt der Deutschen Forschungsgemeinschaft (DFG) mit dem Ziel gestartet, die technischen und organisatorischen Strukturen der WWU den Anforderungen anzupassen, die heute an eine große Universität gestellt werden. Kernpunkt ist die effektive Unterstützung existierender Geschäftsprozesse und die Etablierung neuer Geschäftsprozesse durch eine flexible Informationssystem-Architektur. Als Voraussetzung für integrierte Informationssysteme an der Universität wurde die Berücksichtigung unterschiedlicher Rechte und Rollen identifiziert, welche durch ein zentrales Identitätsmanagement (IdM) verwaltet werden sollen.

Das IdM wird die unterschiedlichen Benutzerverwaltungen universitätsweit konsolidieren. Dafür werden die individuellen Personendaten mit Rolleninformationen angereichert, um sie für den Ansatz einer Rollen-basierten Zugriffskontrolle (RBAC) zu qualifizieren [1]. Die Abbildung von Rollen auf Berechtigungen in Zielsystemen wird durch so genannte Provisioning Policies festgelegt [2]. Eine Provisioning Policy spezifiziert, unter welchen Umständen eine Identität berechtigt ist ein Nutzerkonto auf einem oder mehreren Zielsystemen (basierend auf der Rolle einer Identität) zu besitzen und mit welchen Attributen der Zugang gewährt wird (basierend auf den Attributen der Identität). Sie beinhaltet damit eine oder mehrere Berechtigungen für bestimmte Prozesse auf einem Zielsystem und wird durch eine oder mehrere Rollen angestoßen. Allerdings bleiben selbst mit dem leistungsstärksten IdM System zwei zentrale Fragen unbeantwortet: Erstens, wie wird eine nützliche Menge an Rollen definiert und wie ordnet man sie den Nutzern korrekt zu? Und zweitens, wie implementiert man die Provisioning Policies so, dass jeder Nutzer die richtigen Berechtigungen auf dem jeweiligen Zielsystem erhält?

Der vorliegende Beitrag schlägt zur Vereinfachung der Administration z. B. bei Anpassungen nach wechselnden Rollenzuordnungen, einen modellgetriebenen Ansatz vor, der die Anbindung von IdM Funktionalität an Geschäftsprozessmodelle ermöglicht. Das IdM soll direkt auf Informationen in den Modellen zugreifen und sein Verhalten entsprechend der dokumentierten Prozesse und organisatorischen Verantwortlichkeiten anpassen. Um einen solchen Zugriff zu ermöglichen, muss der Inhalt der Modelle

elektronisch verfügbar sein. Erforderlich ist die Verbindung zwischen dem IdM und einem Modellierungswerkzeug, welches die Modelldaten vorhält. Zur Realisierung des Ansatzes bietet das Paradigma der Service-orientierten Architekturen (SOA) interessante Methoden. Als Lösung wird eine Modellierungsgrammatik für Webservices auf Metamodellebene vorgestellt.

Ziel dieses Beitrags ist es, die entwickelte Modellierungssprache und deren Einsatz am Beispiel des IdM an der WWU zu präsentieren. Abschnitt 2 gibt eine kurze theoretische Einführung in die Terminologie von konzeptuellen Modellen und Service-orientierten Architekturen. Im dritten Abschnitt wird ein Framework vorgestellt, der die Steuerung der Funktionalität universitärer Informationssysteme über konzeptuelle Modelle erlaubt. Abschnitt 4 stellt die Modellierungsmethode vor und demonstriert ihren praktischen Einsatz am Beispiel des IdM. Der Artikel schließt mit einer Diskussion, die die wesentlichen Ideen des Konzeptes zusammenfasst und offene Punkte bezüglich der Realisierung der Integration beleuchtet.

2. Theoretischer Hintergrund

2.1 Konzeptuelle Modellierung

In der ersten Phase des Systementwicklungsprozesses, der Analyse und Dokumentation des Diskursbereichs, haben sich grafische Modelle etabliert – so genannte konzeptuelle Modelle. Diese Art von Modellen repräsentieren die Geschäftsanforderungen für die aktuelle Situation des Unternehmens [3].

Die semantische Mächtigkeit sprachlicher Konstrukte konzeptueller Modelle muss informale Aspekte abdecken, die ein tiefes Verständnis der betrieblichen Domäne und der Reorganisationspotentiale für den Einsatz von Informationstechnologie erlauben. Andererseits muss die Implementierung des zukünftigen Informationssystems durch formalsprachliche Aspekte vorbereitet werden können [4]. Benötigt werden semi-formale Sprachen zur Dokumentation und Strukturierung des gegebenen Problems. Die meisten Grammatiken besitzen jedoch keine ausreichende Anzahl an Sprachkonstrukten, um alle Phänomene in der jeweiligen Domäne zu modellieren [5].

Ein Ansatz, der sich dieser Problematik widmet, sind situationsabhängig entwickelte Modellierungssprachen, die für ein Projekt in einer bestimmten betrieblichen Domäne möglicherweise auch nur für einen kurzen Zeitabschnitt konstruiert werden [6]. Dafür werden entweder existierende Methoden angepasst oder völlig neu aus bestehenden Methodenfragmenten konstruiert ([6]; [7]). Ermöglicht wird dies durch die Unterscheidung zwischen der Objekt- und Metamodellierungssprache [4]. Software, die die generische Methodenentwicklung unterstützt, wird unter dem Konzept der Meta-

CASE Tools zusammengefasst. Bekannte Werkzeuge sind MetaEdit+ [8] oder das Cubetto Toolset [9]. Im Rahmen der steigenden Anzahl an konzeptuellen Modellierungssprachen werden Meta-CASE Tools benötigt, um sprachunabhängig zu modellieren. Sie implementieren eine Methode der Metamodellierung, um konzeptuelle Modellierungssprachen zu erweitern oder zu vereinfachen bzw. völlig neue zu konstruieren.

2.2 Service-orientierte Informationssystem-Architekturen

Unternehmensarchitekturen stellen eine ganzheitliche Sicht auf die horizontale und vertikale Integration von Geschäftsprozessen, Informationssystem-Architektur und deren technologisches Fundament dar [10]. Als Teil-Architektur (Informationssystem-Architektur) einer Unternehmensarchitektur kann eine SOA aufgefasst werden [11]. Mit ihr lässt sich die historisch gewachsene, heterogene Systemlandschaft sehr effizient an Änderungen im Geschäftsprozess anpassen. Dafür beschreibt SOA ein Systemarchitekturkonzept, das die Bereitstellung von Informationssystem-Funktionalität zur Abwicklung einer abgegrenzten betrieblichen Aufgabe in Form eines Service vorsieht. Die bereitgestellten Services sollen abstrakter Art sein und sich so zur Wiederverwendung auf fachlicher Ebene eignen. SOA stellt dazu erweiterte Anforderungen – insb. an deren Interoperabilität.

Services setzen die Idee der Modularisierung des betrieblichen Informationssystems durch die Aufteilung bislang monolithischer Anwendungssysteme in unterschiedlich kombinierbare Funktionsbausteine um. Als Standard für die Implementierung dieser Bausteine haben sich Webservices etabliert [11]. Die notwendige Infrastruktur basiert im Allgemeinen auf einem Web Application Server (WAS), einem Portalsystem und einem System zum übergreifenden Datenmanagement sowie zur Modellierung und Ablaufunterstützung von Geschäftsprozessen (Workflow Management). Sie leistet Aufgaben wie Datentransformation über heterogene Datenformate hinweg sowie Prozesskontrolle, Etablierung von Sicherheitsrichtlinien, Transaktionsmanagement etc. [12].

Neben dem Verhalten von Serviceschnittstellen liegt der Schwerpunkt des Einsatzes von Modellierungssprachen im SOA Umfeld klar auf der Komposition von Webservices. Im Vordergrund steht dabei die Art und Weise in der Webservices miteinander integriert werden, um Geschäftsprozesse, bei denen große Prozessteile durch eine Kommunikation zwischen Webservices in und über Organisationsgrenzen hinweg abgewickelt werden, zu implementieren. Integrationsbestrebungen zwischen Webservices und Modellierungssprachen ergeben sich aus der Motivation heraus

Webservices abhängig von der aktuellen Geschäftssituation im Unternehmen einsetzen zu können.

3. Integrations-Framework

Voraussetzung für die Ausrichtung einer SOA, allgemein einer Informationssystem-Architektur, an die existierenden Geschäftsprozesse ist die Vorabmodellierung aller automatisierbarer Geschäftsprozesse durch einen zentralen Modellierer und eine Modellierungssprache wie die Business Process Modeling Notation (BPMN) [13]. Um die Flexibilität zu erhöhen, müssen für eine Vielzahl von Geschäftsprozessvarianten entsprechende Modelle formuliert werden. Damit verbunden ist eine sinkende Anzahl an Ausführungen pro Prozessdefinition, was bei einer aufwändigen Modellierungstechnik wie BPMN bzw. später der Web Service Business Process Execution Language (WS-BPEL) unprofitable ist und die Effizienz des Ansatzes verschlechtert [14]. Eine Anpassung der Funktionalität bedeutet damit im Moment eine völlig neue Ablaufmodellierung und Abbildung auf das Informationssystem.

Das hier vorgestellte Framework schlägt in diesem Zusammenhang eine direkte Kopplung von konzeptuellen Modellen mit der Informationssystem-Architektur vor. Sobald sich eine Information in einem konzeptuellen Modell ändert, kann diese über Webservices von betreffenden Unternehmensanwendungen abgefragt werden. Anschließend richtet sich das Informationssystem auf die neuen Anforderungen aus. Unser Ansatz basiert auf dem SOA Paradigma, erweitert den Diskursbereich aber auf die Integration mit Modellierungswerkzeugen (Meta-CASE Tools).

Wie Abbildung 1 verdeutlicht, existieren zwei Möglichkeiten, eine Integration zwischen der technischen und fachlichen Architekturebene herzustellen. Einerseits können wir semi-formale BPMN Prozessdefinitionen in ausführbare WS-BPEL Prozesse transformieren, um die Geschäftslogik auf eine zugrunde liegende Webservice Schicht abzubilden. Diese manuelle Transformation führt zu einer statischen Verbindung zwischen fachlicher Geschäftslogik und der unterliegenden technischen Architektur. Dies impliziert potentielle Inkonsistenzen zwischen den Ebenen sobald sich fachliche bzw. technische Architektur unabhängig voneinander weiterentwickeln.

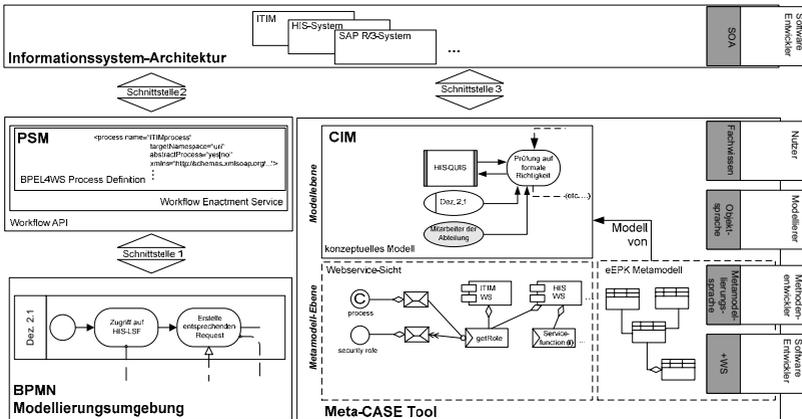


Abbildung 1: Integrationsarten zwischen fachlicher und technologischer Ebene

Der hier vorgestellte Ansatz dient der Steuerung von Anwendungssystemen, deren Ablauflogik wesentlich von einem in konzeptuellen Modellen dokumentierten betrieblichen Kontext abhängt. Der Kontext eines Geschäftsprozesses ist dabei die Gesamtheit an Informationen, die zwischen den Aktivitäten im konzeptuellen Modell weitergereicht werden, verbunden mit den Informationen, die der Modellierer dem Geschäftsprozess hinzufügt [15].

Obwohl verschiedene konzeptuelle Modellierungssprachen mächtig genug sind, eine umfassende Anforderungsanalyse der betrieblichen Domäne zu formulieren [16], unterstützt keine Sprache die Abfrage eines betrieblichen Kontexts zwischen Informationssystem-Architektur und Geschäftsprozessmodell auf Basis von Webservices. Dieses Defizit muss allerdings behoben werden, um Anpassungen der Architektur während der Analysephase im Systementwicklungsprozess berücksichtigen zu können. Um an dieser Stelle den Vorteil der Webservice-Technik, relativ einfach durch formale Entwurfssprachen spezifizierbar zu sein, für den vorliegenden Ansatz nutzbar zu machen, wird eine Beziehung zwischen den sprachlichen Konstrukten auf Entwurfsmodell- und auf der Metamodellebene der konzeptuellen Modellierungssprache hergestellt. Ziel eines mit Webservice Konstrukten semantisch angereicherten Metamodells ist die Modellierung abfragbarer Modellinformationen im Rahmen der Metamodellierung. Zusätzlich zu den Sichten der Anforderungsanalyse wird dafür das jeweilige konzeptuelle Modell durch eine Webservice-Sicht ergänzt, die beschreibt, welche Webservices angeboten werden (vgl. Abbildung 1).

Abschließend ist zu sagen, dass der direkte Bezug zu konzeptuellen Modellen den vorliegenden Ansatz von bisherigen Workflow Management Ansätzen in der

Softwareentwicklung unterscheidet [17]. Im Gegensatz zu der Zusammensetzung von Webservices zu ausführbaren Geschäftsprozessen zielt er auf eine fachliche, modellgetriebene Steuerung einer flexiblen Informationssystem-Architektur.

4. Modellierungsmethode

Um das Framework für eine Vielzahl unterschiedlicher Modellierungssprachen nutzbar zu machen, wird die sprachunabhängige Modellierung von Webservices favorisiert. Dafür erweitern wir eine existierende Methode aus dem Bereich der Methodenentwicklung. Mit der erweiterten Metamodellierungssprache können anschließend während der Entwicklung einer konzeptuellen Modellierungssprache Webservice bezogene sprachliche Konstrukte hinzugefügt werden. An dieser Stelle wird dafür die E³+WS Methode eingeführt werden, die zum einen erlaubt, Webservices für die Abfrage eines betrieblichen Kontexts im Metamodell zu beschreiben und zum anderen deren Funktionalität modellgetrieben zu entwickeln. Die Tätigkeiten im Entwurf und in der Implementierung des Ansatzes sind stark am Vorgehen bei der Methodenentwicklung und -anpassung nach Greiffenberg orientiert [18]. E³+WS basiert auf dem E³-Modell (E³; [18]), das als Metametamodell auf M3-Ebene der Meta-Object-Facility (MOF) Architektur [19] eingeordnet werden kann. Dieser Ansatz wurde gewählt, da es Ziel war, die erstellte Methode in ein von Greiffenberg et al. entwickeltes und auf der E³-Methode basierendes Meta-CASE Tool zu implementieren.

Abstrahiert man vorerst von der Spezifikation der Repräsentationsform, steht an der ersten Stelle die Entwicklung eines sprachbasierten Metamodells auf M3-Ebene. Es umfasst im vorliegenden Fall alle sprachlichen Konstrukte, um ein Metamodell einer Modellierungssprache zu entwickeln, Webservices zu implementieren und ihre Schnittstellen zu beschreiben.

Zur Vereinfachung der Konstruktion sind im Metamodell von E³+WS verschiedene Sichten vorgesehen. Die eE³-Sicht (erweiterte E³-Sicht) umfasst alle Sprachmittel des E³-Modells nach Greiffenberg und eignet sich damit für die Modellierung einer Modellierungssprache. Im Anschluss kann in der Webservice-Sicht die Modellierung der Webservice-Schnittstellen vorgenommen werden. Damit bietet der Ansatz eine zusätzliche Sicht, die zum einen in der Lage ist, WSDL nach dem Standard der W3C zu generieren und zum anderen von der Zielgruppe der E³+WS Methodenentwickler verstanden und angenommen werden kann (vgl. Abbildung 2).

Im Zentrum steht das Element Webservice. In der Webservice-Sicht werden Webservices auf zwei Arten spezifiziert. Seine abstrakte Definition erfolgt über die Spezifikation der Funktionalität und über die Zuordnung eines eindeutigen Namens. Dies wird durch die Aggregation von Servicefunktionen zu einem Webservice-Element

erreicht. Der Bezeichner dient der eindeutigen Identifizierung, besitzt darüber hinaus aber auch einen deskriptiven Charakter. Ein Webservice kann aus einer oder mehreren Funktionen bestehen. Die Zuordnung einer Funktion zu einem Webservice geschieht über die Aggregationskante – im Sinne einer Webservice-Servicefunktions-Zuordnung. Für jede Servicefunktion müssen des Weiteren geeignete Über- und Rückgabeparameter definiert werden. Diese werden über das Element Nachricht spezifiziert, das als Container für Input- bzw. Outputparameterobjekte dient. Die Beschreibung einer Servicefunktion durch ihre Nachrichten geschieht mit Hilfe einer Nachricht-Servicefunktion-Zuordnung, die durch die Konstrukte Nachrichtenkante_in und Nachrichtenkante_out abgebildet werden kann. Dabei steht eine Servicefunktion immer nur mit maximal einer Nachrichtenkante_in und immer genau einer Nachrichtenkante_out in Verbindung. Annahme ist also, dass eine Servicefunktion entweder das Request-Response-Muster oder das Notification-Muster implementiert [20].

Die Grundlage der Spezifikation der einzelnen Nachrichtenparameter ist das Konstrukt Parameterobjekt. Parameterobjekte können in einfache und komplexe Parameterobjekte unterschieden werden. Einfache Parameterobjekte sind zum einen vordefinierten XML Schema Datentypen wie String, Integer oder Double oder zum anderen selbst definierten Typen im WSDL Types Element zuordenbar. Bei der Definition komplexer Parameterobjekte handelt es sich um den Bezug zur Konstruktionsleistung des Methodenentwicklers in der eE³-Sicht. Dabei stellt ein E³-Datentyp die Brücke zu Konstrukten des zu koppelnden Metamodells dar. Beispiele für referenzierte Objekttypen sind Organisationseinheiten oder Rolleninformationen. Die Verbindung zwischen Parameterobjekt und Nachricht wird analog zur Service-Funktions-Zuordnung durch eine Aggregationskante abgebildet. Für detaillierte Informationen über die Notation von E³+WS wird auf Weller et al. verwiesen [21].

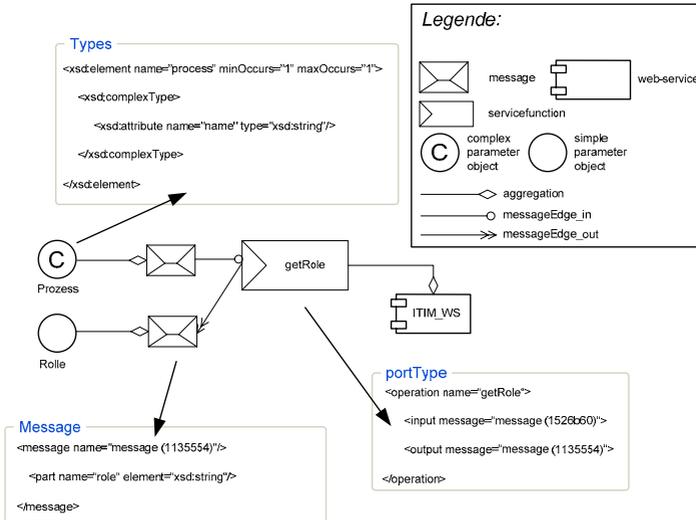


Abbildung 2: Webservice Modellierungssprache

Um möglichst vollständige Transparenz über die Aufbau- und Ablauforganisation herzustellen, wurden an der Universität Münster umfassende Prozessanalysen vorgenommen. Über 160 Prozesse sind dabei in verschiedenen Bereichen wie Prüfungsämtern und den einzelnen Dezernaten mit Hilfe einer erweiterten ARIS Notation [22] aufgenommen wurden. Zu Beginn der Implementierung des vorliegenden Ansatzes wurde dafür ARIS mit Hilfe von E³+WS erweitert und besitzt nun die Fähigkeit Rolleninformationen abhängig zu Prozessaktivitäten in EPK Diagrammen abzubilden und via modelliertem Webservice (vgl. Abbildung 2) verfügbar zu machen. Zur Demonstration des E³+WS Ansatzes stellt Abbildung 3 eine Veränderung in der Rollenauszeichnung des Prozesses „Prüfung auf formale Richtigkeit“ dar. Das Szenario ist Teil des eEPK Modells „Genehmigung der Einschreibung in das Lehrprogramm der Universität“ des Dezernats für studentische Angelegenheiten. Es wird angenommen, dass die Änderung im Modell durch den Modellierer in Abstimmung mit einem Fachexperten vorgenommen wurde. Im vorliegenden Fall ist es nun aus einem bestimmten Grund nicht mehr notwendig, dass der Dienststellenleiter des Dezernats 2.1 den Prozess „Prüfung auf formale Richtigkeit“ mit dem HIS-QUIS System [23] selbst ausführt. Die zugeordnete Rolle wurde geändert auf „Mitarbeiter des Dezernats“, um möglicherweise der Sekretärin des Leiters die Berechtigung zu geben, diese Aufgabe zukünftig wahrzunehmen.

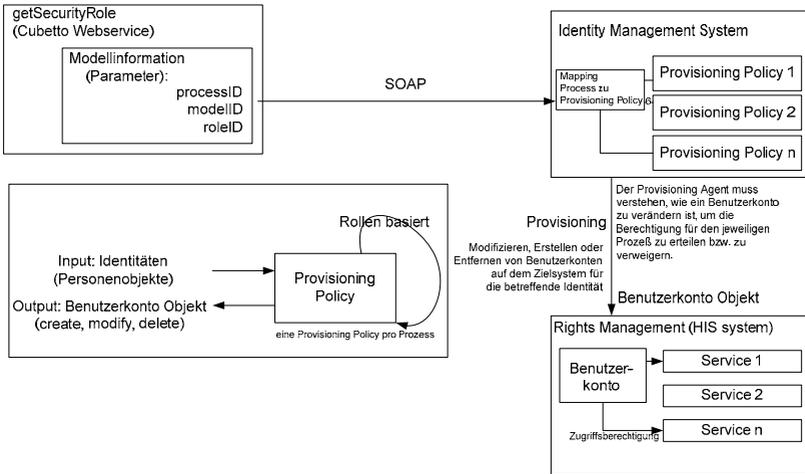


Abbildung 4: Integration zwischen Unternehmensmodell und IdM

Eine erste Implementierung des Ansatzes liegt für das Meta-CASE Tool Cubetto Toolset vor [16]. Mit der Typisierung des erweiterten ARIS Metamodells ermöglichen wir einerseits standardisierte Unternehmensmodellierung, andererseits werden die in E³+WS modellierten Webservices instanziiert und ermöglichen die Kommunikation zwischen Cubetto und ITIM über automatisch generierte WSDL Dateien.

5. Zusammenfassung und Ausblick

Die Nutzung konzeptueller Modelle zur Verwaltung von Anwendungssystemen greift die Forderung nach einer intensiveren Wiederverwendung von Artefakten aus der Analysephase der Systementwicklung auf [24]. Indem Modellinformationen verfügbar gemacht werden, um anpassbare Softwaresysteme zu parametrisieren, wird der Zweck der Erstellung von Unternehmensmodellen auf die Steuerung von Softwaresystemen ausgedehnt [21].

Der Schwerpunkt unserer zukünftigen Arbeit liegt auf der technischen Entwicklung der verbliebenen Bestandteile des Integrations-Frameworks. Während die automatische Generierung der WSDL Dateien bereits realisiert ist, muss die Funktionalität der Servicefunktionen auch modellgetrieben entwickelt und automatisch erzeugt werden können. Neben den technischen Aspekten muss dafür geklärt werden, welche Informationen den Modellen hinzugefügt werden und welche Daten im IdM verbleiben sollten.

Literatur

- [1] Ferrailo, D.F., Kuhn, D.R., 1992, Role Based Access Control, 15th National Computer Security Conference
- [2] Buecker, A., 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
- [3] Fettke, P., Loos, P., 2003, Classification of reference models: a methodology and its application, *Information Systems and e-Business Management*, 35-53
- [4] Frank, U., 1999, Conceptual Modelling as the Core of the Information Systems Discipline – Perspectives and Epistemological Challenges, *Proceedings of the Fifth America's Conference on Information Systems, AMCIS'99, AIS, Milwaukee*, 695-697
- [5] Wand, Y., Weber, R., 2002, Research Commentary: Information System and Conceptual Modelling – A Research Agenda, *Information System Research*, 13, 4, 363-376
- [6] Brinkkemper, S., 1996, Method Engineering: Engineering of Information Systems Development Methods and Tools, *Journal of Information and Software Technology*
- [7] Harmsen, F., Brinkkemper, S., Oei, H., 1994, Situational Method Engineering for Information System Projects, *Proceedings of the IFIP WG8.1 Working Conference CRIS'94, North-Holland, Amsterdam*, 169-194
- [8] 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, Oktober, 22-35
- [9] Cubetto toolset, 2007, Specifications, at <http://wise.wiwi.tu-dresden.de/cubetto>
- [10] Sowa, J., Zachmann, J., 1992, Extending and formalizing the framework for information systems architectures, *IBM System Journal*, 31, 3, 590-616
- [11] Barry, D.K., 2003, *Web Services and Service-Oriented Architecture*, Morgan Kaufmann Publishers
- [12] Schmidtmann, V., 2005, *Web Service-basierte Referenzarchitektur für Enterprise Application Integration*, Dissertation, Institut für Wirtschaftsinformatik, Wissenschaftlicher Verlag, Berlin
- [13] 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*, Lehrstuhl für Wirtschaftsinformatik, TU München, S. 20

- [14] Koch, T., Rehäuser, J., Krcmar, H., 1994, Ein Vergleich ausgewählter Workflowsysteme am Beispiel eines Kreditvergabeprozesses, Arbeitspapier Nr. 66, Lehrstuhl für Wirtschaftsinformatik, Universität Hohenheim
- [15] Leymann, F., Roller, D., Schmidt, M.T., 2002, Web Services and Business Process Management, IBM Systems Journal, 41, 2, 198-211
- [16] Juhirsch, M., 2006, Model-oriented application integration using web-services at the Dresden University Clinic Center, Master Thesis, Wirtschaftswissenschaftliche Fakultät, Lehrstuhl für Wirtschaftsinformatik, TU Dresden
- [17] Busi, N., Gorrieri, R., Guidi, C., Lucchi, R., Zavattaro, G., 2005, Choreography and orchestration: A synergic approach for system design, ISSOC'05, Lecture Notes in computer science / service-oriented computing, 228-240
- [18] Greiffenberg, S., 2004, Method Engineering in Business and Government, Dr. Kovac, Hamburg
- [19] Object Management Group, 2002, Meta Object Facility (MOF) Specification, version 1.4
- [20] Walsh, A.E., 2002, UDDI, SOAP, and WSDL: The Web Services Specification Reference Book, Prentice Hall Professional Technical Reference, New Jersey
- [21] Weller, J., Juhirsch, M., Esswein, W., 2006, Towards using visual process models to control enterprise systems functionalities, Int. J. Networking and Virtual Organisations, 3, 4, 412-424
- [22] Scheer, A.W., 2001, ARIS – Modellierungsmethoden, Metamodelle, Anwendungen, 4. Auflage, Springer, Berlin
- [23] HIS, 2006, Hochschul-Informationssystem GmbH, <http://www.his.de>
- [24] Tissot, F., Crump, W., 1998, An Integrated Enterprise Modelling Environment, In Handbook on architectures of information systems, Bernus, P., Mertins, K., Schmidt, G. (Eds.), 439-567